

FTP2166000EQ
FTP-POS
POSPrinter OPOS Control
Function specification

Version 7
Jun 12, 2024

FCL COMPONENTS LIMITED

Table of Contents

1. Outline	4
1.1. Subject scope of this document.....	4
1.2. OPOS Control Outline.....	5
1.3. Restrictions.....	7
1.4. Attentions.....	7
1.5. Trademark.....	8
2. How to install OPOS Control	9
2.1. Installation condition	9
2.2. Supported printers	9
2.3. Installation media.....	11
2.4. Installation procedure	12
2.5. Installation file list	16
3. How to use the OPOS Control.....	17
3.1. Common	17
3.2. POS Printer	17
3.3. How to implement the OPOS Control.....	18
4. OPOS Interface specifications (Printer)	23
4.1. Summary	23
4.2. Data Characters and Escape Sequences	29
4.3. Common Properties.....	33
4.4. Common Methods.....	43
4.5. Specific properties.....	56
4.6. Specific Methods	73
4.7. Events	104
5. Log file output.....	108
6. Printer Specification.....	110
6.1. FTP-627DSL440	110
6.2. FTP-629DSL310	113
6.3. FTP-629DSL350	116
6.4. FTP-839DSL310	119
6.5. FTP-62GDSDL000	122
6.6. FTP-63GDSDL483/FTP-83GDSDL483	126
6.7. FTP-62HWSL001	129
6.8. FTP-62HDSL100	133
6.9. FTP-62EDSL200.....	136
7. POS for .NET	139

FTP2166000EQ

FTP-POS

POSPrinter OPOS Control Function specification

7.1. Install	139
7.2. Attentions.....	140
7.3. Difference from OPOS.....	141
8. Open Source Software.....	149
8.1. OPOS Common Control Objects	149
8.2. Apache License Version 2.0.....	150
9. Update History	153
10. Driver Update History	154

1. Outline

POS Printer OPOS Control that controls FTP-POS POSPrinter OPOS control is OPOS Control conforming to OPOS 1.9 POS Printer device. When using OPOS Control, refer to "UnifiedPOS specification Version 1.9" document.

In this document, "OPOS Control" means "OPOS OCX".

It is also possible to develop POS for .NET applications using this OPOS control. Please refer to [POS for .NET](#)

1.1. Subject scope of this document

These instructions aim for the main reference of programmers who develop the application for the use of FTP-POS POSPrinter OPOS control, and describe the following contents.

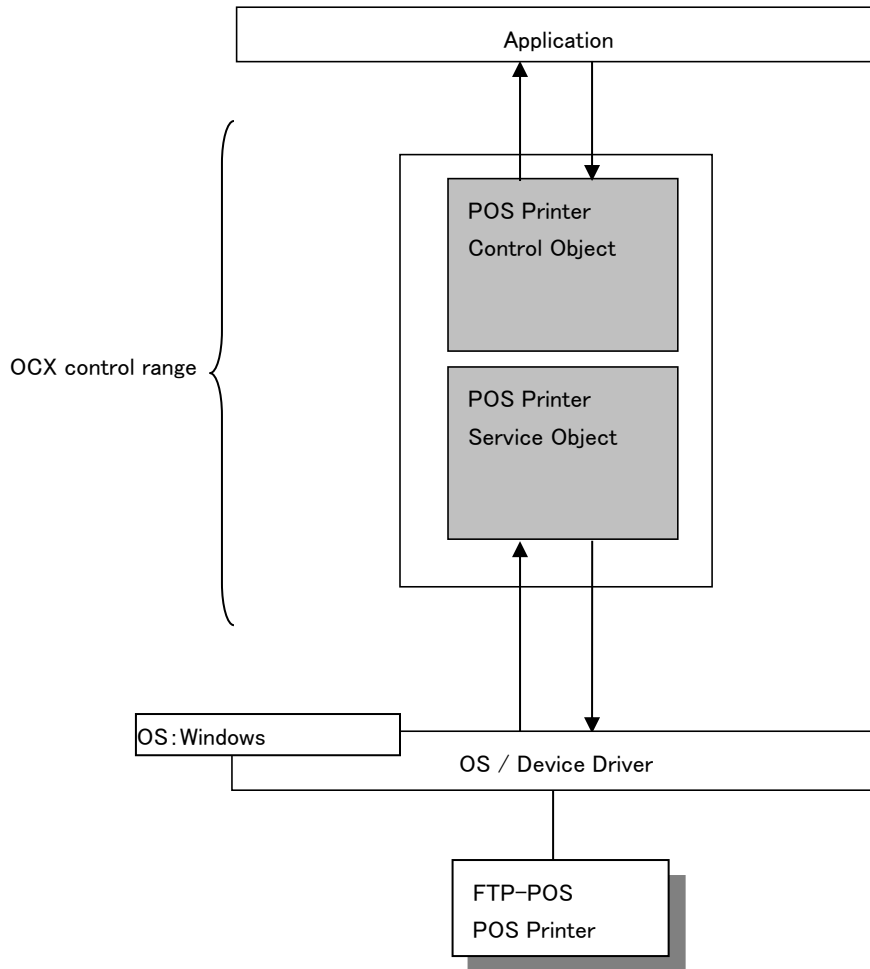
- How to install OPOS Control
- How to use the OPOS Control
- Restriction of OPOS Control
- OPOS Interface specifications (Printer)
- Log file output

1.2. OPOS Control Outline

1.2.1. OPOS Control Configuration drawing

OPOS Control conforms to the ActiveX control specification and provides Property, Method, and Event to the application. Control can not be seen on UI in application execution, and only the application that uses it requests to process through Method and Property. The application receives processing result through the return value or parameter of Method, Property, and Event. This OPOS Control is implemented as an in-process server.

OPOS Control can control only one POS Printer connected with POS (PC).



1.2.2. Terminology

1. Control Object; CO

CO provides application with the set of Property, Method, and Event according to each device class.

This document explains these API.

2. Service Object; SO

SO executes the function which is called from Control Object and which is specified by OPOS for each device.

1.3. Restrictions

There are the following restrictions.

1.3.1. POS Printer

[Restrictions on OPOS specifications]

There are the following restrictions though all interfaces of the OPOS POS printer device are provided.

1. Journal printing and journal property settings are not supported.
2. Slip printing and Slip property settings are not supported.
3. Italic, bold, custom color, shading printing, and cartridge function are not supported.
4. The change of receipt printing character font is not supported.
5. After Enabled, the following methods always return OPOS_E_ILLEGAL(106) .
PrintTwoNormal method
BeginInsertion method
EndInsertion method
BeginRemoval method
EndRemoval method
ChangePrintSide method
ClearPrintArea method
PageModePrint method
ResetStatistics method
RetrieveStatistics method
UpdateStatistics method
6. Only when connected via RS-232C, errors generated after disconnection or power off are notified after connection or power on.
7. Other features may not be supported by the printer. Please refer to [Printer Specification](#).

1.3.2. Common restrictions

This OPOS Control operates under Single Thread Apartment (STA) model, POS Control must be called from only one thread (It is generally the main thread of the application) in the process.

This OPOS Control does not support the simultaneous use from two or more sessions (user). When accessing Method and Property from two or more sessions, the unintended consequence may be caused.

This OPOS Control can control only one POS Printer.

This OPOS Control is not supported for CCO.

1.4. Attentions

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

1.5. Trademark

Microsoft®, Windows®, Visual Studio®, Visual C++®, Visual Basic® and ActiveX® are registered trademarks or trademarks of Microsoft Corporation in the United States and other countries.

The proper name of Windows is Microsoft Windows operating system.

The Bluetooth® word mark and logo is a registered trademark of Bluetooth SIG, Inc. FCL COMPONENTS LIMITED uses these trademarks under the terms of license. Other trademarks and trade names are the property of their respective owners.

QR Code® is a registered trademark of DENSO WAVE INCORPORATED in Japan and in other countries.

Wi-Fi® is a trademark of Wi-Fi Alliance.

All other product and brand names are trademarks and/or registered trademarks of their respective companies.

2. How to install OPOS Control

FTP-POS POSPrinter OPOS control is installed according to the following procedure.

2.1. Installation condition

■ Operation environment

- OS : Microsoft Windows 7 / 8.1 / 10 / 11 32-bits / 64-bits (WOW64 only)
- CPU : System requirements for each operating system.
- RAM : System requirements for each operating system.
- HDD : Space 15MB or more

When installing, logon by an account with administrator privileges.

2.2. Supported printers

	Interface				Remarks
	USB-COM	RS-232C	Bluetooth SPP	Wi-Fi	
FTP-627DSL440	✓	✓			Please select printer model to "FTP-627DSL440" by configuration tool.
FTP-629DSL310	✓	✓			Please select printer model to "FTP-629DSL310" by configuration tool.
FTP-629DSL350 FTP-639USL100 FTP-639USL200	✓	✓			Please select printer model to "FTP-629DSL350" by configuration tool.
FTP-839DSL310	✓	✓			Please select printer model to "FTP-839DSL310" by configuration tool.
FTP-62GDSL000 FTP-62GDSL100 FTP-62GDSL110 FTP-62GDSL120 FTP-62GUSL000 FTP-62GUSL070 FTP-62GUSL100 FTP-63GUSL000 FTP-63GUSL070 FTP-64GDSL130	✓	✓			Please select printer model to "FTP-62GDSL000" by configuration tool.
FTP-63GDSL483	✓	✓			Please select printer model to "FTP-63GDSL483" by configuration tool.
FTP-83GDSL483	✓	✓			Please select printer model to "FTP-83GDSL483" by configuration tool.

					configuration tool.
FTP-62HWSL001	✓		✓		Please select printer model to "FTP-62HWSL001" by configuration tool.
FTP-62HDSL100		✓			Please select printer model to "FTP-62HDSL100" by configuration tool.
FTP-62EDSL200		✓			Please select printer model to "FTP-62EDSL200" by configuration tool.

For RS-232C, please set printer as follows.

Flow control as hardware.

Parity as none.

2.3. Installation media

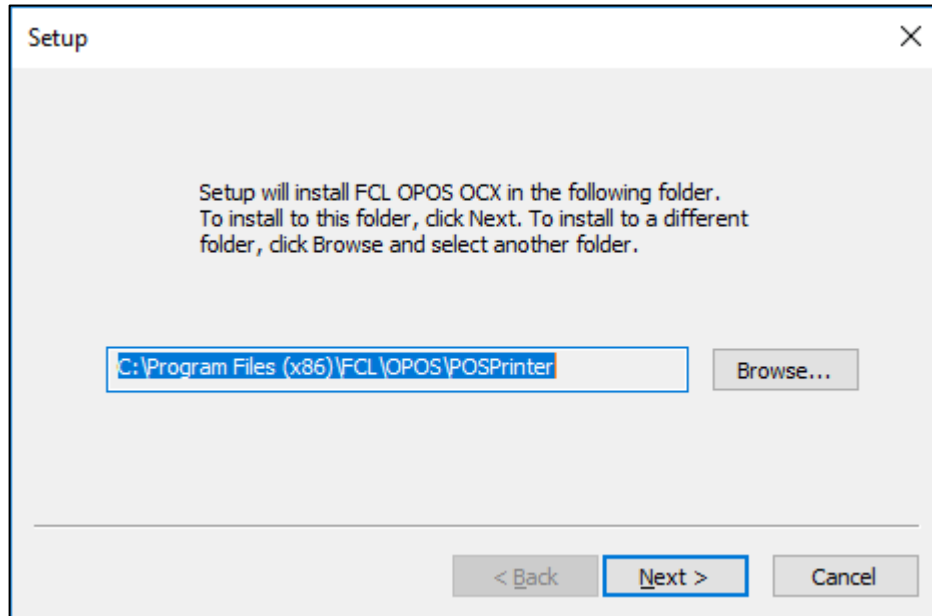
This driver is distributed as a compressed file.

The contents of the uncompressed file are the following.

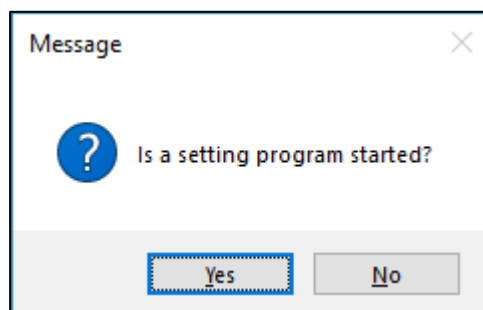
```
FTP2166000EQ_FTP-POS_OPOSVxxx
  OPOSInstaller.exe
  FCLOCXConfig.exe
  FTPOSPrinterCO.ocx
  FTPOSPrinterSO.dll
  AddReg.reg
  AddReg64.reg
  Install.cmd
  Uninstall.cmd
  OPOS_Sample forVB Vxxxx.zip
  OPOS_Sample forVCSharp Vxxxx.zip
  POSforDotNETPluginX64.msi
  \include
```

2.4. Installation procedure

1. Double-click "OPOSInstaller.exe" in "FTP2166000EQ_FTP-POS_OPOSVxxx" folder. If the [User Account Control] dialog box is displayed, click [Yes].
2. The installation is started. The installation folder of default is as follows
32-bits: [System drive]:\Program Files\FCL\OPOS\POSPrinter
64-bits: [System drive]:\Program Files (x86)\FCL\OPOS\POSPrinter



3. Select the installation directory and execute it. When the installation is completed, the following dialog is displayed. Select "yes", and the configuration tool for the setting change of the POS Printer is started.



4. The screen of the configuration tool is the following.

Setting item	Meaning
Printer model	Select printer model
COM port	Set serial port (COM1–COM9) connected with a POS Printer. If you set COM10 or more, input directly. Please do not add prefix ("\\.\").
Connection	Select the connection of POS Printer to POS (PC).
Baud Rate	Set transmission rate (bps).
Printable width	Select according to the printer.
Applicable paper type	Select the paper type from the list. The types that can be specified depend on the printer model. Please refer to Printer Specification
Automatic division	For more information, refer to GS E+n command in the command Specification or product specification of each printer.
Energy	For adjustment, refer to GS E+n command in the command Specification or product specification of each printer.
Log output	Refer to Log file output .

The configuration tool can be executed by the following procedure.

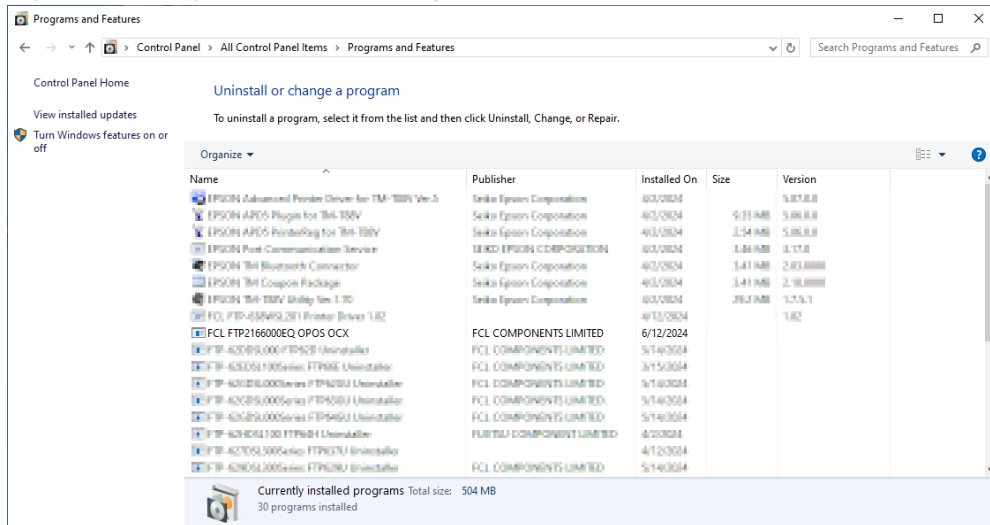
- | | |
|-------------|---|
| Windows 7 | [Start] → [All Programs] → [FCL FTP-POS OPOS OCX]
→ [OCX configuration tool] |
| Windows 8.1 | [Start Screen] → [Apps by category] → [OCX configuration tool] |
| Windows 10 | [Start menu] → [FCL FTP-POS OPOS OCX] → [OCX configuration tool] |
| Windows 11 | [Start menu] → [All apps] → [OCX configuration tool] |

The configuration tool must be executed with administrator privileges.

Uninstallation procedure

1. Run uninstaller.

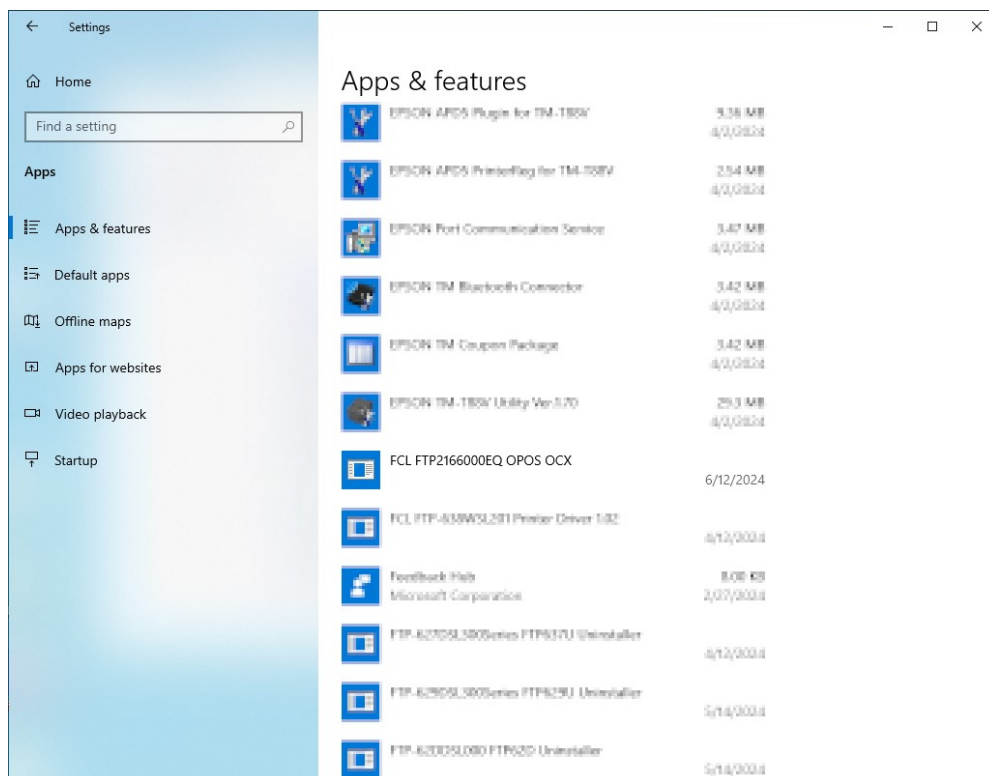
Open the control panel, then launch "Programs and Features".



After that, double-click "FCL FTP2166000EQ OPOS OCX".

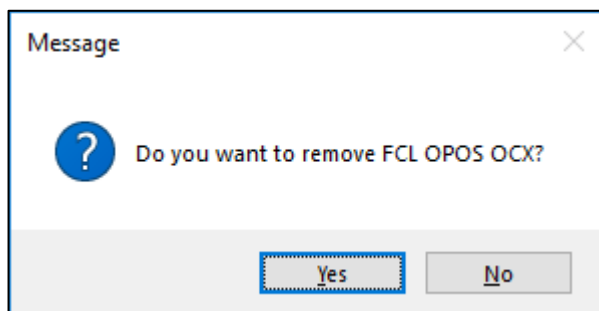
In case of Windows 10, uninstaller can run from “Apps & features” in “Settings” app.

In case of Windows 11, uninstaller can run from “Settings” app -> “Apps” -> “Apps & features”

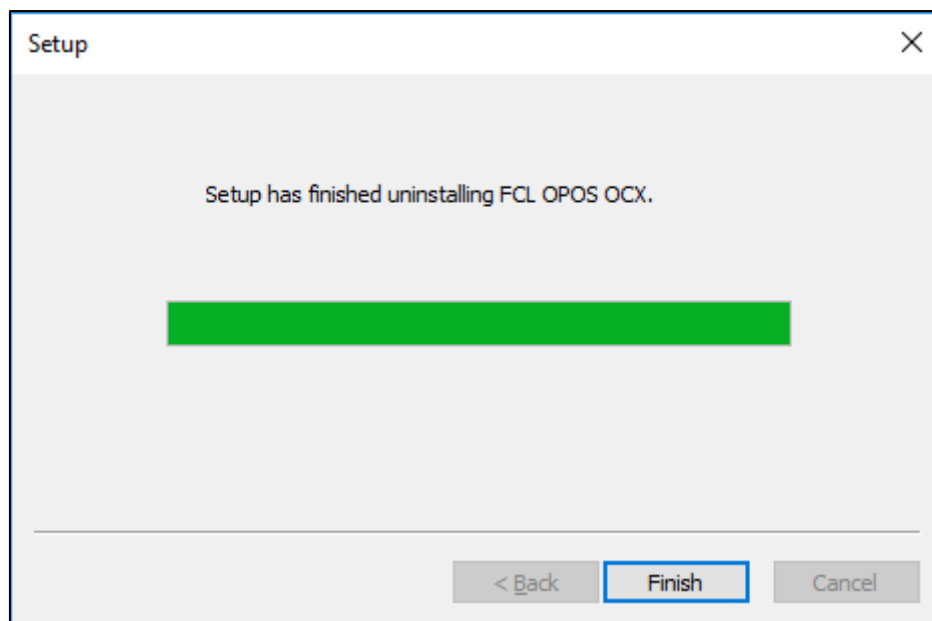


After that, click “FCL FTP2166000EQ OPOS OCX”. Then click “Uninstall” button.

2. The dialog that confirms uninstallation is displayed. Click "Yes".



3. The uninstallation is executed.



This completes uninstallation.

2.5. Installation file list

The file groups of the FTP-POS POSPrinter OPOS control are arranged as follows.

<In case of initial setting on 32-bits OS>

[System drive]:\Program Files\FCL\OPOS\POSPrinter\FTP2166000EQ

FCLOSPrinterCO.ocx	POS Printer Control Object
FCLOSPrinterSO.dll	POS Printer Service Object
FCLOCXConfig.exe	OCX Configuration tool
OPOSInstaller.exe	Uninstaller
Uninstall.cmd	Cmd file for uninstallation

[System drive]:\Users\Public\FCL\OPOS\POSPrinter\FTP2166000EQ\Log
Folder to output log file

[System drive]:\Program Files\FCL\OPOS\POSPrinter\FTP2166000EQ\Include
Include file group defined by OPOS-OCX Control

Opos.h	OPOS General Header File
OposPtr.h	OPOS POS Printer Header File
OPOSEvent.h	OPOS Event Header File
FTPptr.h	OPOS Particular POS Printer Header File
OposAll.bas	OPOS General Definition Module (for Visual Basic)
OposFCL.bas	OPOS Particular POS Printer Definition Module (for Visual Basic)

3. How to use the OPOS Control

3.1. Common

The application uses the OPOS Control according to the following procedures.

1. **Open** method: Called to link the Control Object to the Service Object.
2. **ClaimDevice** method: Called to enable exclusive access to the device. This method is required for the device of exclusive use, and is optional for the device of sharable use.
3. **DeviceEnabled** property: Set to **TRUE** to enable the device.
4. Use the device. (Each Property, Method, Event)
5. **DeviceEnabled** property: Set to **FALSE** to disable the device.
6. **ReleaseDevice** method: Called to release exclusive access to the device.
7. **Close** method: Called to release the Service Object from the Control Object.

Refer to "OPOS APG V1.9" document for details of other usage.

