PT330/PT331

POS Printer, Cash Drawer Application Programmer's Guide of OPOS-OCX Driver for Serial/ USB/ LAN Interface

Index

1. Outline

POS Printer OPOS Control and Drawer OPOS Control that control POS Printer ("PT330-331") and Drawer connected to the printer are OPOS Controls conforming to OPOS 1.13 POS Printer Devise and Drawer Device. When using OPOS Control, refer to "OLE for Retail POS Application Programmer's Guide Instructions The 1.13 Version" (OPOS-APG V1.13), too.
In this guide, "OPOS Control" means same as "OPOS OCX".

1.1. Subject Scope of this document

These instructions (Application Programmer's Guide) aim for the main reference of programmers who develop the application for the use of OPOS Control, and describe the following contents necessary for that.

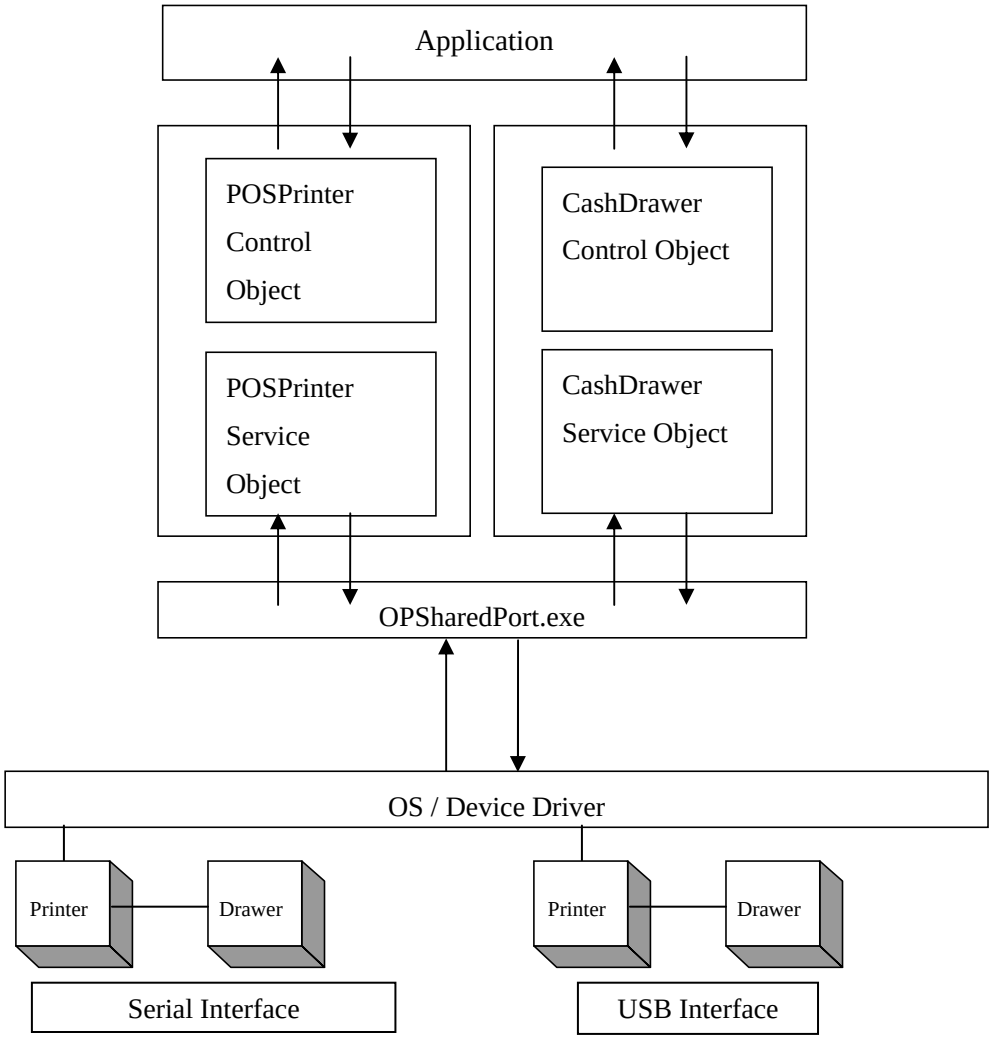
- Installation way of OPOS Control
- Usage of OPOS Control
- Restrictions of OPOS Control
- Interface (Property/Method/Event) Remarks of OPOS Control
- Item Setting Remarks of OPOS Control