3.2. POS Printer

FTP-POS POS Printer supports only "Receipt". For the methods and properties of other that (Journal and Slip), interface is supplied but operation is not supported.

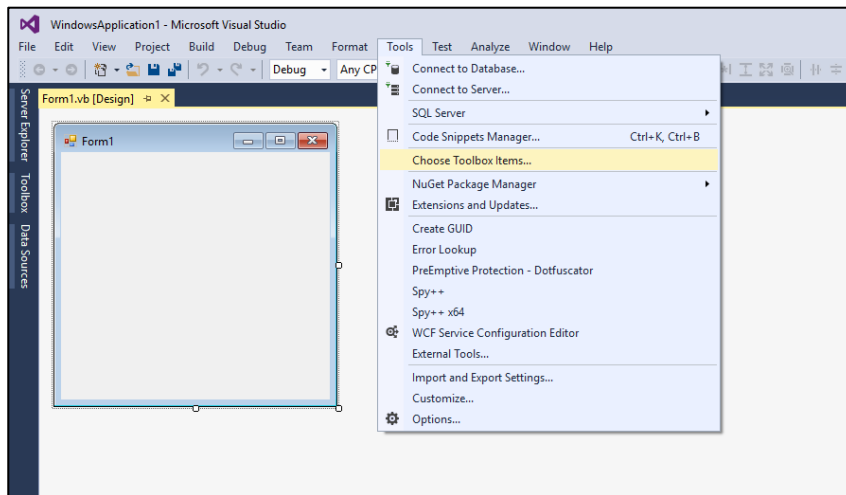
The POS Printer is the device to be used exclusively.

3.3. How to implement the OPOS Control

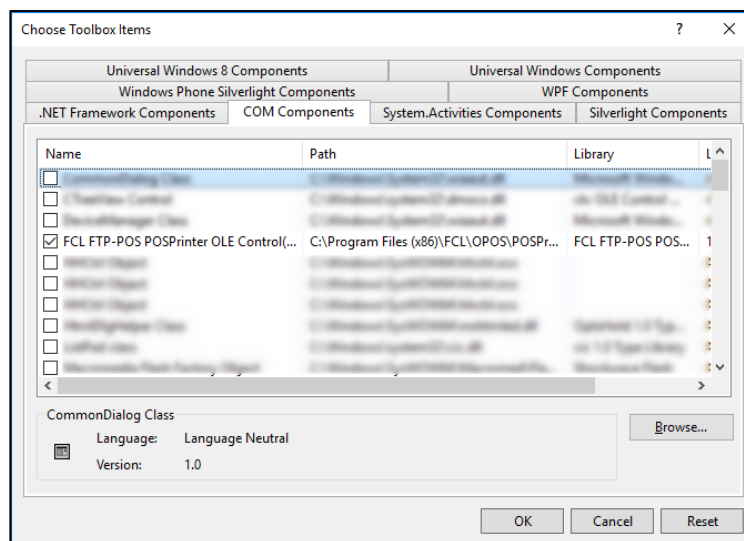
The following shows how to implement the OPOS-OCX Control in the application. This case is for Visual Studio 2015.

3.3.1. To imlement using Visual Basic.

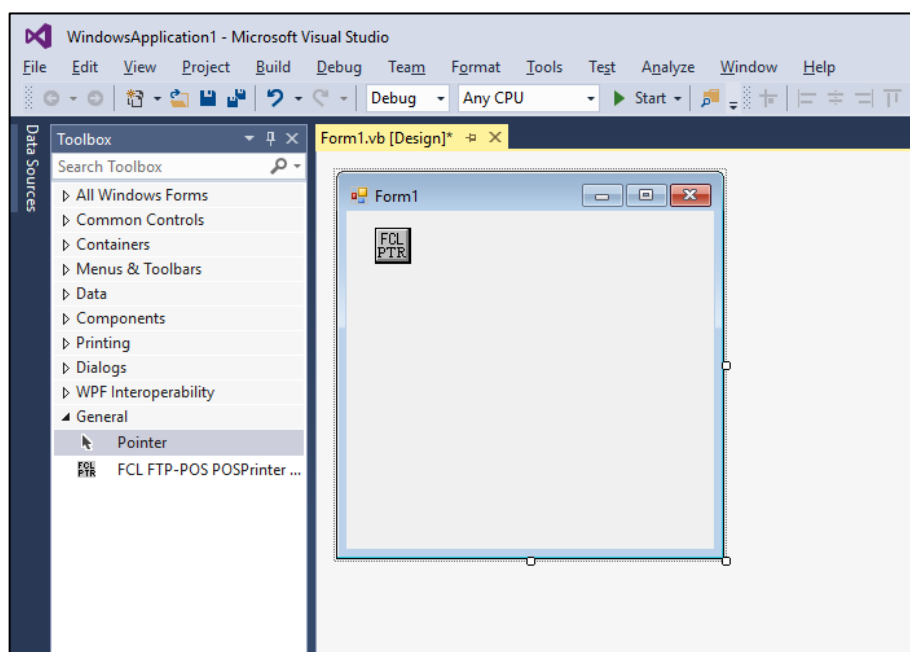
1. Start Visual Basic, and select the kind of project to make. Here is an example of selecting “Windows Forms Application”. Select the tools tab of the menu after making the project, and select “Choose Toolbox Items”.



2. Select the tab of “COM Components” because the dialog is displayed. Put the check in [FCL FTP-POS POSPrinter OLE Control (1.9) module], and click “OK” button.

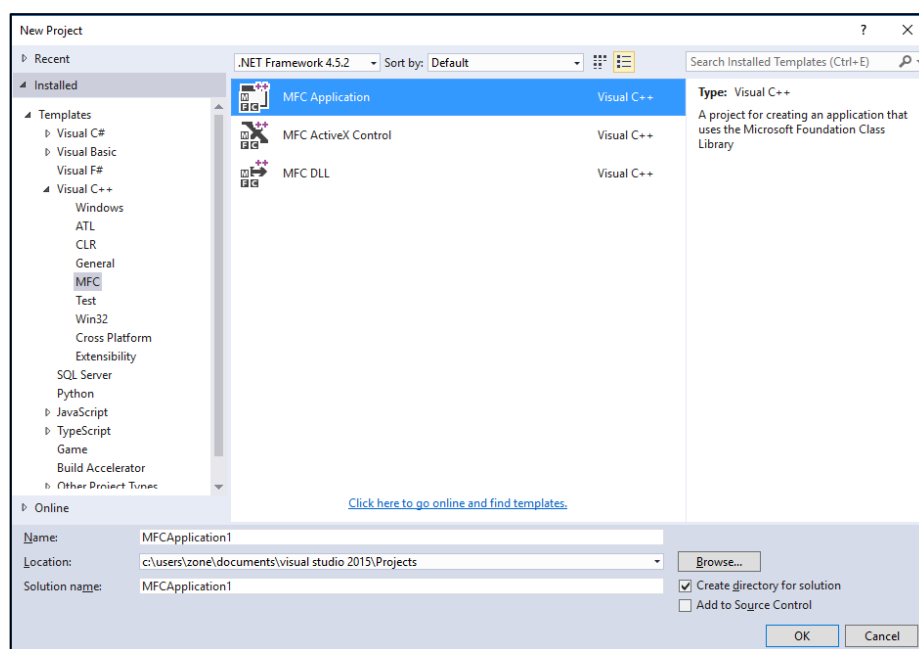


3. When OCX is displayed in tool box, put it on the form. Next, use it as well as usual OCX.

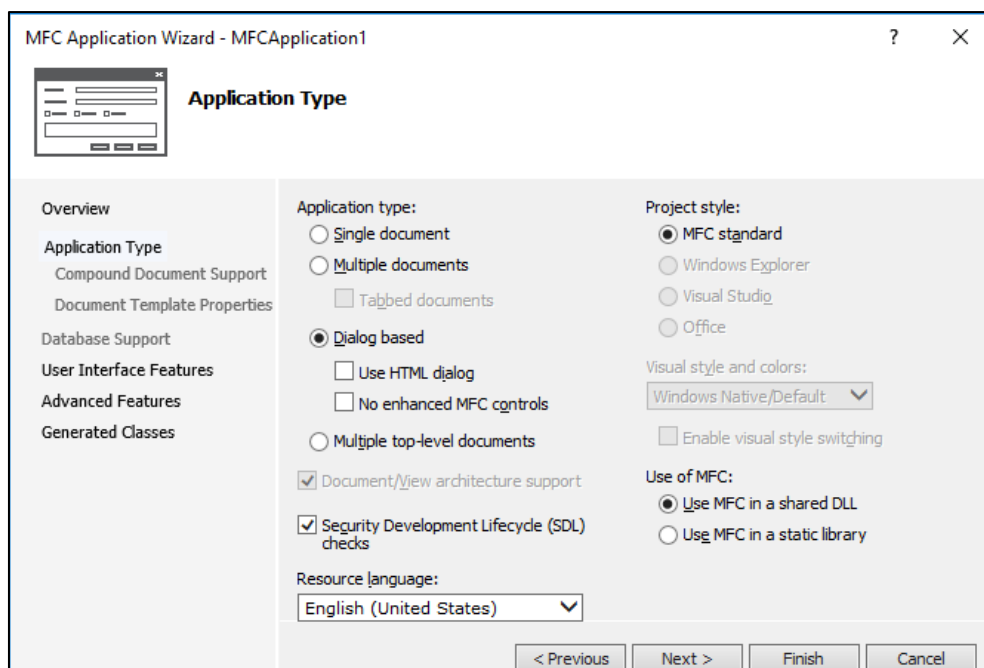


3.3.2. To implement using Visual C++.

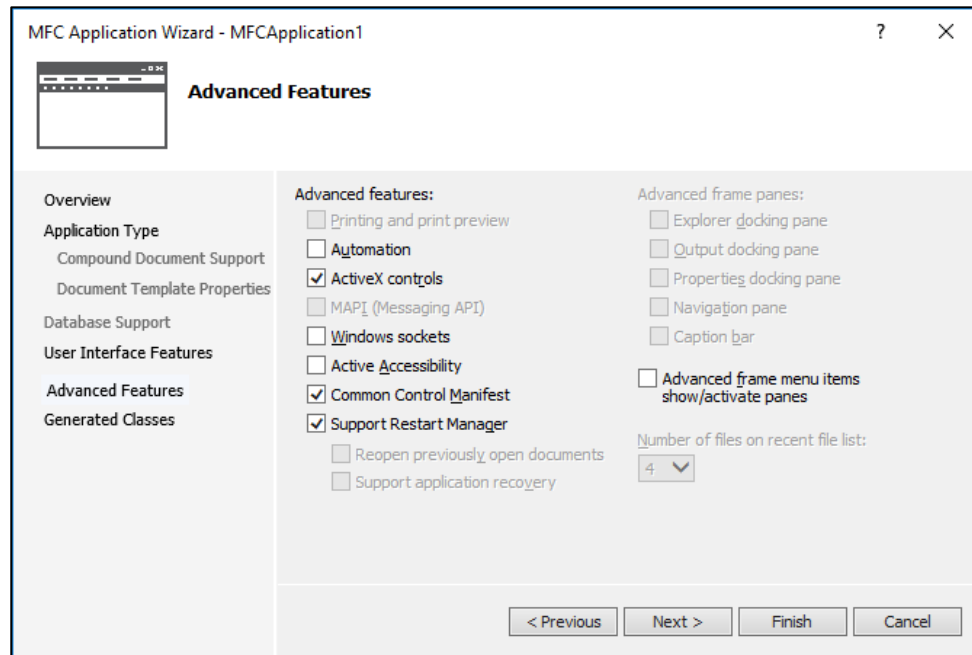
1. Start VisualC++, and select [MFC Application] after a new project is selected. And click [OK].



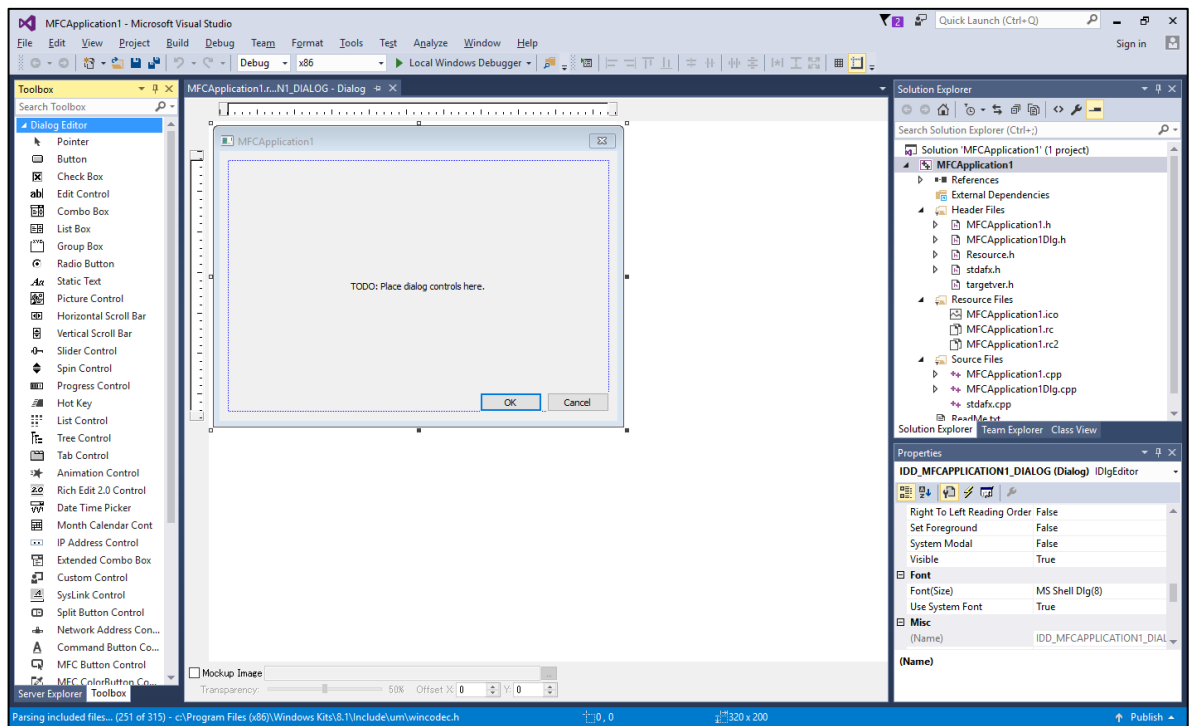
2. On the screen of "Welcome to the MFC Application Wizard", click "Next". On the screen of "Application Type", select "Dialog based" and click "Next".



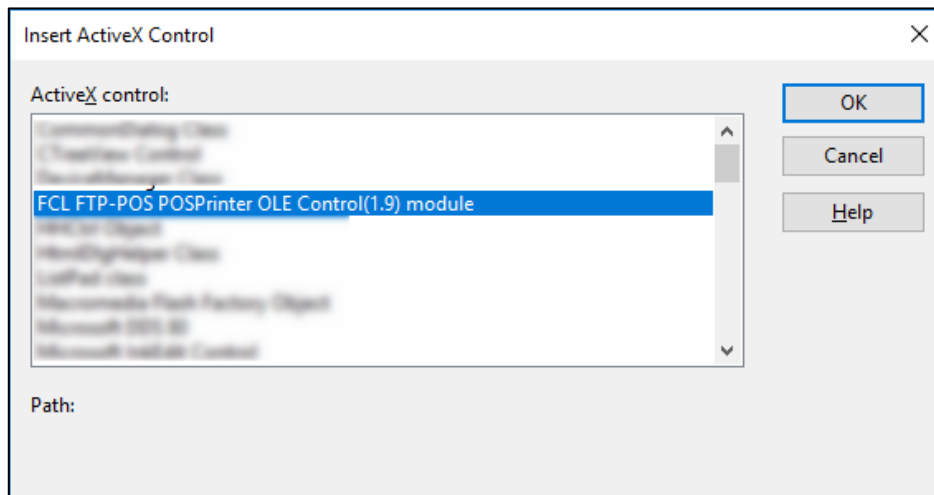
- On the screen of [User Interface Features], after changing the settings of the created project and click [Next]. On the screen of "Advanced Features", select the checkbox of "ActiveX Controls". In other items, change according to the project to be created. Select [next], and set later steps accordingly.



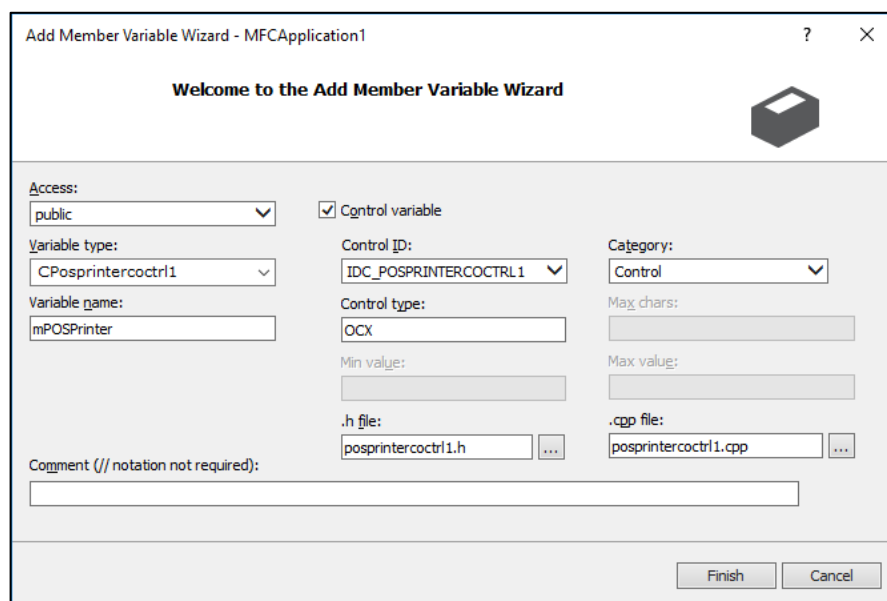
- When the project is made, select the resource view and display the main dialog.



5. Right click on the dialog and display the popup menu. Select "Insert ActiveX Control". When the dialog is displayed, select [FCL FTP-POS POSPrinter OLE Control (1.9) module]. And select [OK].



6. If OCX is inserted in the dialog, right-click on the control. Select "Add Variable" from the pop-up menu. Input arbitrary variable identifier in "Add Member Variable wizard", and click "Finish". When using OCX, you use properties and methods with this member variable.
Usage example: `m_POSPrinter.Open("FTP-POS");`
Delete the corresponding line if error C2664 is generated in the build after the member variable is added.
example : `//, m_POSPrinter(0)`



4. OPOS Interface specifications (Printer)

4.1. Summary

Properties

<i>Common</i>	<i>Type</i>	<i>Access</i>	<i>Initialized After</i>	<i>Condition</i>
BinaryConversion	Long	R/W	Open	OPOS_BC_NONE(0) Made writable after Open.
CapCompareFirmwareVersion	Boolean	R	Open	
CapPowerReporting	Long	R	Open	
CapStatisticsReporting	Boolean	R	Open	
CapUpdateFirmware	Boolean	R	Open	
CapUpdateStatistics	Boolean	R	Open	
CheckHealthText	String	R	Open	
Claimed	Boolean	R	Open	
DeviceEnabled	Boolean	R/W	Open & Claim	
FreezeEvents	Boolean	R/W	Open	Made writable after Open.
OpenResult	Long	R	None	
OutputID	Long	R	Open	
PowerNotify	Long	R/W	Open	Made writable after Open, and unwritable after Enabled.
PowerState	Long	R	Open	
ResultCode	Long	R	--	
ResultCodeExtended	Long	R	Open	
State	Long	R	--	
ControlObjectDescription	String	R	--	
ControlObjectVersion	Long	R	--	
ServiceObjectDescription	String	R	Open	
ServiceObjectVersion	Long	R	Open	
DeviceDescription	String	R	Open	
DeviceName	String	R	Open	

<i>Specific</i>	<i>Type</i>	<i>Access</i>	<i>Initialized After</i>	<i>Condition</i>
CapCharacterSet	Long	R	Open	
CapConcurrentJrnRec	Boolean	R	Open	
CapConcurrentJrnSlp	Boolean	R	Open	
CapConcurrentRecSlp	Boolean	R	Open	
CapConcurrentPageMode	Boolean	R	Open	
CapCoverSensor	Boolean	R	Open	
CapMapCharacterSet	Boolean	R	Open	
CapTransaction	Boolean	R	Open	
CapJrnPresent	Boolean	R	Open	
CapJrn2Color	Boolean	R	Open	

FTP2166000EQ

FTP-POS

POSPrinter OPOS Control Function specification

CapJrnBold	Boolean	R	Open	
CapJrnDhigh	Boolean	R	Open	
CapJrnDwide	Boolean	R	Open	
CapJrnDwideDhigh	Boolean	R	Open	
CapJrnEmptySensor	Boolean	R	Open	
CapJrnItalic	Boolean	R	Open	
CapJrnNearEndSensor	Boolean	R	Open	
CapJrnUnderline	Boolean	R	Open	
CapJrnCartridgeSensor	Long	R	Open	
CapJrnColor	Long	R	Open	
CapRecPresent	Boolean	R	Open	
CapRec2Color	Boolean	R	Open	
CapRecBarCode	Boolean	R	Open	
CapRecBitmap	Boolean	R	Open	
CapRecBold	Boolean	R	Open	
CapRecDhigh	Boolean	R	Open	
CapRecDwide	Boolean	R	Open	
CapRecDwideDhigh	Boolean	R	Open	
CapRecEmptySensor	Boolean	R	Open	
CapRecItalic	Boolean	R	Open	
CapRecLeft90	Boolean	R	Open	
CapRecNearEndSensor	Boolean	R	Open	
CapRecPapercut	Boolean	R	Open	
CapRecPageMode	Boolean	R	Open	
CapRecRight90	Boolean	R	Open	
CapRecRotate180	Boolean	R	Open	
CapRecStamp	Boolean	R	Open	
CapRecUnderline	Boolean	R	Open	
CapRecCartridgeSensor	Long	R	Open	
CapRecColor	Long	R	Open	
CapRecMarkFeed	Long	R	Open	
CapSlpPresent	Boolean	R	Open	
CapSlpFullslip	Boolean	R	Open	
CapSlp2Color	Boolean	R	Open	
CapSlpBarCode	Boolean	R	Open	
CapSlpBitmap	Boolean	R	Open	
CapSlpBold	Boolean	R	Open	
CapSlpDhigh	Boolean	R	Open	
CapSlpDwide	Boolean	R	Open	
CapSlpDwideDhigh	Boolean	R	Open	
CapSlpEmptySensor	Boolean	R	Open	
CapSlpItalic	Boolean	R	Open	
CapSlpLeft90	Boolean	R	Open	
CapSlpNearEndSensor	Boolean	R	Open	
CapSlpPageMode	Boolean	R	Open	
CapSlpRight90	Boolean	R	Open	
CapSlpRotate180	Boolean	R	Open	
CapSlpUnderline	Boolean	R	Open	

CapSlpBothSidesPrint	Boolean	R	Open	
CapSlpCartridgeSensor	Long	R	Open	
CapSlpColor	Long	R	Open	
AsyncMode	Boolean	R/W	Open	Made writable after Enabled.
CartridgeNotify	Long	R/W	Open	Unwritable
CharacterSet	Long	R/W	Open,Claim & Enable	Made writable after Enabled.
CharacterSetList	String	R	Open	
CoverOpen	Boolean	R	Open,Claim & Enable	
ErrorLevel	Long	R	Open	
ErrorStation	Long	R	Open	
ErrorString	String	R	Open	
FontTypefaceList	String	R	Open	
FlagWhenIdle	Boolean	R/W	Open	Made writable after Enabled
MapCharacterSet	Boolean	R/W	Open	
MapMode	Long	R/W	Open	Made writable after Open.
PageModeArea	String	R	Open	
PageModeDescriptor	Long	R	Open,Claim & Enable	
PageModeHorizontalPosition	Long	R/W	Open,Claim & Enable	Unwritable.
PageModePrintArea	String	R/W	Open,Claim & Enable	Unwritable
PageModePrintDirection	Long	R/W	Open,Claim & Enable	Unwritable.
PageModeStation	Long	R/W	Open	Unwritable
PageModeVerticalPosition	Long	R/W	Open,Claim & Enable	Unwritable
RotateSpecial	Long	R/W	Open	Made writable after Open.
JrnLineChars	Long	R/W	Open,Claim & Enable	Unwritable
JrnLineCharsList	String	R	Open	
JrnLineHeight	Long	R/W	Open,Claim & Enable	Unwritable
JrnLineSpacing	Long	R/W	Open,Claim & Enable	Unwritable
JrnLineWidth	Long	R	Open,Claim & Enable	
JrnLetterQuality	Boolean	R/W	Open,Claim & Enable	
JrnEmpty	Boolean	R	Open,Claim & Enable	
JrnNearEnd	Boolean	R	Open,Claim & Enable	

JrnCartridgeState	Long	R	Open,Claim & Enable	
JrnCurrentCartridge	Long	R/W	Open,Claim & Enable	Unwritable
RecLineChars	Long	R/W	Open,Claim & Enable	Made writable after Open.
RecLineCharsList	String	R	Open	
RecLineHeight	Long	R/W	Open,Claim & Enable	Unwritable
RecLineSpacing	Long	R/W	Open,Claim & Enable	Made writable after Open.
RecLineWidth	Long	R	Open,Claim & Enable	
RecLetterQuality	Boolean	R/W	Open,Claim & Enable	Made writable after Open.
RecEmpty	Boolean	R	Open,Claim & Enable	
RecNearEnd	Boolean	R	Open,Claim & Enable	
RecSidewaysMaxLines	Long	R	Open,Claim & Enable	
RecSidewaysMaxChars	Long	R	Open,Claim & Enable	
RecLinesToPaperCut	Long	R	Open,Claim & Enable	
RecBarCodeRotationList	String	R	Open	
RecBitmapRotationList	String	R	Open	
RecCartridgeState	Long	R	Open,Claim & Enable	
RecCurrentCartridge	Long	R/W	Open,Claim & Enable	Unwritable
SlpLineChars	Long	R/W	Open,Claim & Enable	Unwritable
SlpLineCharsList	String	R	Open	
SlpLineHeight	Long	R/W	Open,Claim & Enable	Unwritable
SlpLineSpacing	Long	R/W	Open,Claim & Enable	Unwritable
SlpLineWidth	Long	R	Open,Claim & Enable	
SlpLetterQuality	Boolean	R/W	Open,Claim & Enable	Unwritable
SlpEmpty	Boolean	R	Open,Claim & Enable	
SlpNearEnd	Boolean	R	Open,Claim & Enable	
SlpSidewaysMaxLines	Long	R	Open,Claim & Enable	
SlpSidewaysMaxChars	Long	R	Open,Claim &	

			Enable	
SlpMaxLines	Long	R	Open,Claim & Enable	
SlpLinesNearEndToEnd	Long	R	Open,Claim & Enable	
SlpBarCodeRotationList	String	R	Open	
SlpBitmapRotationList	String	R	Open	
SlpPrintSide	Long	R	Open,Claim & Enable	
SlpCartridgeState	Long	R	Open,Claim & Enable	
SlpCurrentCartridge	Long	R/W	Open,Claim & Enable	Unwritable

* Access 'R' indicates Read-only, 'R/W' indicates writable.

“*Initialized After*” is the method and property required for initialization.

‘Open’ indicates the Open method, and ‘Claim’ indicates the ClaimDevice method, and “Enable” indicates DeviceEnabled properties is set to TRUE.

If required procedure for initializing is not performed, the error may be set in the ResultCode property.

When “*Initialized After*” is ‘Open&Claim’ or ‘Open,Claim&Enable’, the property is available after Open method, but the value may be initialized after Open&Claim&Enabled, access it after conditions are met.

Methods

Common	Necessary condition
Open	--
Close	Open
ClaimDevice	Open
ReleaseDevice	Open, Claim
ClearOutput	Open, Claim & Enable *1
CheckHealth	Open, Claim & Enable
CompareFirmWareVersion	Open, Claim & Enable
UpdateFirmware	Open, Claim & Enable
ResetStatistics	Open, Claim & Enable
RetrieveStatistics	Open, Claim & Enable
UpdateStatistics	Open, Claim & Enable
DirectIO	Open, Claim & Enable *1

Specific	Necessary condition
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

Events

Name	Necessary condition
DirectIOEvent	Open, Claim & Enable *1
ErrorEvent	Open, Claim & Enable
OutputCompleteEvent	Open, Claim & Enable
StatusUpdateEvent	Open, Claim & Enable

*1... The necessary conditions are different from OLE for Retail POS Application Programmer's Guide Version 1.9.

4.2. Data Characters and Escape Sequences

This OCX supports the following Escape Sequences.

(1) Perform indicated action.

Name	Data	Remarks
Paper cut	ESC #P	<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.</p> <p>If '#' is '1' to '99', a partial cut is performed. If '#' is '100' or is omitted, a full cut is performed.</p> <p>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.</p> <p>When a rotated 90° left or right mode by RotatePrint method, this is not supported.</p> <p>The correspondence depends on the printer model.</p> <p>Please refer to Printer Specification.</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.</p> <p>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.</p> <p>When a rotated 90° left or right mode by RotatePrint method, this is not supported.</p> <p>The correspondence depends on the printer model.</p> <p>Please refer to Printer Specification.</p>
Feed, Paper cut, and Stamp	ESC #sP	Not supported.
Print bitmap	ESC #B	<p>Prints the bitmap saved by SetBitmap method. The character '#' is replaced by the bitmap number, and supports the values from '1' to '255'.</p> <p>When a rotated 90° left or right mode by RotatePrint method, this is not supported.</p> <p>The correspondence depends on the printer model.</p> <p>Please refer to Printer Specification.</p>
Print top logo	ESC tL	<p>Prints the top logo saved by SetLogo method.</p> <p>When a rotated 90° left or right mode by RotatePrint method, this is not supported.</p>
Print bottom logo	ESC bL	<p>Prints the bottom logo saved by SetLogo method.</p> <p>When a rotated 90° left or right mode by RotatePrint method, this is not supported.</p>
Fire stamp	ESC sL	Not supported.
Feed lines	ESC #fF	<p>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.</p> <p>The character '#' supports the values from '1' to '255'.</p> <p>When print data is not buffered, specified lines are fed.</p> <p>When print data is buffered, buffered data is printed first, then specified lines are fed.</p>

FTP2166000EQ

FTP-POS

POSPrinter OPOS Control Function specification

Feed units	ESC #uF	<p>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.</p> <p>The character '#' supports the values from '1' to '255' which is converted with PTR_MM_DOTS(1) unit.</p> <p>The length of feed units is not affected by the length of feed line that was set in the printer.</p>
Feed reverse	ESC #rF	<p>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.</p> <p>The character '#' supports the values from '1' to '255'. When print data is not buffered, specified lines are fed backward.</p> <p>When print data is buffered, buffered data is printed first, then specified lines are fed backward.</p>

(2) Print Mode

Characteristics that are remembered until explicitly changed.

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

(3) 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 .
Italic	ESC iC	Not supported.
Alternate color (Custom)	ESC #rC	Not supported.
Alternate color (Custom color Red)	ESC rC	Not supported.
Reverse video	ESC rvC	Prints in a reverse video format. The correspondence depends on the printer model. Please refer to Printer Specification .
Shading	ESC #sC	Not supported.
Single high and wide	ESC 1C	Prints normal size.
Double wide	ESC 2C	Prints double-wide characters.
Double high	ESC 3C	Prints double-high characters.
Double high and wide	ESC 4C	Prints double-high/double-wide characters.
Scale horizontally	ESC #hC	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 .
RGB Color	ESC #fC	Not supported.
Center	ESC cA	Aligns following text in the center. When not specified at the head of line, it may be invalid. 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 not specified at the head of line, it may be invalid. 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 .

FTP2166000EQ

FTP-POS

POSPrinter OPOS Control Function specification

Normal	ESC N	Restores printer characteristics to normal condition. When not specified at the head of line, it may be invalid.
SubScript	ESC tbC	Not supported.
SuperScript	ESC tpC	Not supported.
Pass through embedded data	ESC #E	Send the following “#E” characters of data through to the hardware without modifying it. The character '#' is replaced by an ASCII decimal string telling the number of bytes following the escape sequence that should be passed through as-is to the hardware. If BinaryConversion property is OPOS_BC_NONE, it may not be able to be sent correctly. It is recommended to use OPOS_BC_NIBBLE or OPOS_BC_DECIMAL.

4.3. Common Properties

This section describes the Common Properties provided for POS Printer.

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.

The return values with special meanings are only described. About the errors when accessing without satisfying the initial condition, see the ResultCode property.

BinaryConversion Property R/W

Syntax

LONG BinaryConversion;

Remarks

OPOS passes multi-character input and output using BStrings. BStrings may be safely used for text data. As the BStrings are passed between the application and the OPOS Control, OLE may perform language-specific translations to or from Unicode

When BStrings are used to pass binary data, then these translations may alter the data such that the data byte in a BString character at the application does not match the corresponding byte at the Control. This mismatch is more likely when BString pointers are used, since the Unicode characters are presented to the application and/or Control, and a language difference between them may cause misinterpretation.

Characters between 0x00 and 0x7F may be sent without fear of language-specific translation. Only characters between 0x80 and 0xFF sometimes cause incorrect translations.

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

In the OPOS environment, the format of this data depends upon the value of the **BinaryConversion** property. See **BinaryConversion** property.

The binary conversion values are:

Value	Meaning
OPOS_BC_NONE(0)	Data is placed one byte per BString character, with no conversion.(This is the default.)
OPOS_BC_NIBBLE(1)	Each byte is converted into two characters. (This option provides for the fastest conversion between binary and ASCII 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 0x39 0x3A (= the string "9:"). Note that this conversion is not the more common hexadecimal ASCII, which would have converted 154 to 0x39 0x41 (= the string "9A").
OPOS_BC_DECIMAL(2)	Each byte is converted into three characters. (This option provides for the easiest conversion between binary and ASCII characters for Visual Basic and similar languages.) VAL(<i>string</i>) may be used on each 3 characters to convert from ASCII to binary. RIGHT("^^"+STR(<i>byte</i>), 3) may be used to produce 3 ASCII characters from each byte, where '^' represents the space character. Example 1: Byte value 154 = 0x9A becomes the characters 0x31 0x35 0x34 (= the string "154").

Example 2: Byte value 8 becomes the characters 0x30 0x30 0x38 (= the string “008”).

When **BinaryConversion** is on (that is, not OPOS_BC_NONE) and the property or method parameter description specifies that **BinaryConversion** applies, then the application has the following responsibilities:

- Before setting the property or passing the method parameter, convert the string data into the format specified by the **BinaryConversion** value.
- After getting the property or receiving the method parameter, convert the string data from the format specified by the **BinaryConversion** value.

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

Return

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

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

CapCompareFirmwareVersion Property

Syntax

BOOL CapCompareFirmwareVersion;

Remarks

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 by the **Open** method.

CapPowerReporting Property

Syntax

LONG CapPowerReporting;

Remarks

Identifies the reporting capabilities of the device.

The power reporting values are:

Value	Meaning
OPOS_PR_STANDARD(1)	The Service Object can determine and report two of the power states. (OFF_OFFLINE and ONLINE)

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

CapUpdateFirmware Property

Syntax

BOOL CapUpdateFirmware;

Remarks

If **TRUE**, then the device's firmware can be updated via the **UpdateFirmware** method.
This property is initialized by the **Open** method.

CheckHealthText Property

Syntax

BSTR CheckHealthText;

Remarks

Holds the results of the most recent call to the **CheckHealth** method. The following examples illustrate some possible diagnoses:

- Internal Success: "Internal HCheck: Successful",
 Fail: "Internal HCheck: OFF/OFFLINE"
- External Success : "External HCheck: Successful"
 Fail: "External HCheck: Failure"
- Interactive Complete: "Interactive HCheck: Complete"

This property is empty ("") before the first call to the **checkHealth** method.

Claimed Property

Syntax

BOOL Claimed;

Remarks

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

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

DeviceEnabled Property R/W

Syntax

BOOL 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 property to **TRUE** before using output devices.
The Device's power state (**PowerReporting**) may be reported while **DeviceEnabled** is **TRUE**.
This property is initialized to **FALSE** by the **Open** method.

Return

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

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

OPOS_E_NOTCLAIMED(103)	An exclusive use device must be claimed before the device may be enabled.
OPOS_E_BUSY(113)	Cannot perform operation in progress. Must wait until output is no longer in progress.
Other Values	See ResultCode property.

FreezeEvents Property R/W

Syntax

BOOL FreezeEvents;

Remarks

When **TRUE**, the application has requested that the Control not fire events. Events will be held by the Control 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.

When **ErrorEvent** is frozen, **State** property is set to **OPOS_S_BUSY(3)**. In this condition, the Application cannot close the Control. Therefore, frozen events are discard by **ClearOutput** method or changing **FreezeEvents** to **FALSE** will cause **ErrorEvent** to be fired, then call **Close** method.

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

Return

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

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

OpenResult Property

Syntax

LONG OpenResult;

Remarks

Holds additional details about the most recent **Open** method. The **OpenResult** values are:

Value	Meaning
OPOS_SUCCESS(0)	Successful open.
OPOS_OR_ALREADYOPEN(301)	Control already open.
OPOS_OR_REGBADNAME(302)	The registry does not contain a key for the specified device name. Or device name is not specified.
OPOS_OR_REGPROGID(303)	Could not read the device name key's default value, or could not convert the Programmatic ID it holds into a valid Class ID.
OPOS_OR_CREATE(304)	Could not create a service object instance, or could not get its IDispatch interface.
OPOS_OR_BADIF(305)	The service object does not support one or more of the methods required by its release. The device name may be different from service object.

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

OutputID Property

Syntax

LONG OutputID;

Remarks

Holds the identifier of the most recently started asynchronous output.

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

The output ID numbers are assigned cyclically between 1 – 99999.

PowerNotify Property R/W

Syntax

LONG PowerNotify;

Remarks

Contains the type power notification selection made by the Application.

The power notification values are:

Value	Meaning
OPOS_PN_DISABLED(0)	The Control will not provide any power notifications to the application. No power notification StatusUpdateEvents will be fired, and PowerState may not be set. (Default value)
OPOS_PN_ENABLED(1)	The Control will fire power notification StatusUpdateEvents 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**.

Return

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

Value	Meaning
OPOS_SUCCESS(0)	The property was set successfully.
OPOS_E_ILLEGAL(106)	One of the following occurred. <ul style="list-style-type: none">• The device is already enabled.• An invalid parameter value was specified.
Other Values	See ResultCode property.

PowerState Property

Syntax

LONG PowerState;

Remarks

Contains the current power condition, if **PowerNotify** is set to **OPOS_PN_ENABLED(1)**.

The power reporting values are:

Value	Meaning
OPOS_PS_UNKNOWN(2000)	Cannot determine the device's power state, for one of the following reasons.(default) PowerNotify = OPOS_PN_DISABLED(0) . Power notifications are disabled. DeviceEnabled = FALSE . Power state monitoring does not occur until the device is enabled.
OPOS_PS_ONLINE(2001)	The device is powered on and ready for use.
OPOS_PS_OFF_OFFLINE(2004)	The device is off, offline or not connected. About a restoration method, see OPOS_E_NOHARDWARE(107) in ResultCode property. The condition reported depends on the printer model and the interface. Please refer to Printer Specification .

ResultCode Property

Syntax

LONG ResultCode;

Remarks

This property is set by each method. It is also set when a writable property is set.

This property is always readable. Before the **Open** method is called, it returns the value **OPOS_E_CLOSED(101)**.

The result code values are:

Value	Meaning
OPOS_SUCCESS(0)	Successful operation.
OPOS_E_CLOSED(101)	Attempt was made to access a closed device.
OPOS_E_NOTCLAIMED(103)	Attempt was made to access an exclusive-use device that must be claimed before the method or property set action can be used.
OPOS_E_NOSERVICE(104)	The Control cannot communicate with the Service Object. Most likely, a setup or configuration error must be corrected.
OPOS_E_DISABLED(105)	Cannot perform operation while device is disabled.
OPOS_E_ILLEGAL(106)	Attempt was made to perform an illegal or unsupported operation with the device, or an invalid parameter value was used.
OPOS_E_NOHARDWARE(107)	The OPOS Printer is not powered on, or off-line.
OPOS_E_FAILURE(111)	The device cannot perform the requested procedure, even though the device is connected to the system, powered on, and on-line.
OPOS_E_TIMEOUT(112)	The Service Object timed out waiting for a response from the device, or the Control timed out waiting for a response from the Service Object.
OPOS_E_BUSY(113)	The current Service Object state does not allow this request. For example, if asynchronous output is in progress, certain methods may not be allowed.
OPOS_E_EXTENDED(114)	A class-specific error condition occurred. The error condition code is available in the ResultCodeExtended property.

FTP2166000EQ

FTP-POS

POSPrinter OPOS Control Function specification

ResultCodeExtended Property

Syntax

LONG ResultCodeExtended;

Remarks

When the **ResultCode** is set to **OPOS_E_EXTENDED(114)**, this property is set to a class-specific value, matching one of the values given in the device class documentation.

When the **ResultCode** is set to any other value, this property may be set by the Service Object to an SO-specific value. These values are only meaningful if the application adds Service Object-specific code to handle them.

The ResultCodeExtended values are:

ResultCodeExtended	Constant	Details
201	OPOS_EPTR_COVER_OPEN	The printer cover is opened. To recover from error, close the printer cover. In this case, it is not necessary to disable, ReleaseDevice , and Close .
203	OPOS_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, ReleaseDevice and Close .
206	OPOS_EPTR_TOOBIG	The bitmap is either too wide to print without transformation, or it is too big to transform
207	OPOS_EPTR_BADFORMAT	The specified file is either not a bitmap file, or it is in an unsupported format.
SO-specific error 10001	OPOS_FCL_EPTR_POWERSUPPLY	Printer power-supply voltage error is occurred. This error is occurred when printing method in progress, which is not through the firing of the StatusUpdateEvent . Therefore no way to recover, close the OCX.
SO-specific error 10002	OPOS_FCL_EPTR_DATA	The transmission data are abnormal. This error is occurred when printing method in progress, which is not through the firing of the StatusUpdateEvent . If not recovering after a retry, close the OCX.
SO-specific error 10003	OPOS_FCL_EPTR_CUTTER	The cutter error is occurred. This error is occurred when printing method in progress, which is not through the firing of the

FTP2166000EQ

FTP-POS

POSPrinter OPOS Control Function specification

		<p>StatusUpdateEvent.</p> <p>The printer cannot normally cut because something stiff may be inserted to cutter.</p> <p>To recover from error, remove it. In this case, it is not necessary to disable, ReleaseDevice and Close.</p>
SO-specific error 10004	OPOS_FCL_EPTR_HARDWARE	<p>The hardware error is occurred.</p> <p>This error is occurred when printing method in progress, which is not through the firing of the StatusUpdateEvent.</p> <p>Therefore no way to recover, close the OCX.</p>
SO-specific error 10005	OPOS_FCL_EPTR_HEADHOT	<p>The abnormal head temperature is occurred.</p> <p>This error is occurred when printing method in progress, which is not through the firing of the StatusUpdateEvent.</p> <p>Possible to resume when the head temperature goes down. In this case, it is not necessary to disable, ReleaseDevice and Close.</p>
SO-specific error 10006	OPOS_FCL_EPTR_MARK	<p>The mark can not be scanned.</p> <p>This error is occurred when MarkFeed method is performed.</p> <p>There is a possibility of paper jam.</p> <p>To recover from error, remove it.</p> <p>In this case, it is not necessary to disable, ReleaseDevice and Close.</p>
SO-specific error 10007	OPOS_FCL_EPTR_PRESENTER	<p>The presenter error is occurred.</p> <p>This error is occurred when printing method in progress, which is not through the firing of the StatusUpdateEvent.</p> <p>It is possible that the presenter can not feed such as paper jam.</p> <p>To recover from error, remove it. In this case, it is not necessary to disable, ReleaseDevice and Close.</p>

State Property

Syntax

LONG State;

Remarks

Contains the current state of the Control.

Value	Meaning
OPOS_S_CLOSED(1)	The Control is closed. (Default)
OPOS_S_IDLE(2)	The Control is in a good state and is not busy.
OPOS_S_BUSY(3)	The Control is in a good state and is busy performing output.
OPOS_S_ERROR(4)	An error has been reported, and the application must recover the Control to a good state before normal I/O can resume. This state can be set in ErrorEvent hanler.

This property is always readable.

ControlObjectDescription Property

Syntax

BSTR ControlObjectDescription;

Remarks

The property identifies the Control Object.

This property is always readable.

ControlObjectVersion Property

Syntax

LONG ControlObjectVersion;

Remarks

Control Object version number.

This property is always readable.

ServiceObjectDescription Property

Syntax

BSTR ServiceObjectDescription;

Remarks

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

ServiceObjectVersion Property

Syntax

LONG ServiceObjectVersion;

Remarks

Service object version number.
This property is initialized by the **Open** method.

DeviceDescription Property

Syntax

BSTR DeviceDescription;

Remarks

The property identifies the device and any pertinent information about it.
This property is initialized by the **Open** method.

DeviceName Property

Syntax

BSTR DeviceName;

Remarks

The property identifies the device and any pertinent information about it.
This property is initialized by the **Open** method.

* The following common properties are not supported.

BOOL CapStatisticsReporting;
BOOL CapUpdateStatistics;

4.4. Common Methods

CheckHealth Method

Syntax

LONG CheckHealth (LONG *Level*);

The *Level*/parameter indicates the type of health check to be performed on the device.

The following values may be specified.

Value	Meaning
OPOS_CH_INTERNAL(1)	Perform a on-line check. A result is set to CheckHealthText property as follows. POS Printer is connected and is power-ON = "Internal HCheck: Successful" POS Printer is not connected or is power-OFF = "Internal HCheck: OFF/OFFLINE"
OPOS_CH_EXTERNAL(2)	Perform a test print. A result is set to CheckHealthText property as follows: Test print was succeeded ="External HCheck: Successful" Test print was failed ="External HCheck: Failure"
OPOS_CH_INTERACTIVE(3)	Display a UI dialog and perform a print test. A result is set to CheckHealthText property as follows: "Interactive HCheck: Complete"

Remarks

Called to test the state of a device.

A text description of the results of this method is placed in the **CheckHealthText** property.

The **CheckHealth** method is always synchronous.

If connecting via RS-232C and *Level* is OPOS_CH_INTERNAL, also set **CheckHealthText** property to "Internal HCheck: OFF/OFFLINE" when offline.

In the case of Bluetooth connection, when *Level* is OPOS_CH_INTERNAL is not supported.

Return

The following value is returned by the method and placed in the **ResultCode** property.

Value	Meaning
OPOS_SUCCESS(0)	Indicates that the health checking procedure was initiated properly and, when possible to determine, indicates that the device is healthy. However, the health of many devices can only be determined by a visual inspection of the test results.
OPOS_E_ILLEGAL(106)	The specified health check level is not supported by the Service Object.
OPOS_E_NOHARDWARE(107)	When perform with OPOS_CH_INTERNAL(1), a result is off or offline.
OPOS_E_BUSY(113)	Cannot perform while output is in progress.
Other Values	See ResultCode .