1.2. OPOS Control Outline

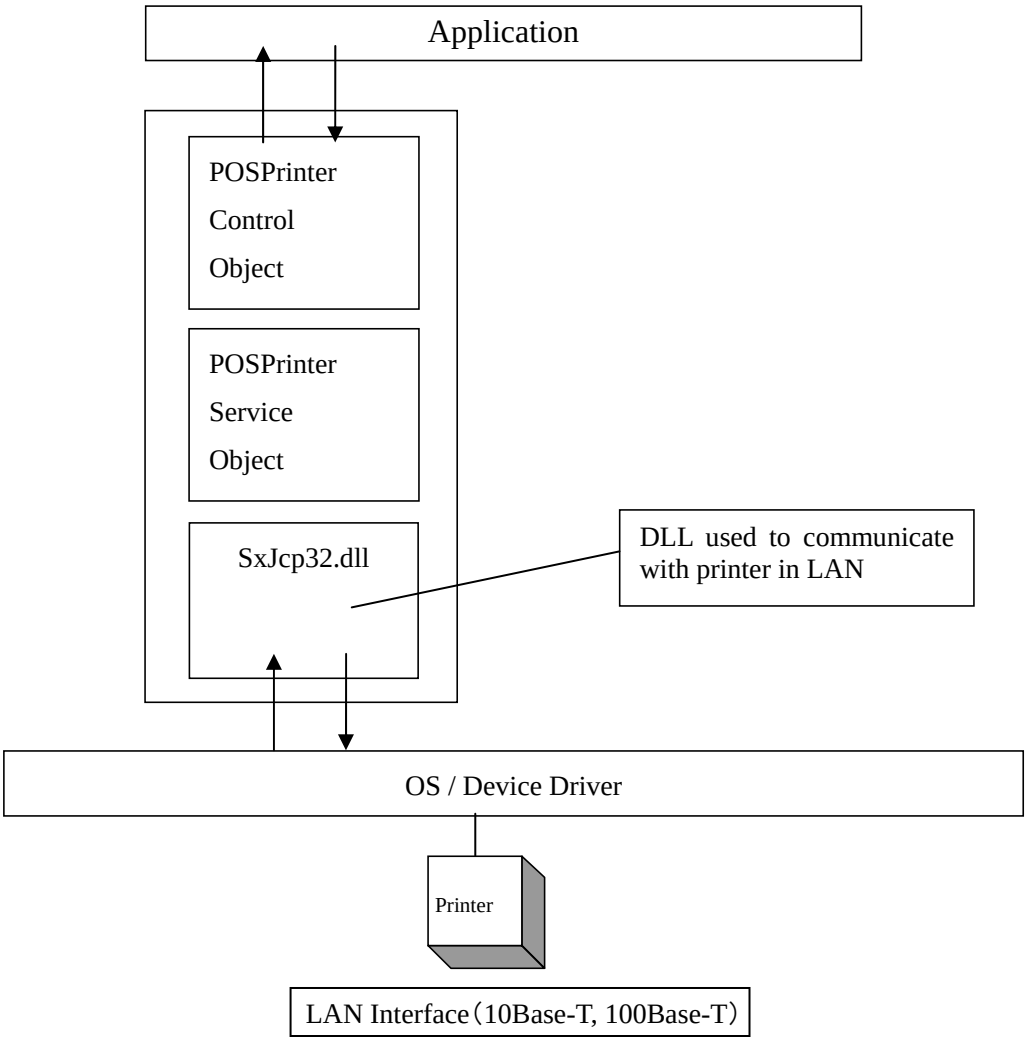
1) OPOS Control Configuration Drawing

OPOS Control conforms to ActiveX specifications and provides Property, Method, and Event to application. Control cannot be seen on UI in application execution. Only application, which uses it, requests to process through Method and Property. Application receives processing result through Method return value, parameter, Property, and Event. This OPOS Control is implemented as in-process server.

<Serial, USB Interface>



<LAN Interface>



*This OPOS Driver supports the control of Serial/USB/LAN Interface Printer.

*Multiple numbers of interface and printers can be set to the driver. For details, refer to Chapter 8 "Using Multiple Printers."

* In case of LAN Interface, OPOS Driver doesn’t support drawer connection.

*This OCX works with Thread Model of STA(Single, Thread or Apartment). To work with several processes, the setting value of "Apartment" in Registry should be set as "1".

*Regarding LAN interface, Network Connection corresponds to 10Base-T, 100Base-T.

2) Terminology

a. Control Object (Control Object; CO)

According to each device class, it provides application with the set of Property, Method, and Event. This Document explains these API.

b. Service Object (Service Object; SO)

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

1.3. Restrictions

Followings are restrictions.

1) POS Printer

[Restrictions on OPOS specifications]

All the interfaces of OPOS POS Printer Device are provided, but there are the following restrictions.

a. It does not support property setting concerning journal printing and journal.

b. It does not support property setting concerning slip printing and slip.

c. It does not support functions of Italic, custom color, shading printing, and cartridge.

d. It does not support change of receipt printing character font. (Printing font change)

e. The following methods always return OPOS_E_ILLEGAL(106) after enabling.

PrintTwoNormal Method
BeginInsertion Method
EndInsertion Method
BeginRemoval Method
EndRemoval Method
ChangePrintSide Method
MarkFeed Method
ResetStatistics Method
RetrieveStatistics Method
UpdateStatistics Method

(Limitation of Cable disconnection and connection)

There are conditions for disconnection and connection of LAN cable under printer "enable" status:

1)When LAN cable is disconnected from PC, printer cannot support any actions. When the cable is disconnected during "enable" status, OCX may fail to be "enable" even though the application restarts.

2)When LAN cable is disconnected from Printer, and while OCX and printer is connecting, printer cannot support any actions even if "disable" is sent. If "disable" is sent in above situation, even though the application is restarted, OCX may fail to enable.

When the LAN cable is disconnected, connect the LAN cable again, and switch off and on the power of Printer.

2) Drawer

[Restrictions of OPOS specifications OPOS]

All the interfaces of OPOS Drawer Device are provided, but there are the following restrictions.

a. **PowerNotify** Property (Power source notifying function setting)

Setting is only for OPOS_PN_DISABLED(0) (Impossible to notify)and unchangeable.

b. **PowerState** Property (Power source state)

Only OPOS_PS_UNKNOWN(2000)(Unclear) is set.

c. **DirectIO** Method (Particular-to-Device function)

It is not supported. After enabling, it always returns OPOS_E_ILLEGAL(106).

d. **WaitForDrawerClose** Method (Waiting for the drawer to close)

It is not supported. After enabling, it always returns OPOS_E_ILLEGAL(106).

e. **DirectIOEvent** Event (Particular- to-Device event)

It is not supported.

f. **DrawerOpened** Property, **StatusUpdateEvent** Event

Status notification of the Drawer is available only when **CapStatus** is TRUE and Printer OCX is enabled (**DeviceEnabled**=TRUE) for the printer connected to the drawer. In case these conditions are not met, the status of the drawer is not notified.

[Restriction of Drawer Hardware Specifications]

It does not support notifying function of drawer power source condition.

3) Common Restrictions on POS Printer and Drawer

The OPOS Control is not thread-safe. When the method or property is accessed from the different thread, unexpected result may occur. In the multithread environment, implementation of exclusive processing for the critical sections is required for accessing the property and executing the method in order to avoid the method and property are executed at the same time.

4) Restriction when Windows driver and OPOS driver are installed in the same system

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

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

5) Setting of Apartment ="0" with connecting LAN interface cable

Apartment is set as "0" in Registry setting, one process can control several printers. But the several processes do "Open" OCX (LAN Interface) at the same time, unexpected results may happen.

6) Setting of Apartment ="1" with connecting LAN interface cable

Apartment is set as "0" in Registry setting, several processes can control printers.