ClaimDevice Method

Syntax

LONG ClaimDevice (LONG *Timeout*);

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

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

If **OPOS_FOREVER(-1)**, 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 **ClaimDevice** 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, fixed data is requested and the **ClaimDevice** method is normally completed.

Return

The following value is returned by the method and placed in the **ResultCode** property.

Value	Meaning
OPOS_SUCCESS(0)	Exclusive access has been granted. The Claimed property is now TRUE . Also returned if this application has already claimed the device.
OPOS_E_ILLEGAL(106)	An invalid <i>Timeout</i> parameter was specified. Or a printer is not connected, or the errors occurred in SO, could not be processed.
OPOS_E_NOHARDWARE(107)	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.
OPOS_E_FAILURE(111)	Could not establish a connection to POS Printer. The communication setting of COM port is wrong, or another application may be already using COM port.
OPOS_E_TIMEOUT(112)	Another application has exclusive access to the device, and did not relinquish control before <i>Timeout</i> milliseconds expired. Or POS printer was not available condition before <i>Timeout</i> milliseconds expired.
Other values	See ResultCode .

ClearOutput Method

Syntax

```
LONG ClearOutput ();
```

Remarks

Called to clear all buffered output data, including all asynchronous output (**PrintNormal**, **CutPaper**, **RotatePrint**, **PrintBarCode**, **PrintBitmap**, **TransactionPrint** method).

The rotation mode by **RotatePrint** method and the transaction mode by **TransactionPrint** method are canceled.

Any output error events that were pending – usually waiting for **FreezeEvents** to be set to **FALSE** – are also cleared.

Return

The following value is returned by the method and placed in the **ResultCode** property.

Value	Meaning
OPOS_SUCCESS(0)	Output has been cleared.
Other values	See ResultCode .

Close Method

Syntax

LONG Close ();

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.).

Return

The following value is returned by the method and placed in the **ResultCode** property.

Value	Meaning
OPOS_SUCCESS(0)	Device has been disabled and closed.
OPOS_E_BUSY(113)	Cannot perform while asynchronous output is in progress.
Oher values	See ResultCode .

CompareFirmWareVersion Method

Syntax

LONG CompareFirmWareVersion(**BSTR** *FirmWareFileName*, **LONG*** *pResult*);

Value	Meaning
<i>FirmWareFileName</i>	Specifies the data file name with full path whose versions are to be compared against those of the device.
<i>pResult</i>	Location in which to return the result of the comparison.

Remarks

This method determines whether the version of the firmware contained in the specified file is newer, older, or the same as the version of the firmware in the physical device.

It may not be supported depending on the printer. Please refer to [Printer Specification](#).

The result of the comparison is returned in the *pResult* parameter and will be one of the following values.

Value	Meaning
OPOS_CFV_FIRMWARE_OLDER (1)	Indicates that the version of one or more of the firmware files is older than the firmware in the device.
OPOS_CFV_FIRMWARE_SAME (2)	Indicates that the versions of all of the firmware files are the same as the firmware in the device.
OPOS_CFV_FIRMWARE_NEWER (3)	Indicates that the version of one or more of the firmware files is newer than the firmware in the device.

Return

The following value is returned by the method and placed in the **ResultCode** property.

Value	Meaning
OPOS_SUCCESS(0)	The method was successful.
OPOS_E_ILLEGAL(106)	An invalid parameter value was specified.
OPOS_E_NOHARDWARE(107)	The POS Printer is either off or offline.
OPOS_E_NOEXIST(109)	The file specified by <i>FirmWareFileName</i> was not found.
OPOS_E_BUSY(113)	Cannot perform while output is in progress.
OPOS_E_EXTENDED(114)	ResultCodeExtended : See the value and the explanation as below.
Other Values	See ResultCode .

* **ResultCodeExtended** value of OPOS_E_EXTENDED(114):

- OPOS_EFIRMWARE_BAD_FILE(281):
The specified data file is not correct or the data format is corrupt.
- OPOS_FCL_EFIRMWARE_DIFFERENT_BOOT(11001):
The class number of Boot program is different.
- OPOS_FCL_EFIRMWARE_DIFFERENT_MAIN(11002):
The class number of Main program is different.
- OPOS_FCL_EFIRMWARE_DIFFERENT_BOOTMAIN(11003):
The class number of both Boot program and Main program are different.

If updating the firmware using the class number of Boot or Main program is different, the firmware may be corrupt and unrecoverable. When updating the firmware of the different class, please confirm to the printer maker whether it is possible to update.

DirectIO Method

Syntax

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

Parameter	Description
<i>Command</i>	Command number. Specific values assigned by the Service Object.
<i>pData</i>	Pointer to additional numeric data. Specific values vary <i>Command</i> and Service Object.
<i>pString</i>	Pointer to additional string data. Specific values vary by <i>Command</i> and Service Object.

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

Value	Function
OPOS_FCL_PTR_DIO_GET_QR_ENCODING(4)	Get QR Code encoding.
OPOS_FCL_PTR_DIO_SET_QR_ENCODING(5)	Set QR Code encoding.
OPOS_FCL_PTR_DIO_SEND_BINARY_DATA (10)	Send binary data.

Remarks

Perform echa functionality following *Command* parameter.

Return

Return values vary by each command. See the description of function of each command.

OPOS_FCL_PTR_DIO_GET_QR_ENCODING

Parameter	Description
<i>Command</i>	OPOS_FCL_PTR_DIO_GET_QR_ENCODING(4)
<i>pData</i>	The current settings are retained.
<i>pString</i>	Not used.

Remarks

Get QR Code encoding.

This setting is initialized to OPOS_FCL_PTR_DIO_SYSTEM_LOCALE by **ClaimDevice** method.

Please refer to [OPOS_FCL_PTR_DIO_SET_QR_ENCODING](#) for the meaning of *pData*.

Return

The following value is returned by the method and placed in the **ResultCode** property.

Value	Function
OPOS_SUCCESS(0)	The method was successful.
OPOS_E_ILLEGAL(106)	An invalid parameter value was specified.
Other Values	See ResultCode .

OPOS_FCL_PTR_DIO_SET_QR_ENCODING

Parameter	Description
<i>Command</i>	OPOS_FCL_PTR_DIO_SET_QR_ENCODING(5)
<i>pData</i>	OPOS_FCL_PTR_DIO_SYSTEM_LOCALE(0) or OPOS_FCL_PTR_DIO_UTF8(1)
<i>pString</i>	Not used.

Remarks

Set QR Code encoding. Default setting is OPOS_FCL_PTR_DIO_SYSTEM_LOCALE.

This setting is canceled and reset to default once **Close()**–**Open()** method is performed, then it is necessary to set again.

If BinaryConversion property is not OPOS_BC_NONE, this setting is fixed to OPOS_FCL_PTR_DIO_SYSTEM_LOCALE.

If *pData* is OPOS_FCL_PTR_DIO_SYSTEM_LOCALE, encode QR Code in system locale.

If *pData* is OPOS_FCL_PTR_DIO_UTF8, encode QR Code in UTF-8.

Return

The following value is returned by the method and placed in the **ResultCode** property.

Value	Function
OPOS_SUCCESS(0)	The method was successful.
OPOS_E_ILLEGAL(106)	An invalid parameter value was specified.
OPOS_E_BUSY(113)	Cannot perform while output is in progress.
Other Values	See ResultCode .

OPOS_FCL_PTR_DIO_SEND_BINARY_DATA

Parameter	Description
<i>Command</i>	OPOS_FCL_PTR_DIO_SEND_BINARY_DATA (10)
<i>pData</i>	PTR_S_RECEIPT(2)
<i>pString</i>	Binary data to send. The max size is 192 x 1024 characters.

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.

The value that can be specified for *pString* is U+0000 to U+00FF.

pString is converted by **BinaryConversion** property.

Return

The following value is returned by the method and placed in the **ResultCode** property.

Value	Function
OPOS_SUCCESS(0)	The method was successful.
OPOS_E_ILLEGAL(106)	An invalid parameter value was specified.
Other Values	See ResultCode .

Open Method

Syntax

LONG Open (BSTR *DeviceName*);

The *DeviceName* parameter specifies the device name to open.
Specifies “FTP-POSPrinter”.

Remarks

Call to open a device for subsequent I/O.
When the **Open** method is successful, Common properties and additional class-specific properties are initialized.

Return

The following value is returned by the method and placed in the **ResultCode** property.

Value	Meaning
OPOS_SUCCESS(0)	Open successful.
OPOS_E_NOSERVICE(104)	Could not establish a connection to the corresponding Service Object.
OPOS_E_ILLEGAL(106)	The Control is already open.
OPOS_E_NOEXIST(109)	The specified <i>DeviceName</i> was not found.
OPOS_E_FAILER(111)	Failed to initialize OCX.

ReleaseDevice Method

Syntax

LONG ReleaseDevice ();

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.).

Return

The following value is returned by the method and placed in the **ResultCode** property.

Value	Meaning
OPOS_SUCCESS(0)	Exclusive access has been released. The Claimed property is now FALSE .
OPOS_E_ILLEGAL(106)	The application does not have exclusive access to the device.
OPOS_E_BUSY(113)	Cannot perform while asynchronous output is in progress.
Other values	See ResultCode .

ResetStatistics Method

Syntax

LONG ResetStatistics(BSTR *StatisticsBuffer*);

Remarks

This method is not supported.

Return

The following value is returned by the method and placed in the **ResultCode** property.

Value	Meaning
OPOS_E_ILLEGAL(106)	This method is not supported.
Other values	See ResultCode .

RetrieveStatistics Method

Syntax

LONG RetrieveStatistics(BSTR* *pStatisticsBuffer*);

Remarks

This method is not supported.

Return

The following value is returned by the method and placed in the **ResultCode** property.

Value	Meaning
OPOS_E_ILLEGAL(106)	This method is not supported.
Other values	See ResultCode .

UpdateStatistics Method

Syntax

LONG UpdateStatistics(BSTR *StatisticsBuffer*);

Remarks

This method is not supported.

Return

The following value is returned by the method and placed in the **ResultCode** property.

Value	Meaning
OPOS_E_ILLEGAL(106)	This method is not supported.
Other values	See ResultCode .

UpdateFirmware Method

Syntax

LONG UpdateFirmware(BSTR *FirmWareFileName*);

Value	Meaning
<i>FirmWareFileName</i>	Specifies the firmware file (including full path) that are to be downloaded into the device

Remarks

This method updates the firmware of a device with the firmware data file specified by the ***FirmWareFileName*** parameter regardless of the firmware version of a device.

When this method is invoked, OCX should check the specified firmware data file.

If data file is correct, this method should return immediately and the remainder of the update firmware process should continue asynchronously.

The OCX should notify the application of the status of the update firmware process by firing **StatusUpdateEvents** with values of OPOS_SUE_UF_PROGRESS(2100) + 1, 25, 50, 75 indicating the completion percentage of the update firmware process.

If the update firmware process completes successfully, the OCX should fire a **StatusUpdateEvent** with OPOS_SUE_UF_COMPLETE.

It may not be supported depending on the printer. Please refer to [Printer Specification](#).

If an error is detected during the asynchronous portion of the firmware updating process, one of the following **StatusUpdateEvents** will be fired.

Value	Meaning
OPOS_UF_FAILED_DEV_OK(2201)	The firmware updating process failed but the device is still operational.
OPOS_UF_FAILED_DEV_UNKNOWN (2204)	The firmware updating process failed and the device is in an indeterminate state.

Return

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

Value	Meaning
OPOS_SUCCESS(0)	The device firmware has been updated.
OPOS_E_ILLEGAL(106)	An invalid parameter value was specified.
OPOS_E_NOHARDWARE(107)	The POS Printer is either off or offline.
OPOS_E_NOEXIST(109)	The file specified by <i>FirmwareFileName</i> does not exist.
OPOS_E_BUSY(113)	Cannot perform while output is in progress.
OPOS_E_EXTENDED(114)	ResultCodeExtended =OPOS_EFIRMWARE_BAD_FILE(281): The specified firmware file is not correct, or file format is corrupt.
Other values	See ResultCode .

4.5. Specific properties

AsyncMode Property R/W

Syntax

BOOL AsyncMode;

Remarks

If **TRUE**, then the print methods **PrintNormal**, **CutPaper**, **MarkFeed**, **PrintBarCode**, **PrintBitmap**, **RotatePrint** and **TransactionPrint** will be performed asynchronously.

If **FALSE**, they will be printed synchronously.

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

CapCharacterSet Property

Syntax

LONG CapCharacterSet;

Remarks

Holds the default character set capability.

It has one of the following values.

Value	Meaning
PTR_CCS_ASCII(998)	The Character set supports all ASCII characters 0x20 to 0x7F;

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

CapCoverSensor Property

Syntax

BOOL CapCoverSensor;

Remarks

TRUE: The printer has a “cover open” sensor.

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

CapMapCharacterSet Property

Syntax

BOOL CapMapCharacterSet;

Remarks

TRUE: The Service object is able to map the characters to the character sets defined in **CharacterSetList**.

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

CapRec2Color Property

Syntax

BOOL CapRec2Color;

Remarks

FALSE: The receipt can not print dark plus an alternate color.

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

CapRecBarCode Property

Syntax

BOOL CapRecBarCode;

Remarks

TRUE: The receipt has barcode printing capability.

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

CapRecBitmap Property

Syntax

BOOL CapRecBitmap;

Remarks

TRUE: The receipt can print bitmaps.

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

CapRecBold Property

Syntax

BOOL CapRecBold;

Remarks

TRUE: The receipt can print bold characters.

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

Even If this property value is TRUE, it may not be supported depending on the printer. Please refer to [Printer Specification](#).

CapRecCartridgeSensor Property

Syntax

LONG CapRecCartridgeSensor;

Remarks

0: The receipt cartridge sensor is not supported.

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

CapRecColor Property

Syntax

LONG CapRecColor;

Remarks

0: The receipt color print is not supported.

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

CapRecDhigh Property

Syntax

BOOL CapRecDhigh;

Remarks

TRUE: The receipt can print double high characters.

This property is initialized by the **Open** metho

CapRecDwide Property

Syntax

BOOL CapRecDwide;

Remarks

TRUE: The receipt can print double wide characters.

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

CapRecDwideDhigh Property

Syntax

BOOL CapRecDwideDhigh;

Remarks

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

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

CapRecEmptySensor Property

Syntax

BOOL CapRecEmptySensor;

Remarks

TRUE: The receipt has an out-of-paper sensor.

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

CapRecItalic Property

Syntax

BOOL CapRecItalic;

Remarks

FALSE: The receipt can not print italic characters.

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

CapRecLeft90 Property

Syntax

BOOL CapRecLeft90;

Remarks

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

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

Even If this property value is TRUE, it may not be supported depending on the printer. Please refer to [Printer Specification](#).

CapRecMarkFeed Property

Syntax

LONG CapRecMarkFeed;

Remarks

PTR_MF_TO_NEXT_TOF(8): Feed the Mark Sensed paper to the paper's next top of form.

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

CapRecNearEndSensor Property

Syntax

BOOL CapRecNearEndSensor;

Remarks

TRUE: The receipt have a low paper sensor.

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

Even If this property value is TRUE, it may not be supported depending on the printer. Please refer to [Printer Specification](#).

CapRecPapercut Property

Syntax

BOOL CapRecPapercut;

Remarks

TRUE: The receipt can perform paper cuts.

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

Even If this property value is TRUE, it may not be supported depending on the printer. Please refer to [Printer Specification](#).

CapRecPresent Property

Syntax

BOOL CapRecPresent;

Remarks

TRUE: The receipt print station is present.

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

CapRecRight90 Property

Syntax

BOOL CapRecRight90;

Remarks

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

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

Even If this property value is TRUE, it may not be supported depending on the printer. Please refer to [Printer Specification](#).

CapRecRotate180 Property

Syntax

BOOL CapRecRotate180;

Remarks

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

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

CapRecStamp Property

Syntax

BOOL CapRecStamp;

Remarks

FALSE: The receipt does not have a stamp capability.

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

CapRecUnderline Property

Syntax

BOOL CapRecUnderline;

Remarks

TRUE: The receipt can underline characters.

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

Even If this property value is TRUE, it may not be supported depending on the printer. Please refer to [Printer Specification](#).

CapTransaction Property

Syntax

BOOL CapTransaction;

Remarks

TRUE: The printer transactions are supported by each station.

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

CartridgeNotify Property R/W

Syntax

LONG CartridgeNotify;

Remarks

Contains the type of cartridge state notification selected by the application.

Value	Meaning
PTR_CN_DISABLED(0)	The Control will not provide any cartridge state notifications to the application. No cartridge state notification StatusUpdateEvents will be fired, and RecCartridgeState may not be set.

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

Return

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

Value	Meaning
OPOS_E_ILLEGAL(106)	This property is not be able to set.

FTP2166000EQ

FTP-POS

POSPrinter OPOS Control Function specification

CharacterSet Property R/W

Syntax

LONG 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	(USA:Standard Europe) character set. When the BinaryConversion property is OPOS_BC_NIBBLE or OPOS_BC_DECIMAL, depending on the printer model, '€' can be printed if 0xFF is specified. Please refer to Printer Specification .
932	Windows code page: Japanese Shift-JIS.
950	Windows code page: Traditional Chinese Big5.
PTR_CS_ASCII (998)	The ASCII character set, supporting the ASCII characters between 0x20 and 0x7F. The value of this constant is 998.
PTR_CS_WINDOWS(999)	Windows ANSI character set. This is the same as CP1252. This supports the ASCII characters between 0x20 and 0x7F. The value of this constant is 999. Depending on the printer model, '€'(0xFF) can be printed. Please refer to Printer Specification .

Return

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

Value	Meaning
OPOS_SUCCESS(0)	The property was set successfully.
OPOS_E_ILLEGAL(106)	An invalid property value was used.
Other values	See ResultCode property.

CharacterSetList Property

Syntax

BSTR CharacterSetList;

Remarks

A string of character set numbers. “437,932,950,998,999” is be set.
This property is initialized by the **Open** method.

CoverOpen Property

Syntax

BOOL 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.

ErrorLevel Property

Syntax

LONG ErrorLevel;

Remarks

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

Value	Meaning
PTR_EL_NONE(1)	No error condition is present.
PTR_EL_RECOVERABLE(2)	A recoverable error has occurred.
PTR_EL_FATAL(3)	A non-recoverable error has occurred. This value is set when following error has occurred. ResultCode =OPOS_E_EXTENDED(114), ResultCodeExtended = OPOS_FCL_EPTR_POWERSUPPLY(10001) OPOS_FCL_EPTR_HARDWARE(10004)

This property is set by the Control just before firing an **ErrorEvent**.
A recoverable error and a non-recoverable error have occurred at the same time, a non-recoverable error is set with priority.
When the error is cleared, then the property is changed to PTR_EL_NONE(1).

ErrorStation Property

Syntax

LONG ErrorStation;

Remarks

Holds the station (PTR_S_RECEIPT(2)) that was printing when an error was detected.
When an error is released, this property is set to 0.
This property is set just before an **ErrorEvent** is fired.

ErrorString Property

Syntax

BSTR ErrorString;

Remarks

Holds a vendor-supplied description of the current error.

This property is set by the Control just before firing an **ErrorEvent**. If no description is available, the property is set to an empty string. When the error is cleared, then the property is changed to an empty string.

When some errors occurred at same time, ErrorString is set as following priority.

The following character string is set in POS Printer.

High priority	
Hardware abnormal	"Hardware Error"
Power supply voltage abnormal	"Power Supply Error"
Printer head temperature abnormal	"Head Hot"
Output data abnormal	"Data Error"
Cutter abnormal	"Cutter Error"
Presenter abnormal	"Presenter Error"
Mark scan error	"Mark Error"
Cover open	"Cover Open"
Out of paper	"Paper End"
Output data time-out	"Timeout"
Printer off / offline	"Offline"
Low priority	

FlagWhenIdle Property R/W

Syntax

BOOL FlagWhenIdle;

Remarks

TRUE: The Control will fire a **StatusUpdateEvent** if it is in the idle state.

FALSE: This event will not be fired.

FlagWhenIdle is automatically reset to **FALSE** when the status event is fired.

The main use of idle status event that is controlled by this property is to give the application control when all outstanding asynchronous outputs have been processed. The event will be fired if the outputs were completed successfully or if they were cleared by an **ErrorEvent** handler.

If the **State** is already set to OPOS_S_IDLE(2) when the **FlagWhenIdle** property is set to **TRUE**, then a **StatusUpdateEvent** is fired immediately. The application can therefore depend upon the event, with no race condition between the starting of its last asynchronous output and the setting of this flag.

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

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

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

FontTypefaceList Property

Syntax

BSTR FontTypefaceList;

Remarks

This property is set to an empty string. The default FontType is only supported.

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

MapCharacterSet Property R/W

Syntax

BOOL MapCharacterSet;

Remarks

TRUE: If **MapCharacterSet** is **TRUE** and when outputting data, the Service maps the characters transferred by the application to the character set selected in the **CharacterSet** property for printing data.

This property is always **TRUE**.

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

MapMode Property R/W

Syntax

LONG 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 which is converted from each unit.

Value	Meaning
PTR_MM_DOTS(1)	Dot-line unit of POS Printer. 0.125 mm (1 dot)
PTR_MM_TWIPS(2)	1/1440 inch (0/1409 dot)
PTR_MM_ENGLISH(3)	0.001 inch (0.203 dot)
PTR_MM_METRIC(4)	0.01 millimeter (0.08 dot)

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

The value of **MapMode** is initialized to PTR_MM_DOTS(1) when the device is first enabled following the **Open** method.

Return

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

Value	Meaning
OPOS_SUCCESS(0)	The property was set successfully
OPOS_E_ILLEGAL(106)	An invalid mapping mode was specified.

RecBarCodeRotationList Property

Syntax

BSTR RecBarCodeRotationList;

Remarks

Holds the directions in which a receipt barcode may be rotated.

Some printer models do not support the direction of rotation that is set in this property. Please refer to [Printer Specification](#).

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

The string consists of rotation strings separated by commas. The legal rotation strings are:

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

RecBitmapRotationList Property

Syntax

BSTR RecBitmapRotationList;

Remarks

Holds the directions in which a receipt bitmap may be rotated.

Some printer models do not support the direction of rotation that is set in this property. Please refer to [Printer Specification](#).

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

The string consists of rotation strings separated by commas. The legal rotation strings are:

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

RecCartridgeState Property

Syntax

LONG RecCartridgeState;

Remarks

This property contains the status of the currently selected Receipt cartridge (ink, ribbon or toner).

This value is fixed at the following value.

Value	Meaning
PTR_CART_UNKNOWN(268435456)	Device does not support cartridge state reporting.

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

RecCurrentCartridge Property R/W

Syntax

LONG RecCurrentCartridge;

Remarks

The selection of Receipt cartridge is not supported. This property is initialized to 0.

Return

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

Value	Meaning
OPOS_E_ILLEGAL(106)	The selection of Receipt cartridge is invalid.
Other values	See ResultCode property.

RecEmpty Property

Syntax

BOOL 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.

RecLetterQuality Property R/W

Syntax

BOOL RecLetterQuality;

Remarks

TRUE: Prints in high quality mode. (Medium speed mode printing)

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

This property is initialized to **FALSE** when the device is first enabled following the **Open** method.

Return

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

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

RecLineChars Property R/W

Syntax

LONG 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**, **RecSideWayMaxChars**, and **RecSidewaysMaxlines**.