In this case, the number of printers (LAN interface) should be guaranteed that one process should be less than one unit. If more than 2 printers per one process were used, unexpected results might occur. The following chart shows the examples of the case that Apartment ="1" can work, and the case that Apartment ="1" cannot work. (OCX uses LAN interface.)

examples of the case that Apartment ="1" can work

ProcessA---PrinterOCX

ProcessB---PrinterOCX

examples of the case that Apartment ="1" cannot work---1

ProcessA---**PrinterOCX, PrinterOCX**

ProcessB---PrinterOCX

(Two PrinterOCX exist in ProcessA.)

7) About the "Found New Hardware Wizard" in Windows

When the printer is connected to PC in USB or Parallel interface and "Found New Hardware Wizard" was displayed, set it in the following procedures.

For Windows 7 / Server 2008 R2

1. The message "Installing device driver software" is displayed in a balloon(lower right corner of the screen).
2. After a while, the message "Device driver software was not successfully installed" is displayed. This is not a problem.

For Windows Vista / Server 2008

1. The "Found New Hardware" is displayed, select [Don't show this message again for this device].

   * When "User Account Control" screen is displayed, select [Continue].

For Windows XP / Server 2003 / WEPOS / WEPOS2009

1. The "Found New Hardware Wizard" screen is displayed. Select [No, not this time] and then click [Next].
2. When "What do you want the wizard to do?" is displayed, select [Install from a list or specific location] and then click [Next].
3. Select [Search for the best driver in these locations], clear all checkboxes and then click [Next].
4. When "Cannot Install this Hardware" is displayed, click [Don't prompt me again to install this software] and then click [Finish].

For Windows 2000

1. The "Found New Hardware Wizard" screen is displayed, click [Next].
2. When "What do you want the wizard to do?" is displayed, select [Search for a suitable driver for my device] and then click [Next].
3. When [Locate Driver Files] is displayed, clear all checkboxes and then click [Next].
4. When "Driver Files Search Results" is displayed, select [Disable the device] and then click [Finish].

1.4. Connection Way to POS Printer

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

Memory Swith 1

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

Memory Swith 2

No.	Setting Item	Setting Contents
1	Cover Open Error	Auto Recovery
2	Error	Recovery by CMND
3	Batch (COM IF)	Disable
4	Batch (Other IF)	Disable
5	Serial Number	Enable
6	ASB	Enable
7	Font-B	Mode1

Print

No.	Setting Item	Setting Contents
1	Paper Width	*Set from the setting program
2	Max Speed	*Set from the setting program
3	Print Density	*Set from the setting program

Hardware

No.	Setting Item	Setting Contents
1	Error Alert	*Set from the setting program
2	Buzzer Interval	*Set from the setting program
3	Buzzer Repetition	*Set from the setting program
4	User NV Memory	192KB
5	Graphic Memory	384KB
6	Cut at CoverClose	*Set from the setting program
7	PNE Detect	*Set from the setting program

Interface

No.	Setting Item	Setting Contents
1	USB	Printer
2	Protocol	XON/XOFF

2. Installation

OPOS Control can be installed just like in the following procedures.

2.1. Installation Condition

Operation Environment

- OS: Microsoft Windows 2000, XP, 2003Server, Vista, WEPOS, POSReady2009 Windows7, Server 2008, Server 2008 R2
- CPU: Pentium3 550MHz and more is recommended
- RAM: 128MB and more is recommended
- HDD: Space 2MB and more

*To install on WEPOS, "Local Management Support" must be installed in the minimum configuration of WEPOS.

* To install on POS Ready 2009 using Batch Installer, "Command-line Application" component must be installed on POS Ready 2009.

(Note)
When OPOS control (POS PRINTER OPOS OCX) is already installed,
please install this OPOS control after uninstallation of OPOS control.
Please refer how to uninstall to **2.4. uninstallation procedures**.

2.2. Installation Media

Installation media is provided as CD.

Configuration of CD is just the following.

\(root) \Drivers\OPOS

\ Driver ..... The root folder of the Installer (x32)

- setup32.exe : Installer for 32bit OS
- setup64.exe : Installer for 64bit OS

\Driver_x32 ..... The folder of the driver (x32)

* Use this when you do not use the installer for 32bit.

\Driver_x64 ..... The folder of the driver (x64)

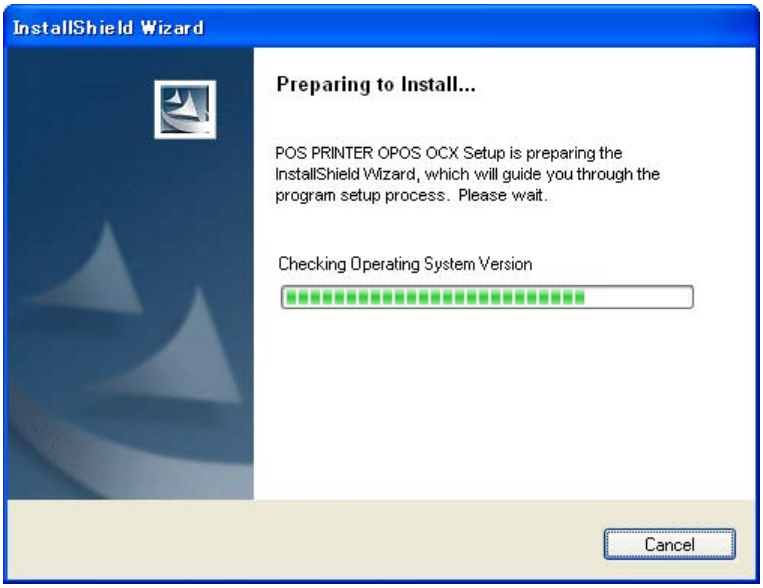
* Use this when you do not use the installer for 64bit

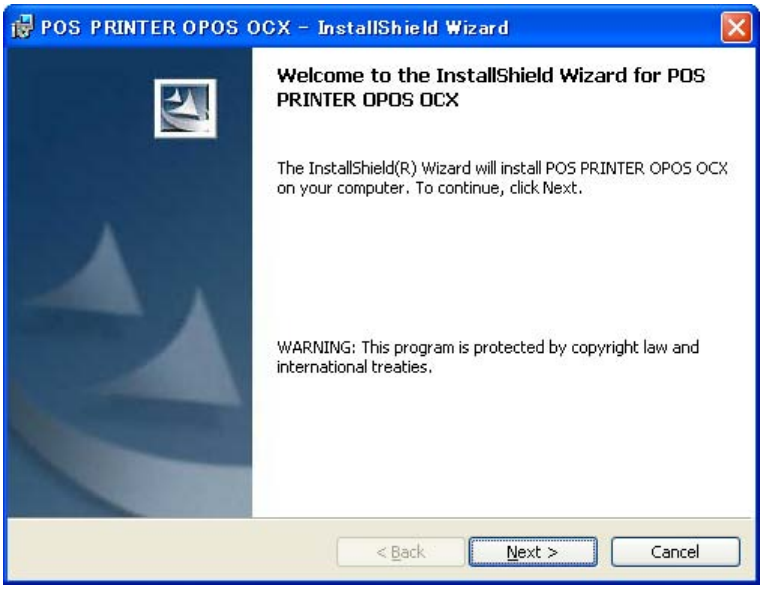
2.3. Installation Procedure

Installation using installer

1. Set the installation Media CD to the Drive. After that, start "My Computer" or "Explorer" and refer to the Drive.
   - In case of CD: Open \Driver folder, and if OS is 32bit, double click Setup32.exe. If OS is 64bit, double click Setup64.exe