Return

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

Value	Meaning
OPOS_SUCCESS(0)	The property was set successfully.
OPOS_E_ILLEGAL(106)	An invalid line character width was specified.

RecLineCharsList Property

Syntax

BSTR RecLineCharsList;

Remarks

Holds the line character width supported by the receipt station.

This property is initialized by the **Open** method. Please refer to [Printer Specification](#).

RecLineHeight Property R/W

Syntax

LONG 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 to the printer's default line height when the device is first enabled the **Open** method.

Please refer to [Printer Specification](#).

Return

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

Value	Meaning
OPOS_E_ILLEGAL(106)	This property cannot be set. This property is always readable.

RecLineSpacing Property R/W

Syntax

LONG 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 initialized to the printer's default line spacing when the device is first enabled 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**.

Return

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

Value	Meaning
OPOS_SUCCESS(0)	The property was set successfully
OPOS_E_ILLEGAL(116)	An invalid value was specified.
Other values	See ResultCode .

RecLinesToPaperCut Property

Syntax

LONG 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.

RecLineWidth Property

Syntax

LONG 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](#).

RecNearEnd Property

Syntax

BOOL 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.

RecSidewaysMaxChars Property

Syntax

LONG 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](#).

RecSidewaysMaxLines Property

Syntax

LONG 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.

RotateSpecial Property R/W

Syntax

LONG RotateSpecial;

Remarks

Holds the rotation orientation for barcodes.

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

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

Return

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

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

The following Specific properties of POS Printer are not supported.

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

4.6. Specific Methods

BeginInsertion Method

Syntax

LONG BeginInsertion (**LONG** *Timeout*);

Remarks

Because this method targets the slip printer, it is not supported by this OCX.

Return

One of the following values is returned by the method and placed in the **ResultCode** property.

Value	Meaning
OPOS_E_ILLEGAL(106)	The POS Printer does not have a slip station.
Other Values	See ResultCode property.

BeginRemoval Method

Syntax

LONG BeginRemoval (**LONG** *Timeout*);

Remarks

Because this method targets the slip printer, it is not supported by this OCX.

Return

One of the following values is returned by the method and placed in the **ResultCode** property.

Value	Meaning
OPOS_E_ILLEGAL(106)	The POS Printer does not have a slip station.
Other Values	See ResultCode .

ChangePrintSide Method

Syntax

LONG ChangePrintSide (**LONG** *Side*);

Remarks

Because this method targets the slip printer, it is not supported by this OCX.

Return

One of the following values is returned by the method and placed in the **ResultCode** property.

Value	Meaning
OPOS_E_ILLEGAL(106)	The POS Printer does not have a slip station.
Other Values	See ResultCode .

ClearPrintArea Method

Syntax

LONG ClearPrintArea ();

Remarks

This method is not supported.

Return

The following value is returned by the method and placed in the **ResultCode** property.

Value	Meaning
OPOS_E_ILLEGAL(106)	This method is not supported.
Oher values	See ResultCode .

CutPaper Method

Syntax

LONG CutPaper (LONG *Percentage*);

The *Percentage* parameter indicates the percentage of paper cut. The value 100 causes a full paper cut. Other values causes a partial cut.

Remarks

Call to cut the receipt paper.

The correspondence depends on the printer. Please refer to [Printer Specification](#).

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

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

Return

One of the following values is returned by the method and placed in the **ResultCode** property.

Value	Meaning
OPOS_SUCCESS(0)	The method was successful.
OPOS_E_ILLEGAL(106)	An invalid percentage was specified, the printer does not have paper cutter.
OPOS_E_NOHARDWARE(107)	The POS Printer is either off or offline.
OPOS_E_FAILURE(111)	OPOS control is in error state. Perform again after recovering from error condition.
OPOS_E_BUSY(113)	Cannot be executed while output is in progress.
Other Values	See ResultCode .

EndInsertion Method

Syntax

LONG EndInsertion ();

Remarks

Because this method targets the slip printer, it is not supported by this OCX.

Return

One of the following values is returned by the method and placed in the **ResultCode** property.

Value	Meaning
OPOS_E_ILLEGAL(106)	The POS Printer does not have a slip station.
Other Values	See ResultCode .

EndRemoval Method

Syntax

LONG EndRemoval ();

Remarks

Because this method targets the slip printer, it is not supported by this OCX.

Return

One of the following values is returned by the method and placed in the **ResultCode** property.

Value	Meaning
OPOS_E_ILLEGAL(106)	The POS Printer does not have a slip station.
Other Values	See ResultCode .

MarkFeed Method

Syntax

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

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

The parameter value of *Type* is as follows.

値	Meaning
PTR_MF_TO_NEXT_TOF(8)	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.

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

Return

One of the following values is returned by the method and placed in the **ResultCode** property.

Value	Meaning
OPOS_SUCCESS(0)	This method completed successfully.
OPOS_E_ILLEGAL(106)	An invalid value was specified.
OPOS_E_NOHARDWARE(107)	The POS Printer is either off or offline.
OPOS_E_FAILURE(111)	OPOS control is in error state. Perform again after recovering from error condition.
OPOS_E_BUSY(113)	Cannot be executed while output is in progress.
Other Values	See ResultCode .

PageModePrint Method

Syntax

LONG PageModePrint(LONG *Control*);

Remarks

This method is not supported.

Return

The following value is returned by the method and placed in the **ResultCode** property.

Value	Meaning
OPOS_E_ILLEGAL(106)	This method is not supported.
Other values	See ResultCode .

PrintBarcode Method

Syntax

LONG PrintBarcode (LONG Station, BSTR Data, LONG Symbology, LONG Height, LONG Width, LONG Alignment, LONG TextPosition);

Parameter	Description
<i>Station</i>	The printer station to be used. Set to PTR_S_RECEIPT(2).
<i>Data</i>	Character string of barcode. The data format depends on the value of BinaryConversion property. See BinaryConversion property for details.
<i>Symbology</i>	Barcode symbol type to use. (Please refer to the following values.)
<i>Height</i>	Barcode height. Expressed in the unit of measure given by MapMode . Set 1–255dot.
<i>Width</i>	Barcode width. Expressed in the unit of measure given by MapMode . When normal / upside-down mode, <i>Width</i> can be set in the range of RecLineWidth . When rotate printing 90° to the left / right, the range of <i>Width</i> depends on the printer model.
<i>Alignment</i>	Placement of the barcode. See values below.
<i>TextPosition</i>	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
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_Code128(110)	Code 128
PTR_BCS_PDF417(201)	PDF417
PTR_BCS_MAXICODE(202)	MaxiCodeMAXICODE(Mode 2)
PTR_BCS_OTHER(501)	MaxiCodeMAXICODE(Mode 3)
PTR_BCS_OTHER+1(502)	MaxiCodeMAXICODE(Mode 4、Mode 5)
PTR_BCS_OTHER+2(503)	MaxiCodeMAXICODE(Mode 6)
PTR_BCS_OTHER+3(504)	QR Code
PTR_BCS_OTHER+4(505)	GS1DataBar Omnidirectional
PTR_BCS_OTHER+5(506)	GS1DataBar Truncated
PTR_BCS_OTHER+6(507)	GS1DataBar Stacked
PTR_BCS_OTHER+7(508)	GS1DataBar Stacked Omnidirectional
PTR_BCS_OTHER+8(509)	GS1DataBar Limited
PTR_BCS_OTHER+9(510)	GS1DataBar Expanded
PTR_BCS_OTHER+10(511)	GS1DataBar Expanded Stacked

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 PTR_BC_LEFT.

Value	Meaning
PTR_BC_LEFT(-1)	Align with the left-most print column. (It is executed in the direction of

FTP2166000EQ

FTP-POS

POSPrinter OPOS Control Function specification

	the print data. Therefore, when turning 180 degrees, it will be right-most aligned in the POS Printer printing direction.)
PTR_BC_CENTER(-2)	Align in the center of the station.
PTR_BC_RIGHT(-3)	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 <i>Width</i> in OCX) + distance from left-most print column exceeds RecLineWidth , it returns the value OPOS_E_ILLEGAL(106).

The *TextPosition* parameter values are as follows. The correspondence depends on the printer. Please refer to [Printer Specification](#).

Value	Meaning
PTR_BC_TEXT_NONE(-11)	No text is printed. Only print the barcode.
PTR_BC_TEXT_ABOVE(-12)	Print the text above the barcode.
PTR_BC_TEXT_BELOW(-13)	Print the text below the barcode.

Remarks

Call to print a barcode on the specified printer station.

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

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

Symbologies	Printable characters	String length	Width (dots)
PTR_BCS_UPCA	'0' - '9'	11 - 12	95 - 864
PTR_BCS_UPCE		11 - 12	51 - 864
PTR_BCS_EAN8		7 - 8	67 - 864
PTR_BCS_JAN8			
PTR_BCS_EAN13		12 - 13	95 - 864
PTR_BCS_JAN13			
PTR_BCS_CODE39	'0' - '9', 'A' - 'Z', space, '\$', '%', '+', -', ':', '/' (Start and stop character '*' is automatically added.)	1 - 34	47 - 864
PTR_BCS_ITF	'0' - '9'	2 - 62	27 - 864
PTR_BCS_CODABAR	'0' - '9', 'A' - 'D', '\$', '+', '-', ':', '/', , '	3 - 71	41 - 864
PTR_BCS_CODE128	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 - 51	46 - 864
PTR_BCS_PDF417	0x00 - 0xFF	1 - 410byte	158 - 864

PTR_BCS_MAXICODE (Mode 2)	Primary Message Number '0' - '9' Secondary Message 0x01 - 0xFF	18~85	228
PTR_BCS_OTHER (MAXI mode 3)	Primary Message Postal code '0' - '9' , 'A' - 'Z' , space, “ “#&%&'()*+,-./ ” Country code Service code Number '0' - '9' Secondary Message 0x01 ~ 0xFF	15~85	228
PTR_BCS_OTHER + 1 (MAXI mode4, mode 5)	0x01 - 0xFF	1~80	228
PTR_BCS_OTHER + 2 (MAXI mode6)	0x01 - 0xFF	1~80	228
PTR_BCS_OTHER + 3 (QRcode)	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 - 2048byte	21 -
PTR_BCS_OTHER + 4 (GS1DataBar Omnidirectional)	'0' - '9'	13 - 14	- 864
PTR_BCS_OTHER + 5 (GS1DataBar Truncated)			
PTR_BCS_OTHER + 6 (GS1DataBar Stacked)			
PTR_BCS_OTHER + 7 (GS1DataBar Stacked Omnidirectional)			
PTR_BCS_OTHER + 8 (GS1DataBar Limited)			

PTR_BCS_OTHER + 9 (GS1DataBar Expanded) PTR_BCS_OTHER + 10 (GS1DataBar Expanded Stacked)	‘0’-‘9’, ‘A’-‘Z’, ‘a’-‘z’, space, ‘!’, ‘“’, ‘%’, ‘&’, ‘“’, ‘(’, ‘)’, ‘*’, ‘+’, ‘,’, ‘-’, ‘:’, ‘/’, ‘:’, ‘;’, ‘<’, ‘=’, ‘>’, ‘?’, ‘_’, FNC1(0x1D)	1 – 77 (Only Numeric) *) Maximum data length changes according to the kind of character.	– 864
--	--	--	-------

The determination algorithm of barcode width for each symbology are shown below.

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

Symbology	Calculation formula of printing width
PTR_BCS_UPCA	Barcode width = $95 * \text{dotNarrow}$
PTR_BCS_UPCE	Barcode width = $51 * \text{dotNarrow}$
PTR_BCS_JAN8	Barcode width = $67 * \text{dotNarrow}$
PTR_BCS_JAN13	Barcode width = $95 * \text{dotNarrow}$
PTR_BCS_CODE39	Barcode width = $(6 * \text{dotNarrow} + 3 * \text{dotWide}) * (2 + \text{Length}) + (\text{Length} + 1) * \text{dotNarrow}$ (Length = Number of print characters, dotWide = $3 * \text{dotNarrow}$)
PTR_BCS_ITF	Barcode width = $(3 * \text{dotNarrow} + 2 * \text{dotWide}) * \text{Length} + (6 * \text{dotNarrow} + \text{dotWide})$ (Length = Number of print characters, dotWide = $3 * \text{dotNarrow}$)
PTR_BCS_CODABAR	Barcode width = $(5 * \text{dotNarrow} + 2 * \text{dotWide}) * (\text{Length} - \text{Wlen}) + (4 * \text{dotNarrow} + 3 * \text{dotWide}) * \text{Wlen} + \text{dotNarrow} * \text{Length}$ (Length = Number of print characters, dotWide = $3 * \text{dotNarrow}$, Wlen = Number of “:”, “/”, “.”, “+”, “A”, “B”, “C”, “D”)
PTR_BCS_CODE128	Barcode width = $11 * \text{dotNarrow} * \text{Length} + 13 * \text{dotNarrow}$ (Length = Number of print characters – Number of special characters*) *) The number of characters combined with “[”. If “[AA [BA]” is specified, Length is “Length = 6 – 2 = 4”.
PTR_BCS_PDF417	Barcode width = $(17 * \text{LineSum} + 69) * \text{dotNarrow} + 2 * 2 * \text{dotNarrow}$ (LineSum = column number in 1 line)
PTR_BCS_MAXICODE PTR_BCS_OTHER PTR_BCS_OTHER + 1 PTR_BCS_OTHER + 2	Symbol width = 228 dots Symbol height = 228 dots
PTR_BCS_OTHER + 4 (GS1DataBar Omnidirectional)	Symbol width = $96 * \text{Module size}$ Symbol height = $33 * \text{Module size}$
PTR_BCS_OTHER + 5 (GS1DataBar Truncated)	Symbol width = $96 * \text{Module size}$ Symbol height = $13 * \text{Module size}$
PTR_BCS_OTHER + 6 (GS1DataBar Stacked)	Symbol width = $50 * \text{Module size}$ Symbol height = $(13 + \text{Magnification of separation pattern}) * \text{Module size}$
PTR_BCS_OTHER + 7 (GS1DataBar Stacked omnidirectional)	Symbol width = $50 * \text{Module size}$ Symbol height = $(66 + 3 * \text{Magnification of separation pattern}) * \text{Module size}$
PTR_BCS_OTHER + 8 (GS1DataBar Limited)	Symbol width = $74 * \text{Module size}$ Symbol height = $10 * \text{Module size}$
PTR_BCS_OTHER + 9 (GS1DataBar Expanded)	Symbol width = $((49 * \text{Number of segments}) + 8) * \text{Module size} \div 2$ Symbol height = $34 * \text{Module size}$ *) The number of segments changes depending on data length.

PTR_BCS_OTHER + 10 (GS1DataBar Expanded Stacked)	<p>Symbol width = $((49 * \text{Number of segments}) + 8) * \text{Module size} \div 2$</p> <p>Symbol height = $(34 * \text{Stacked step} + 3 * \text{Magnification of separation pattern} \times (\text{Stacked step} - 1)) * \text{Module size}$</p> <p>*) The number of segments and the stacked lines vary depending on the data length.</p> <p>It is can be printed only if they meet the following formula.</p> <p>$(\text{symbol width} / 8) * \text{symbol height} < 64 \text{ KB}$</p>
--	--

Notes for the barcode printing

- 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.
- When ITF is specified, the number of print characters must be specified as an even number. If you specify an odd number, OPOS_E_ILLEGAL (106) is returned.
- When specifying CODABAR, the beginning and the end of the character are always "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, OPOS_E_ILLEGAL (106) is returned.
- 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. OPOS_E_ILLEGAL (106) is returned 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 → OPOS_E_ILLEGAL is returned.

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

- When printing CODE 128, set the character by the followings.
 - 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.
 - 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, OPOS_E_ILLEGAL (106) is returned if you specify something other than FUNC 1.
 - In CODE B, specify "{{" to set "{".
 - Specify "{S" to set SHIFT. At this setting, the code set of one character is shifted to CODE A or CODE B. OPOS_E_ILLEGAL(106) is returned when specifying it in CODE C.

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

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)

FTP2166000EQ

FTP-POS

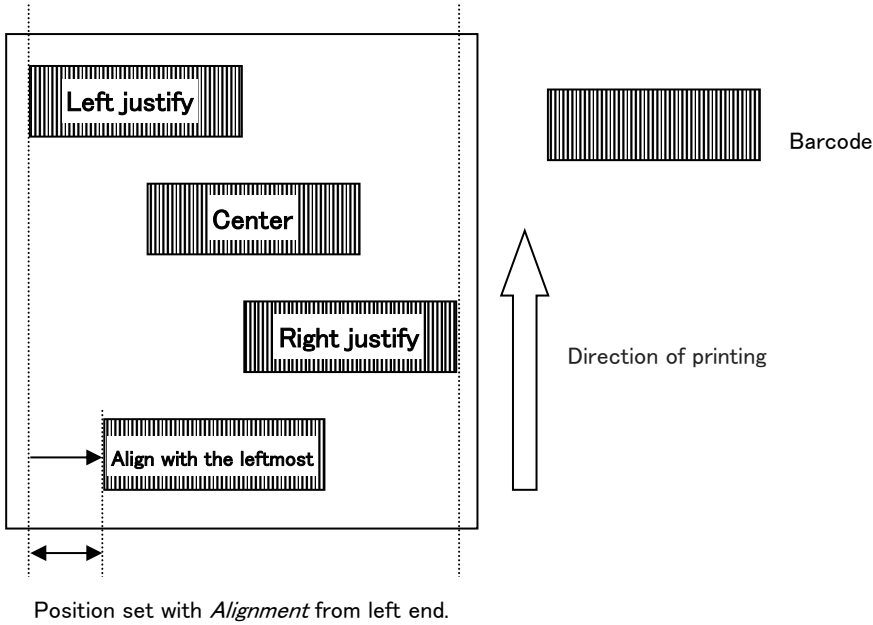
POSprinter OPOS Control Function specification

@	@	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”

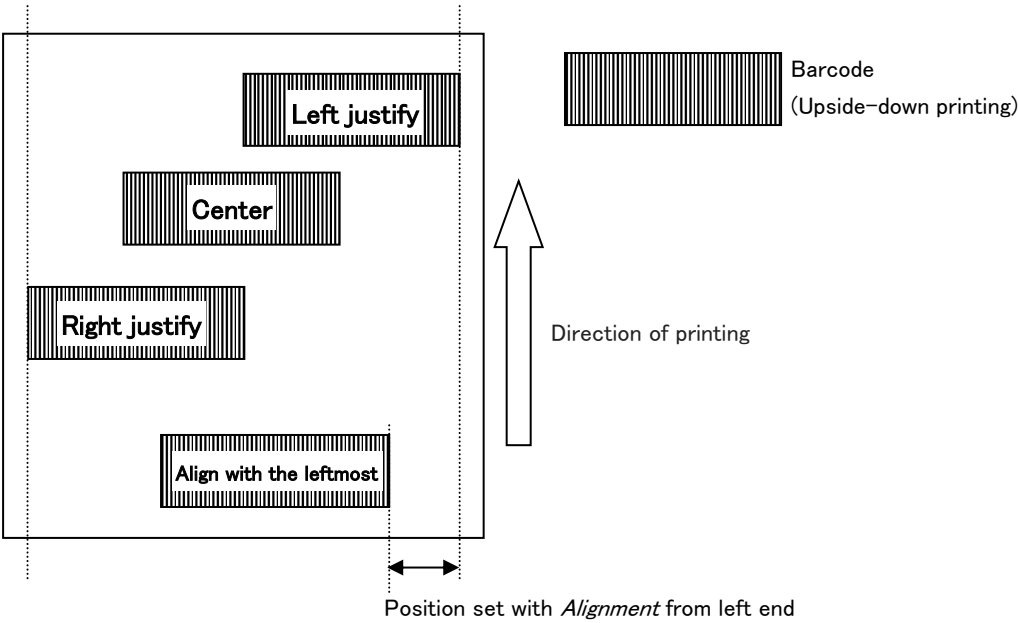
6. PDF417 increases the number of barcode rows as data length increases.OCX internally calculates the vertical and horizontal sizes and prints them according to the specified *Width* parameter and *Height* parameter.
If it is judged that printing is impossible, OPOS_E_ILLEGAL(106) will be returned.
If the **BinaryConversion** property is OPOS_BC_NONE, PDF417 will be printed using *Data* encoded with system locale.
If the **BinaryConversion** property is not OPOS_BC_NONE, PDF417 will be printed using *Data* converted according to the value of **BinaryConversion** property.
7. If the **BinaryConversion** property is OPOS_BC_NONE, QR Code will be printed using *Data* encoded with system locale. It is possible to change the encoding to UTF-8 by **DirectIO** method.
If the **BinaryConversion** property is not OPOS_BC_NONE, QR Code will be printed using *Data* converted according to the value of **BinaryConversion** property.
QR Code is the same size in both the vertical and horizontal, it prints QRcode by the approximate value of the value specified by the *Width* parameter. If the *Height* parameter is less than or equal to 0, OPOS_E_ILLEGAL(106) is returned. The print size is decided by data length, the character mode, the module size, and the error correction level (H, Q, M, L) 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.
8. The size and the module size of GS1 DataBar are decided from the approximate value of the *Width* parameter value.
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.
9. Set the presence and the position of HRI character (Human Readable Interpretation) by specifying TextPosition.
Specify the position of the HRI character in PTR_BC_TEXT_ABOVE (upper side of the barcode) or PTR_BC_TEXT_BELOW (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.
10. 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.



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



Return

When this property is set, one of the following values is stored in the ResultCode property.

Value	Meaning
OPOS_SUCCESS(0)	The method was successful.
OPOS_E_ILLEGAL(106)	One of the following errors occurred. <ul style="list-style-type: none">– <i>Station</i> does not exist.– <i>Station</i> does not support barcode printing.– <i>Height</i> or <i>Width</i> is zero or is too big.– <i>Symbology</i> is not supported.– Not all characters in <i>data</i> are supported by <i>Symbology</i>.– <i>Alignment</i> 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.)– <i>TextPosition</i> is invalid.
OPOS_E_NOHARDWARE(107)	The POS Printer is either off or offline.
OPOS_E_FAILURE(111)	OPOS control is in error state. Perform again after recovering from error condition.
OPOS_E_BUSY(113)	Cannot be executed while output is in progress.
Other Values	See ResultCode .

PrintBitmap Method

Syntax

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

Parameter	Meaning
<i>Station</i>	The printer station to be used. Set to PTR_S_RECEIPT(2).
<i>FileName</i>	Name of Windows bitmap file. The file must be in uncompressed format. (Specify the full path or the 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)	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 PTR_BM_LEFT.

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.
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 <i>Width</i> and this does not exceed the parameter limit of <i>Width</i> .

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 achieve good performance. It is recommended to use **SetBitmap** and escape sequences.

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 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.

The printable Windows bitmap file is 1/4/8/24 bit and uncompressed format.

Return

One of the following values is returned by the method and placed in the **ResultCode** property.

Value	Meaning
OPOS_SUCCESS(0)	The method was successful.
OPOS_E_ILLEGAL(106)	One of the following errors 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.
OPOS_E_NOHARDWARE(107)	The POS Printer is either off or offline.
OPOS_E_FAILURE(111)	OPOS control is in error state. Perform again after recovering from error

FTP2166000EQ

FTP-POS

POSPrinter OPOS Control Function specification

OPOS_E_NOEXIST(109)	condition. <i>FileName</i> was not found.
OPOS_E_BUSY(113)	Cannot be executed while output is in progress. (Can only be returned if AsyncMode is FALSE .)
Other Values	See ResultCode .

PrintImmediate Method

Syntax

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

Parameter	Description
<i>Station</i>	The printer station to be used. Set to PTR_S_RECEIPT(2).
<i>Data</i>	The characters to be printed. May consist of printable characters, escape sequences, carriage returns (13 decimal), and line feeds (10 decimal). See BinaryConversion property for details.