2. After the dialogue below is shown, Setup Program starts.

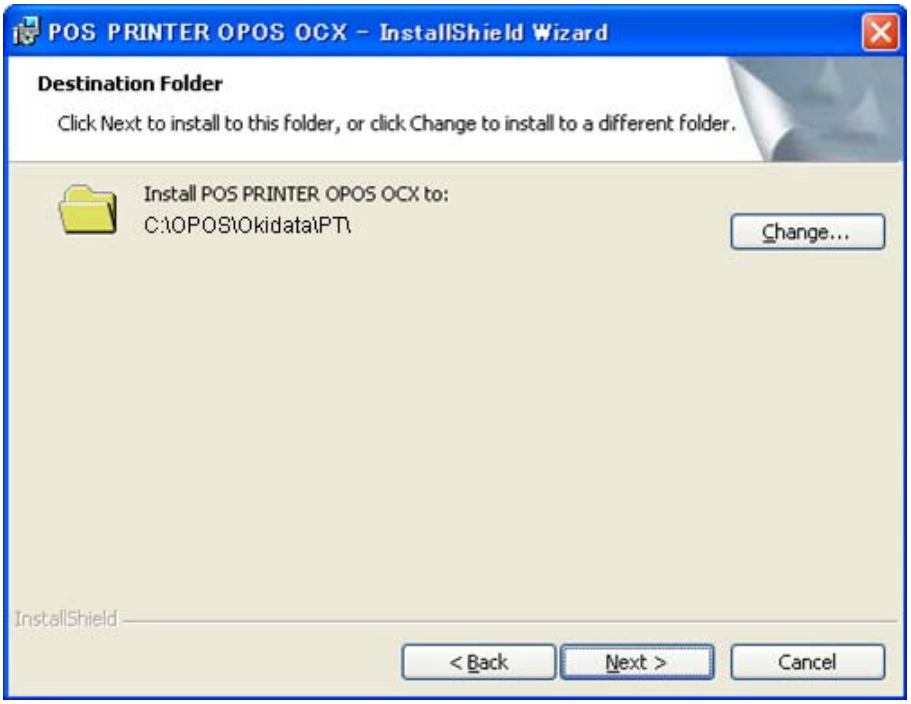




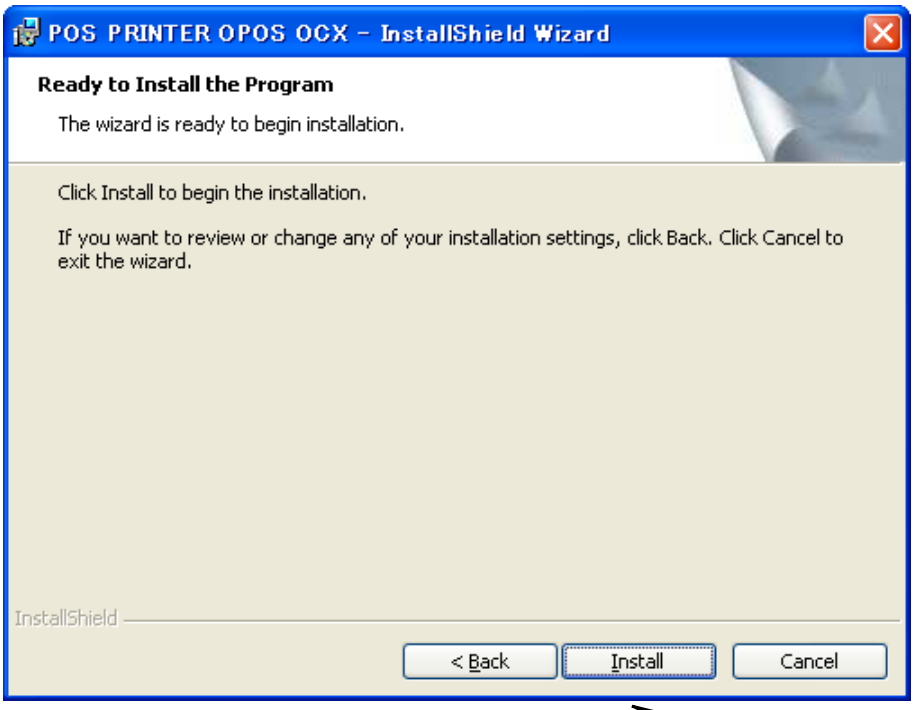
click "Next"