Remarks

Prints *data* on the printer *station* immediately.

Call this method when you print Data on the POS Printer. Print data exceeding the maximum number of digits is printed on the next line.

Special character values within *data* are:

Value	Meaning
Newline/Line Feed (10)	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)	If a Carriage Return immediately precedes a Line Feed, then it is ignored. The carriage return behaves 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.

Return

One of the following values is returned by the method and placed in the **ResultCode** property.

Value	Meaning
OPOS_SUCCESS(0)	The method was successful.
OPOS_E_ILLEGAL(106)	The specified <i>Station</i> does not exist.
OPOS_E_NOHARDWARE(107)	The POS Printer is either off or offline.
OPOS_E_FAILURE(111)	OPOS control is in error state. Perform again after recovering from error condition.
OPOS_E_BUSY(113)	Cannot be executed while output is in progress.
Other Values	See ResultCode .

PrintNormal Method

Syntax

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

Parameter	Description
<i>Station</i>	The printer station to be used. Set to PTR_S_RECEIPT(2).
<i>Data</i>	The characters to be printed. May consist of printable characters, escape sequences, carriage returns (13 decimal), and line feeds (10 decimal). See BinaryConversion property for details.

Remarks

Prints *data* on the printer *Station*.

Call this method when you print Data on the POS Printer. Print data exceeding the maximum characters in one line is printed on the next 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)	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)	If a Carriage Return immediately precedes a Line Feed, then it is ignored. The carriage return behaves 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.

Return

When this property is set, one of the following values is stored in the ResultCode property.

Value	Meaning
OPOS_SUCCESS(0)	The method was successful.
OPOS_E_ILLEGAL(106)	The specified <i>Station</i> does not exist.
OPOS_E_NOHARDWARE(107)	The POS Printer is either off or offline.
OPOS_E_FAILURE(111)	OPOS control is in error state. Perform again after recovering from error condition.
OPOS_E_BUSY(113)	Cannot be executed while output is in progress.
Other Values	See ResultCode .

PrintTwoNormal Method

Syntax

LONG PrintTwoNormal (**LONG** *Stations*, **BSTR** *Data1*, **BSTR** *Data2*);

Parameter	Description
<i>Station</i>	The printer stations to be used.
<i>Data1</i>	The characters to be printed on the first station.
<i>Data2</i>	The characters to be printed on the second station.
	See BinaryConversion property for details.

Remarks

Prints two strings on two print stations simultaneously. This method targets the slip printer then is not supported in this OCX.

Return

One of the following values is returned by the method and placed in the **ResultCode** property.

Value	Meaning
OPOS_E_ILLEGAL(106)	The specified <i>Station</i> does not exist.
Other Values	See ResultCode .

RotatePrint Method

Syntax

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

Parameter	Description
<i>Station</i>	The printer station to be used. Set to PTR_S_RECEIPT(2).
<i>Rotation</i>	Direction of rotation. See values below.
The <i>Rotation</i> parameter has one of the following values.	
Value	Meaning
PTR_RP_RIGHT90(257)	Start rotated printing 90° to the right (clockwise)
PTR_RP_LEFT90(258)	Start rotated printing 90° to the left (counter-clockwise)
PTR_RP_ROTATE180(259)	Start rotated printing 180°, that is, print upside-down
PTR_RP_BARCODE(4096)	Start rotated the barcode printing. This value is ORed with one of the above start rotated print values.
PTR_RP_BITMAP(8192)	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.
PTR_RP_NORMAL(1)	End rotated printing.

Remarks

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

If *Rotation* includes PTR_RP_ROTATE180, then upside-down print mode is entered. Subsequent calls to **PrintNormal** or **PrintImmediate** will print the data upside-down until **RotatePrint** is called with *Rotation* set to 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** and **PrintImmediate** may be used while in upside-down print mode.

If *Rotation* includes PTR_RP_RIGHT90 or PTR_RP_LEFT90, then sideways print mode is entered. Subsequent calls to **PrintNormal** and **DirectIO** will buffer the print data until **RotatePrint** is called with *Rotation* set to PTR_RP_NORMAL.

(In this case, the above 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 fired. 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 normal print mode, it is printed upside-down.)

If *Rotation* includes PTR_RP_NORMAL(1), 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 PTR_RP_BARCODE(4096) and/or PTR_RP_BITMAP(8192), then any barcodes (printed with **PrintBarcode**) and/or bitmaps (printed with **PrintBitmap**) submitted for printing during the **RotatePrint** processing cycle will also be rotated. Such rotation will be within the limitations that may be

FTP2166000EQ

FTP-POS

POSPrinter OPOS Control Function specification

specified by the **RecBarcodeRotationList**, and **RecBitmapRotationList** properties respectively. The rotated print of barcode and bitmap is canceled by specifying PTR_RP_NORMAL(1).

When **TransactionPrint** 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 PTR_RP_BARCODE(4096) and PTR_RP_BITMAP(8192).

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

Return

One of the values in the following table is returned by the method and placed in the **ResultCode** property.

Value	Meaning
OPOS_SUCCESS(0)	The method was successful..
OPOS_E_ILLEGAL(106)	The specified <i>Station</i> does not exist, or the <i>Station</i> does not support the specified rotation. It is in different rotation mode. Once set PTR_RP_NORMAL(1), or cancel the rotated print by ClearOutput and then perform again.
OPOS_E_NOHARDWARE(107)	The POS Printer is either off or offline.
OPOS_E_FAILURE(111)	OPOS control is in error state. Perform again after recovering from error condition.
OPOS_E_BUSY(113)	Cannot be executed while output is in progress. (Can only returned if AsyncMode is FALSE .)
Other Values	See ResultCode .

SetBitmap Method

Syntax

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

Parameter	Description
<i>BitmapNumber</i>	The number to be assigned to this bitmap. Valid bitmap numbers are 1 through 255.
<i>Station</i>	The printer station to be used. Set to PTR_S_RECEIPT(2).
<i>FileName</i>	Name of Windows bitmap file. The file must be in uncompressed format. (Specify the full path or the relative path.) If set to an empty string (""), then the bitmap is unset from the POS Printer.
<i>Width</i>	Printed width of the bitmap to be performed. (See PrintBitmap for values.)
<i>Alignment</i>	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** or **PrintImmediate** method with the print bitmap escape sequence in the print data.

The savable bitmap size is the *Width* of less than **RecLineWidth** dot (if *Alignment* is distance from the left-most print column, $Width + Alignment \leq RecLineWidth$). The height depends on the printer model. Please refer to [Printer Specification](#). If this condition is not satisfied, **SetBitmap** returns OPOS_EPTR_TOOBIG (206).

In this OCX, when SetBitmap is performed, the bitmap is saved to the nonvolatile memory in POS Printer. Therefore after performing **ReleaseDevice** method, the saved bitmap is valid. In other words, bitmap printing by escape sequence is possible after once saving the bitmap by **SetBitmap**.

The savable Windows bitmap file is 1/4/8/24bit and uncompressed format.

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

Return

One of the following values is returned by the method and placed in the **ResultCode** property.

Value	Meaning
OPOS_SUCCESS(0)	The method was successful .
OPOS_E_ILLEGAL(106)	One of the following errors occurred. <ul style="list-style-type: none">– <i>BitmapNumber</i> is invalid– <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
OPOS_E_NOEXIST(109)	<i>FileName</i> not found.
OPOS_E_FAILURE(111)	The bitmap data was not able to be transmitted to the POS Printer. Either the cover open, no receipt paper, or POS Printer power off may have occurred.
OPOS_E_BUSY(113)	Cannot be executed while output is in progress.
OPOS_E_EXTENDED(114)	ResultCodeExtended = OPOS_EPTR_TOOBIG(206): The bitmap is either too wide to print without transformation, or it is too big to transform. ResultCodeExtended = OPOS_EPTR_BADFORMAT(207): The specified file is either not a bitmap file, or it is in an unsupported format.

FTP2166000EQ

FTP-POS

POSPrinter OPOS Control Function specification

Other Values

See **ResultCode**.

SetLogo Method

Syntax

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

Parameter	Description
<i>Location</i>	The logo to be set. May be PTR_L_TOP (1) or PTR_L_BOTTOM (2).
<i>Data</i>	The characters that produce the logo. May consist of printable characters, escape sequences, carriage returns (13 decimal), and line feeds (10 decimal).

See **BinaryConversion** property for details.

Remarks

Saves 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.

Return

One of the following values is returned by the method and placed in the **ResultCode** property.

Value	Meaning
OPOS_SUCCESS(0)	The method was successful.
OPOS_E_BUSY(113)	Cannot be executed while output is in progress.
OPOS_E_ILLEGAL(106)	An invalid <i>Location</i> was specified.
Other Values	See ResultCode .

TransactionPrint Method

Syntax

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

Parameter	Description
<i>Station</i>	The printer station to be used. Set to PTR_S_RECEIPT(2).
<i>Control</i>	Transaction control. See values below.

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

Value	Meaning
PTR_TP_TRANSACTION(11)	Begin a transaction
PTR_TP_NORMAL(12)	End a transaction by printing the buffered data

Remarks

Enters or exits transaction mode.

If *control* is PTR_TP_TRANSACTION(11), then transaction mode is entered. Subsequent calls to **PrintNormal**, **CutPaper**, **MarkFeed**, **RotatePrint**, **PrintBarCode**, and **PrintBitmap** will buffer the print data at the Service object until **TransactionPrint** is called with the *Control* parameter set to PTR_TP_NORMAL(12). (In this case, the print methods only validate the method parameters and buffer the data – they do not initiate printing. Also, the value of the **AsyncMode** property does not affect their operation: No **OutputID** will be assigned to the request, nor will an **OutputCompleteEvent** be fired. 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 PTR_TP_NORMAL, then transaction mode is exited. If some data was buffered by calls to the methods **PrintNormal**, **CutPaper**, **RotatePrint**, **PrintBarCode**, **PrintBitmap**, **MarkFeed**, 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.

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

Printing by **RotatePrint** (Rotate printing mode), **PrintNormal**, and **RotatePrint**

(PTR_RP_PTR_RP_NORMAL (1)) is not performed until you execute the **TransactionPrint** method to exit the transaction mode. Also, when calling **RotatePrint** (Rotate printing mode) and **TransactionPrint** (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 **TransactionPrint** method.

Return

One of the values in the following table is returned by the method and placed in the **ResultCode** property.

Value	Meaning
OPOS_SUCCESS(0)	The method was successful.
OPOS_E_ILLEGAL(106)	The specified <i>Station</i> does not exist.
OPOS_E_NOHARDWARE(107)	The POS Printer is either off or offline.
OPOS_E_FAILURE(111)	OPOS control is in error state. Perform again after recovering from error condition.
OPOS_E_BUSY(113)	Cannot be executed while output is in progress. (Can only be returned if AsyncMode is FALSE and the <i>Control</i> parameter is PTR_TP_NORMAL(12).)
Other Values	See ResultCode .

ValidateData Method

Syntax

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

Parameter	Description
<i>Station</i>	The printer station to be used. Set to PTR_S_RECEIPT(2).
<i>Data</i>	The data to be validated. May include printable data and escape sequences. See BinaryConversion property for details.

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 POS Printer.

Even If this method returns OPOS_SUCCESS, it may not be supported depending on the printer. Please refer to [Printer Specification](#).

Return

One of the following values is returned by the method and placed in the **ResultCode** property.

Value	Meaning
OPOS_SUCCESS(0)	The data is valid.
OPOS_E_ILLEGAL(106)	Some of the escape sequences are not supported, but the Control can select valid alternatives.
OPOS_E_FAILURE(111)	Some of the data is not supported. No alternatives can be selected.
Other Values	See ResultCode .

Cases which cause OPOS_E_ILLEGAL(106) to be returned are:

Escape Sequence	Condition
Paper cut / Feed and Paper cut	The percentage '#' is not precisely supported.
Feed units	The feed count '#' is too large. (Valid from 0 to 255 only)
Feed reverse	The feed count '#' is too large. (Valid from 0 to 255 only)
Underline	The thickness '#' is not supported.
Scale vertically	The scaling factor '#' is not supported.
Scale horizontally	The scaling factor '#' is not supported.

Cases which will cause OPOS_E_FAILURE(111) to be returned are:

Escape Sequence	Condition
Feed, Paper cut, and Stamp	Not supported
Fire stamp	Not supported
Print bitmap	The bitmap number '#' is out of range. (Valid from 1 to 255 only)
Font typeface	Not supported.
Bold	Not supported.
Italic	Not supported.
Custom color	Not supported.
RED Color	Not supported.
RGB Color	Not supported.
Shading	Not supported.
Alternate color	Not supported.
SubScript / SuperScript	Not supported.

Data	Condition
<i>data1</i> CR <i>data2</i> LF	(Where CR is a Carriage Return and LF is a Line Feed.)

FTP2166000EQ

FTP-POS

POSPrinter OPOS Control Function specification

Not able to print data and remain on the same line. The data data1 will
print on one line, and the data data2 will print on the next line.

4.7. Events

DirectIOEvent Event

Syntax

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

Parameter	Description
<i>EventNumber</i>	Event number. Specific values are assigned by the Service Object.
<i>pData</i>	Pointer to numeric data.
<i>pString</i>	Pointer to string data.

EventNumber parameter is set to one of the following values.

Value	Meaning
OPOS_FCL_DIE_PTR_PRESENTER_ERROR(2)	Presenter is error.
OPOS_FCL_DIE_PTR_PRESENTER_OK(3)	Presenter is OK.
OPOS_FCL_DIE_PTR_LOW_BATTERY(4)	The battery is low.
OPOS_FCL_DIE_PTR_BATTERY_OK(5)	The battery has recovered from low.

Remarks

Provides Service Object information directly to the application.

OPOS_FCL_DIE_PTR_PRESENTER_ERROR

Parameter	Description
<i>EventNumber</i>	OPOS_FCL_DIE_PTR_PRESENTER_ERROR
<i>pData</i>	Not used.
<i>pString</i>	Not used.

Remarks

Notify when a presenter error occurs.

OPOS_FCL_DIE_PTR_PRESENTER_OK

Parameter	Description
<i>EventNumber</i>	OPOS_FCL_DIE_PTR_PRESENTER_OK
<i>pData</i>	Not used.
<i>pString</i>	Not used.

Remarks

Notify when recovering from the presenter error.

OPOS_FCL_DIE_PTR_LOW_BATTERY

Parameter	Description
<i>EventNumber</i>	OPOS_FCL_DIE_PTR_LOW_BATTERY
<i>pData</i>	Not used.
<i>pString</i>	Not used.

Remarks

Notify when low battery occurs.

OPOS_FCL_DIE_PTR_BATTERY_OK

Parameter	Description
<i>EventNumber</i>	OPOS_FCL_DIE_PTR_BATTERY_OK
<i>pData</i>	Not used.
<i>pString</i>	Not used.
Remarks	
Notify when recovering from the low battery.	

ErrorEvent Event

Syntax

```
void ErrorEvent (LONG ResultCode, LONG ResultCodeExtended, LONG ErrorLocus, LONG* pErrorResponse);
```

Parameter	Description
<i>ResultCode</i>	Result code causing the error event. See ResultCode for values.
<i>ResultCodeExtended</i>	Extended result code causing the error event. See values below.
<i>ErrorLocus</i>	OPOS_EL_OUTPUT(1) is set. Error occurs in asynchronous output.
<i>pErrorResponse</i>	Pointer to the error event response. See values below.

If *ResultCode* is OPOS_E_EXTENDED (114), *ResultCodeExtended* has one of the following values.

Value	Meaning
OPOS_EPTR_COVER_OPEN(201)	The printer cover is open.
OPOS_EPTR_REC_EMPTY(203)	The printer runs out of paper.
OPOS_FCL_EPTR_POWERSUPPLY(10001)	The power supply voltage of the printer is abnormal.
OPOS_FCL_EPTR_DATA(10002)	Transmission data is abnormal.
OPOS_FCL_EPTR_CUTTER(10003)	The cutter is abnormal.
OPOS_FCL_EPTR_HARDWARE(10004)	Hardware is abnormal.
OPOS_FCL_EPTR_HEADHOT(10005)	The temperature of thermal head is abnormal.
OPOS_FCL_EPTR_MARK(10006)	Mark can not be detected.
OPOS_FCL_EPTR_PRESENTER(10007)	Presenter is abnormal.

The contents assigned by the *pErrorResponse* are preset to a default value OPOS_ER_RETRY (11).

The application sets one of the following values.

Value	Meaning
OPOS_ER_RETRY(11)	Retry the asynchronous output. The error state is exited.
OPOS_ER_CLEAR(12)	Clear all buffered output data including all asynchronous output. (The effect is the same as when ClearOutput is called.).The error state is exited.

Remarks

Notified when an error is detected and the Control state changes to error state when the method is executing asynchronously.

OutputCompleteEvent Event

Syntax

void OutputCompleteEvent (LONG *OutputID*);

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

Remarks

Notified when a previously started asynchronous output request completes successfully.

StatusUpdateEvent Event

Syntax

void StatusUpdateEvent (LONG *Status*);

Remarks

Status parameter is set to one of the following values.

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_IDLE(1001)	All asynchronous output has finished, either successfully completed or deleted. The Printer State is now OPOS_S_IDLE(2). The FlagWhenIdle property must be TRUE for this event to be fired, and the Control automatically resets the property to FALSE just before firing the event.
OPOS_SUE_POWER_ONLINE(2001)	The device is powered on and ready for use. (Can be fired when PowerNotify = OPOS_PN_ENABLED (1).)
OPOS_SUE_POWER_OFF_OFFLINE(2004)	The device is either off or offline. (Can be fired when PowerNotify = OPOS_PN_ENABLED (1).)

Remarks

Notified when a significant device status change has occurred.

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

See Also

CapPowerReporting Property, **PowerNotify** Property

5. Log file output

FTP-POS POSPrinter OCX Control outputs the log file by setting **LogFolder**, **LogFileName**, and **LogLevel** in the registry or using the Configuration tool. The operation of log file output is as follows.

1. Create the log file according to the following naming rule.

LogFolder ="C:\Users\Public\FCL\OPOS\POSPrinter\FTP2166000EQ\Log\"

LogFileName ="FTP-POS" and date when this OCX was executed is YYYY / MM / DD, the log file is as follows :

C:\Users\Public\FCL\OPOS\POSPrinter\FTP2166000EQ\Log\FTP-POS[DD].log

Example: Output log file on 27th September 2016

C:\Users\Public\FCL\OPOS\POSPrinter\FTP2166000EQ\Log\FTP-POS27.log

2. If a file created by the naming rule in the 1st step already exists and the current month and the latest update month of the file are different, then the existing log file is deleted. In other case, the log file is newly created or additionally written.
3. As the results, the log files of the latest one month are pooled on the POS (PC). Upper bound of file size for each log file is not limited. If output the detailed logs, the log file size may be large, take care of the disk space on the POS (PC).
4. The following log file is output according to **LogLevel** setting.
 - LogLevel** = -1 : Outputs no log.
 - LogLevel** = 0 : Outputs normal trace log, warning, and error log.
 - LogLevel** = 1 : Outputs only error log.

•How to read the log file

LogLevel = 0

Example of log file:

<pre> 20160927:165025:Inf:Result : ResultCode is 0 20160927:165025:Inf:Exit : CPOSPrinter::PrintNormal 20160927:165025:Inf:Prop Get: ResultCode is 0 20160927:165025:Inf:Prop Get: State is 2 20160927:165025:Inf:Prop Get: ResultCodeExtended is 0 20160927:165025:Inf:Enter : CPOSPrinter::PrintNormal 20160927:165025:Inf:Enter : Param: Station is 2, Data is 123456789ABCDE </pre>	<p>Indicates the time the log is output.</p>
<pre> 20160927:165025:Inf:Send Data 20160927:165025:Inf:DoSendThreadSerial Addr : +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F ----- 0000 : 1B 61 00 31 32 33 34 35 36 37 38 39 41 42 43 44 a 0010 : 45 46 47 0A EFG </pre>	<p>“Enter” in the log indicates entering a function, and “Exit” indicates exiting a function. “Param” indicates the setting parameter and “ResultCode” indicates the result. When the OCX method is executed, “OCX Method” is output at the beginning.</p>
<pre> 20160927:165025:Inf:Result : ResultCode is 0 20160927:165025:Inf:Exit : CPOSPrinter::PrintNormal 20160927:165025:Inf:Prop Get: ResultCode is 0 20160927:165025:Inf:Prop Get: State is 2 20160927:165025:Inf:Prop Get: ResultCodeExtended is 0 20160927:165025:Inf:Enter : CPOSPrinter::PrintNormal 20160927:165025:Inf:Enter : Param: Station is 2, Data is 123456789ABCDE </pre>	<p>“Inf” indicates the general information log, and “Err” indicates an error.</p>
<pre> 20160927:165025:Inf:Send Data 20160927:165025:Inf:DoSendThread Addr : +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F ----- 0000 : 1B 61 00 31 32 33 34 35 36 37 38 39 41 42 43 44 a 0010 : 45 46 47 0A EFG </pre>	<p>“Prop Get” and “Prop Set” indicate that the value is acquired from the OCX property and is set in the OCX property.</p>
<pre> 20040119:165025:Inf:Default:Result : ResultCode is 0 20040119:165025:Inf:Default:Exit : CPOSPrinter::PrintNormal 20040119:165025:Inf:Default:Prop Get: ResultCode is 0 20040119:165025:Inf:Default:Prop Get: State is 2 20040119:165025:Inf:Default:Prop Get: ResultCodeExtended is 0 </pre>	<p>Indicates exchange of messages. The receive data and send data are binary-dumped.</p>

6. Printer Specification

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

6.1. FTP-627DSL440

6.1.1. Printable width

Printable width
432 dots
576 dots

6.1.2. Resolution

Resolution
203 dpi

6.1.3. 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

6.1.4. 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	✓	

FTP2166000EQ
FTP-POS
POSprinter OPOS Control Function specification

MaxiCode		
PDF417	✓	
GS1 Databar	✓	

6.1.5. Feature

(*The value if MapMode property is set to 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	
Right justify	Supported	
HRI of barcode	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	1920 dots	
Rotate right 90° and left 90°	Supported	
Rotate 180°	Supported	
Near end sensor	Supported	
RotateSpecial property	PTR_RP_NORMAL PTR_RP_RIGHT90 PTR_RP_LEFT90 PTR_RP_ROTATE180	
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	
RecBarCodeRotationList property	“0,R90,L90,180”	
RecBitmapRotationList property		
RecLineCharsList property	Printable width 576 dots: ”48,72” Printable width 432 dots: “36,54”	
RecLineWidth property	Printable width 576 dots:	

FTP2166000EQ

FTP-POS

POSPrinter OPOS Control Function specification

	576 Printable width 432 dots: 432	
Max of printing width of PrintBarCode method when a rotated 90° left or right mode	864 dots	
Update firmware and compare firmware version	Not supported	
The condition when OPOS_PS_OFF_OFFLINE is reported by PowerState property	USB-COM: Off or disconnected. RS-232C: Off, disconnected or when printer error occurs.	

6.1.6. Font

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

Printable width	RecLineChars property	RecLineHeight property	RecSidewaysMaxChars property	Font
576 dots	48	24	160(Fullwidth 80)	12x24 dots
	72	16	240(Fullwidth 120)	8x16 dots
432 dots	36	24	160(Fullwidth 80)	12x24 dots
	54	16	240(Fullwidth 120)	8x16 dots

6.1.7. Remarks

- Near end sensor is disabled by default on some printer models. To enable it, send FS 9 command or GS (E pL pH fn a b8 to b1 (fn = 3)) by DirectIO method with PTR_DIO_SEND_BINARY_DATA.
For more information on the command, refer to the product specifications or command specifications of the printer.
- Cutter is disabled by default on some printer models. To enable it, send GS (E pL pH fn a b8 to b1 (fn = 3)) by DirectIO method with PTR_DIO_SEND_BINARY_DATA.
For more information on the command, refer to the product specifications or command specifications of the printer.

6.2. FTP-629DSL310

6.2.1. Printable width

Printable width
432 dots
576 dots
640 dots

6.2.2. Resolution

Resolution
203 dpi

6.2.3. 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

6.2.4. 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	✓	
MaxiCode		
PDF417	✓	
GS1 Databar	✓	

6.2.5. Feature

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

Feature	Corresponding range	Remarks
Bold	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	
Right justify	Supported	
HRI of barcode	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	1920 dots	
Rotate right 90° and left 90°	Supported	
Rotate 180°	Supported	
Near end sensor	Supported	
RotateSpecial property	PTR_RP_NORMAL PTR_RP_RIGHT90 PTR_RP_LEFT90 PTR_RP_ROTATE180	
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	
RecBarCodeRotationList property	"0,R90,L90,180"	
RecBitmapRotationList property		
RecLineCharsList property	Printable width 640 dots: "53,80" Printable width 576 dots: "48,72" Printable width 432 dots: "36,54"	
RecLineWidth property	Printable width 640 dots: 640 Printable width 576 dots: 576	

FTP2166000EQ

FTP-POS

POSPrinter OPOS Control Function specification

	Printable width 432 dots: 432	
Max of printing width of PrintBarCode method when a rotated 90° left or right mode	864 dots	
Update firmware and compare firmware version	Not supported	
The condition when OPOS_PS_OFF_OFFLINE is reported by PowerState property	USB-COM: Off or disconnected. RS-232C: Off, disconnected or when printer error occurs.	

6.2.6. Font

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

Printable width	RecLineChars property	RecLineHeight property	RecSidewaysMaxChars property	Font
640 ドット	53	24	144 (Fullwidth 72)	12 × 24 dots
	80	16	216 (Fullwidth 108)	8 × 16 dots
576 dots	48	24	160 (Fullwidth 80)	12x24 dots
	72	16	240 (Fullwidth 120)	8x16 dots
432 dots	36	24	160 (Fullwidth 80)	12x24 dots
	54	16	240 (Fullwidth 120)	8x16 dots

6.2.7. Remarks

- Near end sensor is disabled by default on some printer models. To enable it, send FS 9 command or GS (E pL pH fn a b8 to b1 (fn = 3)) by DirectIO method with PTR_DIO_SEND_BINARY_DATA.
For more information on the command, refer to the product specifications or command specifications of the printer.
- Cutter is disabled by default on some printer models. To enable it, send GS (E pL pH fn a b8 to b1 (fn = 3)) by DirectIO method with PTR_DIO_SEND_BINARY_DATA.
For more information on the command, refer to the product specifications or command specifications of the printer.

6.3. FTP-629DSL350

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

6.3.1. Printable width

Printable width
576 dots

6.3.2. Resolution

Resolution
203 dpi

6.3.3. 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

6.3.4. 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		

6.3.5. Feature

(*The value if MapMode property is set to 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	Supported only with firmware of V1.05 or higher.
Right justify	Supported	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	PTR_RP_NORMAL PTR_RP_ROTATE180	
Max printable height of PrintBitmap method	1023 dots	
Max savable height of SetBitmap method	512 dots	
RecBarCodeRotationList property	“0,,180”	
RecBitmapRotationList property		
RecLineCharsList property	Printable width 576 dots: ”48,72”	
RecLineWidth property	Printable width 576 dots: 576	
Max of printing width of PrintBarCode method when a rotated 90° left or right mode	Not supported	
Update firmware and	Not supported	

FTP2166000EQ

FTP-POS

POSPrinter OPOS Control Function specification

compare firmware version		
The condition when OPOS_PS_OFF_OFFLINE is reported by PowerState property	USB-COM: Off or disconnected. RS-232C: Off, disconnected or when a printer error occurs	

6.3.6. Font

(*The value if the MapMode property is set to 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

6.3.7. 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
PrintImmediate method
PrintNormal method
SetBitmap method
- When using the following escape sequence with PrintNormal method, PrintImmediate 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

6.4. FTP-839DSL310

6.4.1. Printable width

Printable width
960 dots

6.4.2. Resolution

Resolution
300 dpi

6.4.3. Paper type

Type	Paper
1	Unused
2	Unused
3	Unused
4	PD150R
5	Unused
6	Unused
7	Unused
8	Unused
9	Unused
10	Unused
11	Unused
12	Unused

6.4.4. 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	✓	
MaxiCode		
PDF417	✓	
GS1 Databar	✓	

6.4.5. Feature

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

Feature	Corresponding range	Remarks
Bold	Supported	
Reverse video	Supported	
Scale horizontally	1 – 4	
Scale vertically	1 – 4	
Underline	1 – 2	
Paper cut	Not supported	
Feed and Paper cut	Not supported	
Print bitmap by escape sequence	Supported	Only the model equipped with the extended nonvolatile memory.
Center	Supported	
Right justify	Supported	
HRI of barcode	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	1152 dots	
Rotate right 90° and left 90°	Supported	
Rotate 180°	Supported	
Near end sensor	Supported	
RotateSpecial property	PTR_RP_NORMAL PTR_RP_RIGHT90 PTR_RP_LEFT90 PTR_RP_ROTATE180	
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	
RecBarCodeRotationList property	"0,R90,L90,180"	
RecBitmapRotationList property		
RecLineCharsList property	"80,120"	
RecLineWidth property	960	
Max of printing width of PrintBarCode method when a rotated 90° left or right mode	864 dots	
Update firmware and compare firmware version	Not supported	
The condition when OPOS_PS_OFF_OFFLINE is reported by PowerState	USB-COM: Off or disconnected. RS-232C: Off, disconnected	

FTP2166000EQ

FTP-POS

POSPrinter OPOS Control Function specification

property	or when printer error occurs.	
----------	-------------------------------	--

6.4.6. Font

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

Printable width	RecLineChars property	RecLineHeight property	RecSidewaysMaxChars property	Font
960 dots	80	24	96 (Fullwidth 48)	12 × 24 dots
	120	16	144 (Fullwidth 72)	8 × 16 dots

6.4.7. Remarks

- Near end sensor is disabled by default on some printer models. To enable it, send FS 9 command or GS (E pL pH fn a b8 to b1 (fn = 3)) by DirectIO method with PTR_DIO_SEND_BINARY_DATA.
For more information on the command, refer to the product specifications or command specifications of the printer.

6.5. 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.

6.5.1. Printable width

Printable width
432 dots
576 dots
832 dots

6.5.2. Resolution

Resolution
203 dpi

6.5.3. 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

6.5.4. 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.

6.5.5. Feature

(*The value if MapMode property is set to 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	Supporting is different depending on the FW version. Please refer to *1 .
Right justify	Supported	Supporting is different depending on the FW version. Please refer to *1 .
HRI of barcode	Supported	Supporting is different depending on the FW version. Please 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. Please refer to *1 .
Rotate right 90° and left 90°	Supported	Supporting is different depending on the FW version. Please refer to *1 . Only the model equipped with the extended volatile memory.
Rotate 180°	Supported	
Near end sensor	Supported	
RotateSpecial property	PTR_RP_NORMAL PTR_RP_RIGHT90 PTR_RP_LEFT90 PTR_RP_ROTATE180	Supporting of PTR_RP_RIGHT90 and PTR_RP_LEFT90 is different depending on the FW version. Please refer to *1 . If the extended volatile memory is not equipped, PTR_RP_RIGHT90 and PTR_RP_LEFT90 are not supported.
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	512 dots	

SetBitmap method		
RecBarCodeRotationList property	"0,R90,L90,180"	Supporting of R90 and L90 is different depending on the FW version. Please refer to *1 . If the extended volatile memory is not equipped, R90 and L90 are not supported.
RecBitmapRotationList property		
RecLineCharsList property	Printable width 832 dots: "69, 104" Printable width 576 dots: "48,72" Printable width 432 dots: "36,54"	
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	
Update firmware and compare firmware version	Supported	
The condition when OPOS_PS_OFF_OFFLINE is reported by PowerState property	USB-COM: Off or disconnected. RS-232C: Off, disconnected or when printer error occurs.	

6.5.6. Font

(*The value if the MapMode property is set to 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

*1) Supported FW version is as follows.

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

6.6. FTP-63GDSL483/FTP-83GDSL483

6.6.1. Printable width

Printable width
FTP-63GDSL483: 640 dots
FTP-83GDSL483: 960 dots

6.6.2. Resolution

Resolution
FTP-63GDSL483: 203dpi
FTP-83GDSL483: 300dpi

6.6.3. Paper type

Type	Paper
1	Unused
2	Unused
3	Unused
4	PD150R PD450-145
5	Unused
6	Unused
7	Unused
8	Unused
9	Unused
10	Unused
11	Unused
12	Unused

6.6.4. 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	✓	
MaxiCode		
PDF417	✓	
GS1 Databar	✓	

6.6.5. Feature

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

Feature	Corresponding range	Remarks
Bold	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	
Right justify	Supported	
HRI of barcode	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	FTP-63G: 1728 dots FTP-83G: 1152 dots	
Rotate right 90° and left 90°	Supported	
Rotate 180°	Supported	
Near end sensor	Supported	
RotateSpecial property	PTR_RP_NORMAL PTR_RP_RIGHT90 PTR_RP_LEFT90 PTR_RP_ROTATE180	
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	
RecBarCodeRotationList property	"0,R90,L90,180"	
RecBitmapRotationList property		
RecLineCharsList property	Printable width 960 dots: "80, 120" Printable width 640 dots: "63,80"	
RecLineWidth property	Printable width 960 dots: 960 Printable width 640 dots: 640	
Max of printing width of	864 dots	

FTP2166000EQ

FTP-POS

POSPrinter OPOS Control Function specification

PrintBarCode method when a rotated 90° left or right mode		
Update firmware and compare firmware version	Not supported	
The condition when OPOS_PS_OFF_OFFLINE is reported by PowerState property	USB-COM: Off or disconnected. RS-232C: Off, disconnected or when printer error occurs.	

6.6.6. Font

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

Printable width	RecLineChars property	RecLineHeight property	RecSidewaysMaxChars property	Font
960 dots	80	24	96 (Fullwidth 48)	12 × 24 dots
	120	16	144 (Fullwidth 72)	8 × 16 dots
640 dots	53	24	144 (Fullwidth 72)	12 × 24 dots
	80	16	216 (Fullwidth 108)	8 × 16 dots

6.6.7. Remarks

- Near end sensor is disabled by default on some printer models. To enable it, send FS 9 command or GS (E pL pH fn a b8 to b1 (fn = 3)) by DirectIO method with PTR_DIO_SEND_BINARY_DATA.
For more information on the command, refer to the product specifications or command specifications of the printer.
- Cutter is disabled by default on some printer models. To enable it, send GS (E pL pH fn a b8 to b1 (fn = 3)) by DirectIO method with PTR_DIO_SEND_BINARY_DATA.
For more information on the command, refer to the product specifications or command specifications of the printer.

6.7. FTP-62HWSL001

6.7.1. Printable width

Printable width
384 dots

6.7.2. Paper type

Type	Paper
1	Unused
2	Unused
3	Unused
4	PD150R
5	TP50KJ-R
6	Unused
7	Unused
8	HW54S
9	Unused
10	Unused
11	Unused
12	Unused

6.7.3. Resolution

Resolution
203 dpi

6.7.4. 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	✓	
MaxiCode	✓	
PDF417	✓	
GS1 Databar	✓	

6.7.5. Feature

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

Feature	Corresponding range	Remarks
Bold	Supported	Supported only with firmware of V1.01 or higher.
Reverse video	Supported	
Scale horizontally	1 – 8	It does not correspond to 0.5 unit.
Scale vertically	1 – 8	It does not correspond to 0.5 unit.
Underline	1 – 2	
Paper cut	Not supported	
Feed and Paper cut	Not supported	
Print bitmap by escape sequence	Supported	
Center	Not supported	
Right justify	Not supported	
HRI of barcode	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	1920 dots	
Rotate right 90° and left 90°	Supported	
Near end sensor	Not supported	
RotateSpecial property	PTR_RP_NORMAL PTR_RP_RIGHT90 PTR_RP_LEFT90 PTR_RP_ROTATE180	
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	512dots	
RecBarCodeRotationList property	"0,R90,L90,180"	
RecBitmapRotationList property		
RecLineCharsList property	"32,48"	
RecLineWidth property	384	
Max of printing width of PrintBarCode method when a rotated 90° left or right mode	864 dots	
Update firmware and compare firmware version	Not supported	
The condition when OPOS_PS_OFF_OFFLINE is reported by PowerState property	USB-COM: Off, disconnected or when printer error occurs. Bluetooth: When printer error occurs.	

FTP2166000EQ

FTP-POS

POSPrinter OPOS Control Function specification

6.7.6. Font

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

Printable width	RecLineChars property	RecLineHeight property	RecSidewaysMaxChars property	Font
384 dots	32	24	160 (Fullwidth 80)	12x24 dots
	48	16	240 (Fullwidth 120)	8x16 dots

6.7.7. Remarks

- At the time of execution of ClaimDevice method, In addition to the time specified in *Timeout* parameter, there is a maximum connection waiting time of approximately 3000 milliseconds.
- When using Bluetooth interface, automatic reconnection is not performed when returning from off or disconnection. Please reconnect using ReleaseDevice and ClaimDevice method.
- If undetected mark error occurs, it is returned by either removing the cause of the paper jam or closing the cover.
- When OPOS_CH_INTERNAL is specified in *Level* parameter of CheckHealth method, CheckHealthText property is "Internal HCheck: Successful" regardless of the connection or power state.

6.8. FTP-62HDSL100

6.8.1. Printable width

Printable width
384 dots
576 dots
832 dots

6.8.2. Resolution

Resolution
203 dpi

6.8.3. 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

6.8.4. 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	✓	
MaxiCode	✓	
PDF417	✓	
GS1 Databar	✓	

6.8.5. Feature

(*The value if MapMode property is set to 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	
Right justify	Supported	
HRI of barcode	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	1144 dots	
Rotate right 90° and left 90°	Supported	
Rotate 180°	Supported	
Near end sensor	Supported	
RotateSpecial property	PTR_RP_NORMAL PTR_RP_RIGHT90 PTR_RP_LEFT90 PTR_RP_ROTATE180	
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	
RecBarCodeRotationList property	"0,R90,L90,180"	
RecBitmapRotationList property		
RecLineCharsList property	Printing width 832 dots: "69,104" Printable width 576 dots: "48,72" Printable width 384 dots: "32,48"	
RecLineWidth property	Printing width 832 dots: 832 Printable width 576 dots: 576	

FTP2166000EQ

FTP-POS

POSPrinter OPOS Control Function specification

	Printable width 384 dots: 384	
Max of printing width of PrintBarCode method when a rotated 90° left or right mode	864 dots	
Update firmware and compare firmware version	Not supported	
The condition when OPOS_PS_OFF_OFFLINE is reported by PowerState property	USB-COM: Off or disconnected. RS-232C: Off, disconnected or when printer error occurs.	

6.8.6. Font

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

Printable width	RecLineChars property	RecLineHeight property	RecSidewaysMaxChars property	Font
832 dots	69	24	95(Fullwidth 47)	12x24 dots
	104	16	143(Fullwidth 71)	8x16 dots
576 dots	48	24	160(Fullwidth 80)	12x24 dots
	72	16	240(Fullwidth 120)	8x16 dots
432 dots	36	24	160(Fullwidth 80)	12x24 dots
	54	16	240(Fullwidth 120)	8x16 dots

6.8.7. Remarks

- Near end sensor is disabled by default on some printer models. To enable it, send FS 9 command or GS (E pL pH fn a b8 to b1 (fn = 3)) by DirectIO method with PTR_DIO_SEND_BINARY_DATA.
For more information on the command, refer to the product specifications or command specifications of the printer.

6.9. FTP-62EDSL200

6.9.1. Printable width

Printable width
1152 dots
1696 dots

6.9.2. Resolution

Resolution
203 dpi

6.9.3. Paper type

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

6.9.4. 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		
MaxiCode		
PDF417		
GS1 Databar		

6.9.5. Feature

(*The value if MapMode property is set to 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	Not supported	
Feed and Paper cut	Not supported	
Print bitmap by escape sequence	Not supported	
Center	Not supported	
Right justify	Not supported	
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	–	
Rotate right 90° and left 90°	Not supported	
Rotate 180°	Supported	
Near end sensor	Supported	
RotateSpecial property	PTR_RP_NORMAL PTR_RP_ROTATE180	
Max printable height of PrintBitmap method	1023 dots	
Max savable height of SetBitmap method	–	
RecBarCodeRotationList property	“0,R90,L90,180”	
RecBitmapRotationList property		
RecLineWidth property	Printing width 1152 dots: 1152 Printable width 696 dots: 1696	
Max of printing width of PrintBarCode method when a rotated 90° left or right mode	–	
Update firmware and compare firmware version	Not supported	
The condition when OPOS_PS_OFF_OFFLINE is reported by PowerState property	USB-COM: Off or disconnected. RS-232C: Off, disconnected or when printer error occurs.	

FTP2166000EQ

FTP-POS

POSPrinter OPOS Control Function specification

6.9.6. Font

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

Printable width	RecLineChars property	RecLineHeight property	RecSidewaysMaxChars property	Font
1696 dots	96	24	–	12x24 dots
	144	16	–	8x16 dots
1152 dots	96	24	–	12x24 dots
	144	16	–	8x16 dots

6.9.7. Remarks

- Near end sensor is disabled by default on some printer models. To enable it, send FS 9 command or GS (E pL pH fn a b8 to b1 (fn = 3)) by DirectIO method with PTR_DIO_SEND_BINARY_DATA.

For more information on the command, refer to the product specifications or command specifications of the printer.

7. POS for .NET

This OPOS Control can be used on POS for .NET.

7.1. Install

7.1.1. Installation condition

■ Operation environment

- OS : Microsoft Windows 7 Service Pack 1 / 8.1 / 10 / 11 32-bits / 64-bits
- CPU : System requirements for each operating system.
- RAM : 各 System requirements for each operating system.
- Storage : Space 10MB or more
- Microsoft .NET Framework 4.0 or later

7.1.2. Installation procedure

When installing, logon by an account with administrator privileges.

1. Install Microsoft POS for .NET
Download Microsoft Point of Service for .NET v1.14.1 from the website below, and then install it.
<https://www.microsoft.com/en-us/download/details.aspx?id=55758>
2. Install OPOS Common Control Object
Download 1.14.001 CCO Installer from the website below, and then install it.
http://monroeccs.com/oposccos_current.htm
3. Install OPOS Control
Refer to [2.4. Installation procedure] to install OPOS Control and configure the POS Printer by configuration tool.
4. Install plugin for 64-bits OS
If OS is 64-bits, install POSforDotNETPluginX64.msi.

7.1.3. Uninstallation procedure

1. Open the control panel, then launch "Programs and Features".
2. Uninstall plugin for 64-bits OS
If OS is 64-bits, uninstall FTP2166000EQ POS for .NET Plugin x64.
If you have installed a third-party OPOS that uses the CCO, uninstall the 64-bit OS plug-in, then uninstall the CCO, and then reinstall the CCO.
3. Uninstall OPOS Control
Refer to [0Uninstallation procedure] to uninstall OPOS Control.
4. Uninstall OPOS Common Control Object
Uninstall OposCCOs.
If you have a third-party OPOS that uses the CCO installed, do not uninstall it.
5. Uninstall Microsoft POS for .NET
Uninstall Microsoft POS for .NET 1.14.1.
If you have a third-party software that uses the POS for .NET installed, do not uninstall it.

7.2. Attentions

It may not work properly in an environment where OPOS or POS for .NET products made by third-party are installed.
POS for .NET specific features are not supported.

When Retrieves the device POSExplorer API, specify the parameter compatibility of POSExplorer.GetDevices method to DeviceCompatibilities.Opos or DeviceCompatibilities.OposAndCompatibilityLevel1.

Features not defined in OPOS V1.9 cannot be used. Please refer to [4. OPOS Interface specifications (Printer)].

Plug and Play is not supported.

7.3. Difference from OPOS

7.3.1. Constants and enumerations

The correspondence between OPOS constants and POS for .NET enumerations is as follows.

Some are constants, not enumerations.

It also contains constants and enumerations that are not available in this product.

OPOS constants	POS for .NET enumerations		
	Class name	Parameter	
		Type	Name
OPOS_S_CLOSED	ControlState	enum	Closed
OPOS_S_IDLE	ControlState	enum	Idle
OPOS_S_BUSY	ControlState	enum	Busy
OPOS_S_ERROR	ControlState	enum	Error
OPOS_SUCCESS	ErrorCode	enum	Success
	ErrorCode	enum	Unspecified
OPOS_E_CLOSED	ErrorCode	enum	Closed
OPOS_E_CLAIMED	ErrorCode	enum	Claimed
OPOS_E_NOTCLAIMED	ErrorCode	enum	NotClaimed
OPOS_E_NOSERVICE	ErrorCode	enum	NoService
OPOS_E_DISABLED	ErrorCode	enum	Disabled
OPOS_E_ILLEGAL	ErrorCode	enum	Illegal
OPOS_E_NOHARDWARE	ErrorCode	enum	NoHardware
OPOS_E_OFFLINE	ErrorCode	enum	Offline
OPOS_E_NOEXIST	ErrorCode	enum	NoExist
OPOS_E_EXISTS	ErrorCode	enum	Exists
OPOS_E_FAILURE	ErrorCode	enum	Failure
OPOS_E_TIMEOUT	ErrorCode	enum	Timeout
OPOS_E_BUSY	ErrorCode	enum	Busy
OPOS_E_EXTENDED	ErrorCode	enum	Extended
OPOS_CH_INTERNAL	HealthCheckLevel	enum	Internal
OPOS_CH_EXTERNAL	HealthCheckLevel	enum	External
OPOS_CH_INTERACTIVE	HealthCheckLevel	enum	Interactive
OPOS_PR_NONE	PowerReporting	enum	None
OPOS_PR_STANDARD	PowerReporting	enum	Standard
OPOS_PR_ADVANCED	PowerReporting	enum	Advanced
OPOS_PN_DISABLED	PowerNotification	enum	Disabled
OPOS_PN_ENABLED	PowerNotification	enum	Enabled
OPOS_PS_UNKNOWN	PowerState	enum	Unknown
OPOS_PS_ONLINE	PowerState	enum	Online
OPOS_PS_OFF	PowerState	enum	Off
OPOS_PS_OFFLINE	PowerState	enum	Offline
OPOS_PS_OFF_OFFLINE	PowerState	enum	OffOffline

FTP2166000EQ

FTP-POS

POSPrinter OPOS Control Function specification

OPOS constants	POS for .NET enumerations		
	Class name	Parameter	
		Type	Name
OPOS_EL_OUTPUT	ErrorLocus	enum	Output
OPOS_EL_INPUT	ErrorLocus	enum	Input
OPOS_EL_INPUT_DATA	ErrorLocus	enum	InputData
OPOS_ER_RETRY	ErrorResponse	enum	Retry
OPOS_ER_CLEAR	ErrorResponse	enum	Clear
OPOS_ER_CONTINUEINPUT	ErrorResponse	enum	ContinueInput
OPOS_SUE_POWER_ONLINE	PosCommon	int	StatusPowerOnline
OPOS_SUE_POWER_OFF	PosCommon	int	StatusPowerOff
OPOS_SUE_POWER_OFFLINE	PosCommon	int	StatusPowerOffline
OPOS_SUE_POWER_OFF_OFFLINE	PosCommon	int	StatusPowerOffOffline
OPOS_FOREVER	PosCommon	int	WaitForever
OPOS_BC_NONE	BinaryConversion	enum	None
OPOS_BC_NIBBLE	BinaryConversion	enum	Nibble
OPOS_BC_DECIMAL	BinaryConversion	enum	Decimal
	PrinterStation	enum	None
PTR_S_JOURNAL	PrinterStation	enum	Journal
PTR_S_RECEIPT	PrinterStation	enum	Receipt
PTR_S_SLIP	PrinterStation	enum	Slip
PTR_S_JOURNAL_RECEIPT	PrinterStation	enum	TwoReceiptJournal
PTR_S_JOURNAL_SLIP	PrinterStation	enum	TwoSlipJournal
PTR_S_RECEIPT_SLIP	PrinterStation	enum	TwoSlipReceipt
PTR_TWO_RECEIPT_JOURNAL	PrinterStation	enum	TwoReceiptJournal
PTR_TWO_SLIP_JOURNAL	PrinterStation	enum	TwoSlipJournal
PTR_TWO_SLIP_RECEIPT	PrinterStation	enum	TwoSlipReceipt
	CharacterSetCapability	enum	Numeric
PTR_CCS_ALPHA	CharacterSetCapability	enum	Alpha
PTR_CCS_ASCII	CharacterSetCapability	enum	Ascii
PTR_CCS_KANA	CharacterSetCapability	enum	Kana
PTR_CCS_KANJI	CharacterSetCapability	enum	Kanji
PTR_CCS_UNICODE	CharacterSetCapability	enum	Unicode
PTR_CS_UNICODE	PosPrinter	System.Int32	CharacterSetUnicode
PTR_CS_ASCII	PosPrinter	System.Int32	CharacterSetAscii
PTR_CS_WINDOWS		System.Int32	Specify 999
PTR_CS_ANSI	PosPrinter	System.Int32	CharacterSetAnsi

OPOS constants	POS for .NET enumerations		
	Class name	Parameter	
		Type	Name
PTR_EL_NONE	PrinterErrorLevel	enum	None
PTR_EL_RECOVERABLE	PrinterErrorLevel	enum	Recoverable
PTR_EL_FATAL	PrinterErrorLevel	enum	Fatal
PTR_MM_DOTS	MapMode	enum	Dots
PTR_MM_TWIPS	MapMode	enum	Twips
PTR_MM_ENGLISH	MapMode	enum	English
PTR_MM_METRIC	MapMode	enum	Metric
	PrinterColors	enum	None
PTR_COLOR_PRIMARY	PrinterColors	enum	Primary
PTR_COLOR_CUSTOM1	PrinterColors	enum	Custom1
PTR_COLOR_CUSTOM2	PrinterColors	enum	Custom2
PTR_COLOR_CUSTOM3	PrinterColors	enum	Custom3
PTR_COLOR_CUSTOM4	PrinterColors	enum	Custom4
PTR_COLOR_CUSTOM5	PrinterColors	enum	Custom5
PTR_COLOR_CUSTOM6	PrinterColors	enum	Custom6
PTR_COLOR_CYAN	PrinterColors	enum	Cyan
PTR_COLOR_MAGENTA	PrinterColors	enum	Magenta
PTR_COLOR_YELLOW	PrinterColors	enum	Yellow
PTR_COLOR_FULL	PrinterColors	enum	Full
PTR_CART_UNKNOWN	PrinterCartridgeStates	enum	Unknown
PTR_CART_OK	PrinterCartridgeStates	enum	OK
PTR_CART_REMOVED	PrinterCartridgeStates	enum	Removed
PTR_CART_EMPTY	PrinterCartridgeStates	enum	Empty
PTR_CART_NEAREND	PrinterCartridgeStates	enum	NearEnd
PTR_CART_CLEANING	PrinterCartridgeStates	enum	Cleaning
PTR_CN_DISABLED	PrinterCartridgeNotify	enum	Disabled
PTR_CN_ENABLED	PrinterCartridgeNotify	enum	Enabled
PTR_CP_FULLCUT	PosPrinter	System.Int32	PrinterCutPaperFullCut
PTR_BC_LEFT	PosPrinter	System.Int32	PrinterBarCodeLeft
PTR_BC_CENTER	PosPrinter	System.Int32	PrinterBarCodeCenter
PTR_BC_RIGHT	PosPrinter	System.Int32	PrinterBarCodeRight
PTR_BC_TEXT_NONE	BarCodeTextPosition	enum	None
PTR_BC_TEXT_ABOVE	BarCodeTextPosition	enum	Above
PTR_BC_TEXT_BELOW	BarCodeTextPosition	enum	Below
	BarCodeSymbology	enum	Unknown
PTR_BCS_UPCA	BarCodeSymbology	enum	Upca
PTR_BCS_UPCE	BarCodeSymbology	enum	Upce

OPOS constants	POS for .NET enumerations		
	Class name	Parameter	
		Type	Name
PTR_BCS_JAN8	BarCodeSymbology	enum	EanJan8
PTR_BCS_EAN8			
PTR_BCS_JAN13	BarCodeSymbology	enum	EanJan13
PTR_BCS_EAN13			
PTR_BCS_TF	BarCodeSymbology	enum	TF
PTR_BCS_ITF	BarCodeSymbology	enum	Itf
PTR_BCS_Codabar	BarCodeSymbology	enum	Codabar
PTR_BCS_Code39	BarCodeSymbology	enum	Code39
PTR_BCS_Code93	BarCodeSymbology	enum	Code93
PTR_BCS_Code128	BarCodeSymbology	enum	Code128
PTR_BCS_UPCA_S	BarCodeSymbology	enum	UpcaS
PTR_BCS_UPCE_S	BarCodeSymbology	enum	Upces
PTR_BCS_UPCD1	BarCodeSymbology	enum	Upcd1
PTR_BCS_UPCD2	BarCodeSymbology	enum	Upcd2
PTR_BCS_UPCD3	BarCodeSymbology	enum	Upcd3
PTR_BCS_UPCD4	BarCodeSymbology	enum	Upcd4
PTR_BCS_UPCD5	BarCodeSymbology	enum	Upcd5
PTR_BCS_EAN8_S	BarCodeSymbology	enum	Ean8S
PTR_BCS_EAN13_S	BarCodeSymbology	enum	Ean13S
PTR_BCS_EAN128	BarCodeSymbology	enum	Ean128
PTR_BCS_OCRA	BarCodeSymbology	enum	Ocra
PTR_BCS_OCRB	BarCodeSymbology	enum	Ocrb
PTR_BCS_Code128_Parsed	BarCodeSymbology	enum	Rss14
PTR_BCS_RSS14	BarCodeSymbology	enum	RssExpanded
PTR_BCS_RSS_EXPANDED	BarCodeSymbology	enum	Cca
	BarCodeSymbology	enum	Ccb
	BarCodeSymbology	enum	Ccc
PTR_BCS_PDF417	BarCodeSymbology	enum	Pdf417
PTR_BCS_MAXICODE	BarCodeSymbology	enum	Maxicode
PTR_BCS_OTHER	BarCodeSymbology	enum	Other
PTR_BM_ASIS	PosPrinter	System.Int32	PrinterBitmapAsIs
PTR_BM_LEFT	PosPrinter	System.Int32	PrinterBitmapLeft
PTR_BM_CENTER	PosPrinter	System.Int32	PrinterBitmapCenter
PTR_BM_RIGHT	PosPrinter	System.Int32	PrinterBitmapRight
PTR_RP_NORMAL	PrintRotation	enum	Normal
PTR_RP_RIGHT90	PrintRotation	enum	Right90
PTR_RP_LEFT90	PrintRotation	enum	Left90
PTR_RP_ROTATE180	PrintRotation	enum	Rotate180
PTR_RP_BARCODE	PrintRotation	enum	Barcode
PTR_RP_BITMAP	PrintRotation	enum	Bitmap
PTR_L_TOP	PrinterLogoLocation	enum	Top

OPOS constants	POS for .NET enumerations		
	Class name	Parameter	
		Type	Name
PTR_L_BOTTOM	PrinterLogoLocation	enum	Bottom
PTR_TP_TRANSACTION	PrinterTransactionControl	enum	Transaction
PTR_TP_NORMAL	PrinterTransactionControl	enum	Normal
	PrinterMarkFeeds	enum	None
PTR_MF_TO_TAKEUP	PrinterMarkFeeds	enum	Takeup
PTR_MF_TO_CUTTER	PrinterMarkFeeds	enum	Cutter
PTR_MF_TO_CURRENT_TOF	PrinterMarkFeeds	enum	CurrentTopOfForm
PTR_MF_TO_NEXT_TOF	PrinterMarkFeeds	enum	NextTopOfForm
PTR_PS_UNKNOWN	PrinterSide	enum	Unknown
PTR_PS_SIDE1	PrinterSide	enum	Side1
PTR_PS_SIDE2	PrinterSide	enum	Side2
PTR_PS_OPPOSITE	PrinterSide	enum	Opposite
PTR_SUE_COVER_OPEN	PrinterStatus	enum	CoverOpen
PTR_SUE_COVER_OK	PrinterStatus	enum	CoverOK
PTR_SUE_JRN_EMPTY	PrinterStatus	enum	JournalEmpty
PTR_SUE_JRN_NEAREMPTY	PrinterStatus	enum	JournalNearEmpty
PTR_SUE_JRN_PAPEROK	PrinterStatus	enum	JournalPaperOK
PTR_SUE_REC_EMPTY	PrinterStatus	enum	ReceiptEmpty
PTR_SUE_REC_NEAREMPTY	PrinterStatus	enum	ReceiptNearEmpty
PTR_SUE_REC_PAPEROK	PrinterStatus	enum	ReceiptPaperOK
PTR_SUE_SLP_EMPTY	PrinterStatus	enum	SlipEmpty
PTR_SUE_SLP_NEAREMPTY	PrinterStatus	enum	SlipNearEmpty
PTR_SUE_SLP_PAPEROK	PrinterStatus	enum	SlipPaperOK
PTR_SUE_JRN_CARTRIDGE_EMPTY	PrinterStatus	enum	JournalCartridgeEmpty
PTR_SUE_JRN_CARTRIDGE_NEAREMPTY	PrinterStatus	enum	JournalCartridgeNearEmpty
PTR_SUE_JRN_HEAD_CLEANING	PrinterStatus	enum	JournalHeadCleaning
PTR_SUE_JRN_CARTRIDGE_OK	PrinterStatus	enum	JournalCartridgeOK
PTR_SUE_REC_CARTRIDGE_EMPTY	PrinterStatus	enum	ReceiptCartridgeEmpty
PTR_SUE_REC_CARTRIDGE_NEAREMPTY	PrinterStatus	enum	ReceiptCartridgeNearEmpty
PTR_SUE_REC_HEAD_CLEANING	PrinterStatus	enum	ReceiptHeadCleaning
PTR_SUE_REC_CARTRIDGE_OK	PrinterStatus	enum	ReceiptCartridgeOK
PTR_SUE_SLP_CARTRIDGE_EMPTY	PrinterStatus	enum	SlipCartridgeEmpty
PTR_SUE_SLP_CARTRIDGE_NEAREMPTY	PrinterStatus	enum	SlipCartridgeNearEmpty
PTR_SUE_SLP_HEAD_CLEANING	PrinterStatus	enum	SlipHeadCleaning
PTR_SUE_SLP_CARTRIDGE_OK	PrinterStatus	enum	SlipCartridgeOK
PTR_SUE_JRN_COVER_OPEN	PrinterStatus	enum	JournalCoverOpen
PTR_SUE_JRN_COVER_OK	PrinterStatus	enum	JournalCoverOK
PTR_SUE_REC_COVER_OPEN	PrinterStatus	enum	ReceiptCoverOpen
PTR_SUE_REC_COVER_OK	PrinterStatus	enum	ReceiptCoverOK

FTP2166000EQ

FTP-POS

POSPrinter OPOS Control Function specification

OPOS constants	POS for .NET enumerations		
	Class name	Parameter	
		Type	Name
PTR_SUE_SLP_COVER_OPEN	PrinterStatus	enum	SlipCoverOpen
PTR_SUE_SLP_COVER_OK	PrinterStatus	enum	SlipCoverOK
PTR_SUE_IDLE	PrinterStatus	enum	Idle
OPOS_EPTR_COVER_OPEN	PosPrinter	System.Int32	ExtendedErrorCoverOpen
OPOS_EPTR_JRN_EMPTY	PosPrinter	System.Int32	ExtendedErrorJrnEmpty
OPOS_EPTR_REC_EMPTY	PosPrinter	System.Int32	ExtendedErrorRecEmpty
OPOS_EPTR_SLP_EMPTY	PosPrinter	System.Int32	ExtendedErrorSlpEmpty
OPOS_EPTR_SLP_FORM	PosPrinter	System.Int32	ExtendedErrorSlpForm
OPOS_EPTR_TOOBIG	PosPrinter	System.Int32	ExtendedErrorTooBig
OPOS_EPTR_BADFORMAT	PosPrinter	System.Int32	ExtendedErrorBadFormat
OPOS_EPTR_JRN_CARTRIDGE_REMOVED	PosPrinter	System.Int32	ExtendedErrorJrnCartridgeRemoved
OPOS_EPTR_JRN_CARTRIDGE_EMPTY	PosPrinter	System.Int32	ExtendedErrorJrnCartridgeEmpty
OPOS_EPTR_JRN_HEAD_CLEANING	PosPrinter	System.Int32	ExtendedErrorJrnHeadCleaning
OPOS_EPTR_REC_CARTRIDGE_REMOVED	PosPrinter	System.Int32	ExtendedErrorRecCartridgeRemoved
OPOS_EPTR_REC_CARTRIDGE_EMPTY	PosPrinter	System.Int32	ExtendedErrorRecCartridgeEmpty
OPOS_EPTR_REC_HEAD_CLEANING	PosPrinter	System.Int32	ExtendedErrorRecHeadCleaning
OPOS_EPTR_SLP_CARTRIDGE_REMOVED	PosPrinter	System.Int32	ExtendedErrorSlpCartridgeRemoved
OPOS_EPTR_SLP_CARTRIDGE_EMPTY	PosPrinter	System.Int32	ExtendedErrorSlpCartridgeEmpty
OPOS_EPTR_SLP_HEAD_CLEANING	PosPrinter	System.Int32	ExtendedErrorSlpHeadCleaning

Vendor-specific constants

OPOS constants	Type	Value
OPOS_FCL_EPTR_POWER_SUPPLY	System.Int32	10001
OPOS_FCL_EPTR_DATA	System.Int32	10002
OPOS_FCL_EPTR_CUTTER	System.Int32	10003
OPOS_FCL_EPTR_HARDWARE	System.Int32	10004
OPOS_FCL_EPTR_HEADHOT	System.Int32	10005
OPOS_FCL_EPTR_MARK	System.Int32	10006
OPOS_FCL_EPTR_PRESENTER	System.Int32	10007
OPOS_FCL_EFIRMWARE_DIFFERENT_BOOT	System.Int32	11001
OPOS_FCL_EFIRMWARE_DIFFERENT_MAIN	System.Int32	11002
OPOS_FCL_EFIRMWARE_DIFFERENT_BOOTMAIN	System.Int32	11003
OPOS_FCL_PTR_DIO_SET_QR_ENCODING	System.Int32	5
OPOS_FCL_PTR_DIO_SYSTEM_LOCALE	System.Int32	0
OPOS_FCL_PTR_DIO_UTF8	System.Int32	1
OPOS_FCL_DIE_PTR_PRESENTER_ERROR	System.Int32	2
OPOS_FCL_DIE_PTR_PRESENTER_OK	System.Int32	3

FTP2166000EQ

FTP-POS

POSPrinter OPOS Control Function specification

OPOS constants	Type	Value
OPOS_FCL_DIE_PTR_LOW_BATTERY	System.Int32	4
OPOS_FCL_DIE_PTR_BATTERY_OK	System.Int32	5

PrintBarcode constants

Barcode type	OPOS Constants	Value
MaxiCode(Mode 3)	PTR_BCS_OTHER	BarCodeSymbology.Other
MaxiCode(Mode 4, Mode 5)	PTR_BCS_OTHER+1	BarCodeSymbology.Other+1
MaxiCode(Mode 6)	PTR_BCS_OTHER+2	BarCodeSymbology.Other+2
QRCODE	PTR_BCS_OTHER+3	BarCodeSymbology.Other+3
GS1DataBar Omnidirectional	PTR_BCS_OTHER+4	BarCodeSymbology.Other+4
GS1DataBar Truncated	PTR_BCS_OTHER+5	BarCodeSymbology.Other+6
GS1DataBar Stacked	PTR_BCS_OTHER+6	BarCodeSymbology.Other+6
GS1DataBar Stacked Omnidirectional	PTR_BCS_OTHER+7	BarCodeSymbology.Other+7
GS1DataBar Limited	PTR_BCS_OTHER+8	BarCodeSymbology.Other+8
GS1DataBar Expanded	PTR_BCS_OTHER+9	BarCodeSymbology.Other+9
GS1DataBar Expanded Stacked	PTR_BCS_OTHER+10	BarCodeSymbology.Other+10

7.3.2. ILegacyControlObject Interface

The product supports ILegacyControlObject interface, so the following properties are available from POS for .NET.

BinaryConversion

ControlObjectDescription

ControlObjectVersion

7.3.3. BinaryConversion Property

When using the BinaryConversion property other than None, specify as follows.

Example: Specify “1 あ” by Nibble.	
OPOS	Specify as follows. string text = “3182:0”;
POS for .NET (C#)	Specify as follows. string text = “1” + “¥x82¥xA0”; string text = “¥x31” + “¥x82¥xA0”; string text = “¥x31” + “¥x82” + “¥xA0”; These are also OK.
POS for .NET (VB)	Specify as follows. Dim text As String text = “1” + ChrW(&H82) + ChrW(&HA0)

7.3.4. Open Method

There are no parameters.

When throwing an exception, the value of POSControlException.ErrorCode may be a possible value of the OpenResult property of OPOS.

7.3.5. Claim Method

Corresponds to the ClaimDevice method of OPOS. See ClaimDevice method.

7.3.6. CheckHealth Method

The dialog displayed when HealthCheckLevel.Interactive is specified for the parameter level of the CheckHealth method is written using OPOS constants. Please read as appropriate for enumerations of POS for .NET.

7.3.7. Close Method

Does not throw a PosControlException when executing the Close method in the closed state.

After starting the process, ArgumentNullException may be thrown when executing the Close method in a state where it has never been opened.

7.3.8. DirectIO Method

Specify the parameter *obj* as String class object.

Check the result with the return value DirectIOData structure instead of the parameter.

8. Open Source Software

This product uses the following open source software.

8.1. OPOS Common Control Objects

<http://monroeecs.com/index.htm>

Copyright 2015 Monroe Consulting Services, Inc.

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

8.2. Apache License Version 2.0

Apache License

Version 2.0, January 2004

<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent

FTP2166000EQ

FTP-POS

POSPrinter OPOS Control Function specification

licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold

FTP2166000EQ

FTP-POS

POSPrinter OPOS Control Function specification

each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License.

You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

See the License for the specific language governing permissions and limitations under the License.

9. Update History

DRW NO.		A1NC40416-6000EQ/6	
Edi t	Date	Place / Reason / Content	
01	20180928	First edition (tentative version)	
02	20181130	Overall rewriting.	
03	20190118	Support FTP-629DSL350 Support presenter.	
		Added the QR Code encoding setting function to DirectIO method.	
		About the encoding for QR Code and PDF417 printing was added.	
		Fixed incorrect description of DirectIOEvent event.	
		CharacterSet property supported PTR_CS_WINDOWS.	
4	20190329	Supported FTP-62HWSL001.	
		Supported FTP-62GDSL110, FTP-62GDSL120, FTP-62GUSL070, FTP-63GUSL070 and FTP-64GDSL130.	
		Added remarks of PowerState property.	
		Added details about OPOS_FCL_EPTR_MARK of ResultCodeExtended property.	
		Added remarks of CheckHealth method.	
		Added remarks of SetBitmap method.	
		Added low battery event to DirectIOEvent event.	
		Added about the condition when OPOS_PS_OFF_OFFLINE is reported by PowerState property.	
		Added restrictions on errors that occurred after power off or disconnected.	
5	20210820	Supported POS for .NET.	
		Added CCO restrictions.	
		Added the relationship between the QR code encoding setting by DirectIO method and BinaryConversion property.	
6	20211116	Support FTP-627DSL440, FTP-629DSL310, FTP-839DSL310, FTP-63GDSL483 and FTP-83GDSL483	
		Added the QR Code encoding getting function to DirectIO method.	
		Added the binary data sending function to DirectIO method.	
7	20240612	Change company name.	
		Support FTP-62HDSL100 and FTP-62EDSL200.	

10. Driver Update History

Version	Content of change	Date	File name
V0.01	First version (tentative version)	20180222	FTP2166000EQ_62GDSL000_OPOSV001.zip
V3.00	Stable version	20181130	FTP2166000EQ_FTP-POS_OPOSV300.zip
V3.01	Supported FTP-629DSL350 Supproted presenter.	20190118	FTP2166000EQ_FTP-POS_OPOSV301.zip
	Added the QR Code encoding setting function to DirectIO method.		
	CharacterSet property supported PTR_CS_WINDOWS.		
	Even if Binaryconversion property is OPOS_BC_NONE, 0x80 or later characters of code page 437 can be printed.		
	Fixed an issue when specifying a QR Code with a size close to the maximum width.		
	Fixed an issue in PrintNormal medhod, PrintImmediate method and SetLogo method, the encoding result changed depending on the system locale and the thread locale.		
	Fixed to encode by CharacterSet property.		
	Fixed an issue in printing QR Code, the encoding result changed depending on the thread locale.		
	Fixed to encode by system locale.		
	Fixed an issue in printing PDF417, the encoding result changed depending on the thread locale.		
	Fixed to encode by system locale.		
	Fixed an issue when specifying a character of 0x80 or later of CP437 by Setlogo method when CharacterSet property is 437.		
	Fixed an issue that the data might be replaced in pass through embedded data escape sequence.		
	FTP-62GDSL000 supported PDF 417.		
	Fixed an issue that ‘,’ can not be specified correctly when printing QR Code.		
V3.02	Supported FTP-62HWSL001.	20190329	FTP2166000EQ_FTP-POS_OPOSV302.zip
	Supported FTP-62GDSL110, FTP-62GDSL120, FTP-62GUSL070, FTP-63GUSL070 and FTP-64GDSL130.		
	Added low battery event to DirectIOEvent event.		
V3.03	Include the document in driver set.	20200318	FTP2166000EQ_FTP-POS_OPOSV303.zip
V3.04	Supported POS for .NET.	20210820	FTP2166000EQ_FTP-POS_OPOSV304.zip

FTP2166000EQ
FTP-POS
POSPrinter OPOS Control Function specification

	Fixed an issue where full cuts were not possible when using paper cut / feed and paper cut escape sequences.		
V3.05	Support FTP-627DSL440, FTP-629DSL310, FTP-839DSL310, FTP-63GDSL483 and FTP-83GDSL483	20211116	FTP2166000EQ_FTP-POS_OPOSV305.zip
	Added the QR Code encoding getting function to DirectIO method.		
	Added the binary data sending function to DirectIO method.		
V4.00	Change company name.	20240612	FTP2166000EQ_FTP-POS_OPOSV400.zip
	Support FTP-62HDSL100 and FTP-62EDSL200		