3. Installation starts. Installation folder is:

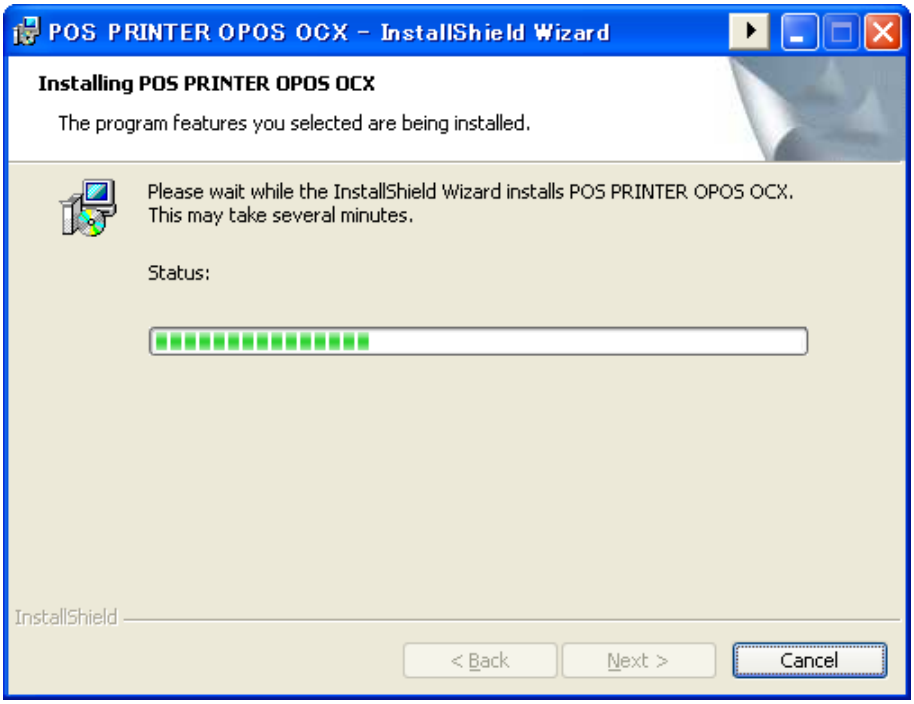
[System Drive]:\OPOS\Okidata\PT\

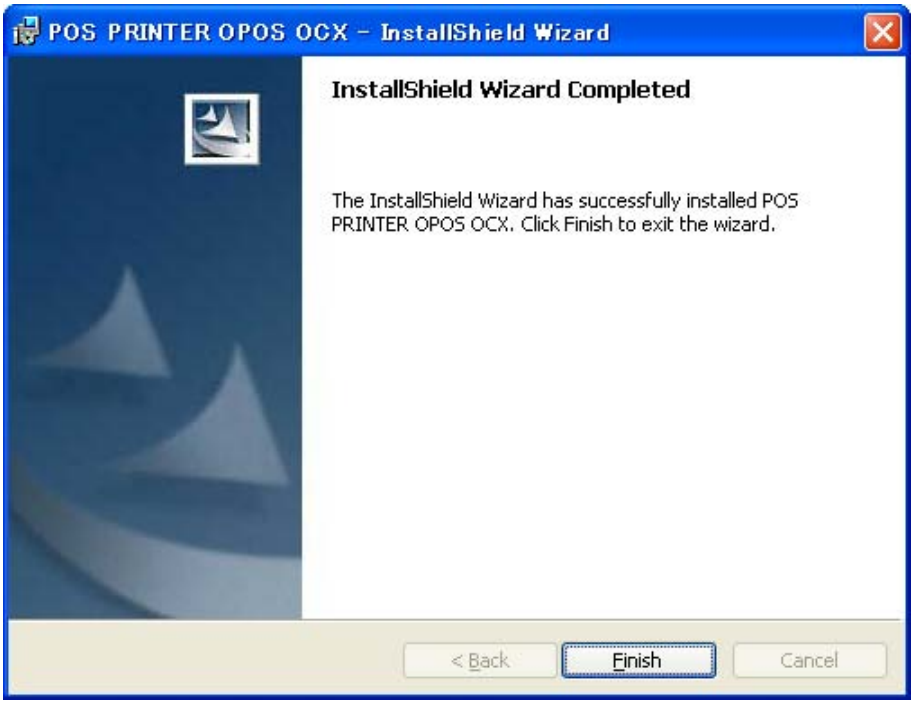


click "Next"



click "Install"





click "Finish"

4. Start setting program.

If the [User Account Control] dialog box is displayed,

[Windows 7 / Server 2008 R2]

click [Yes].

[Windows Vista / Server 2008]

click [Allow].

【For Windows 7 / Windows Vista】

1) Click the button indicated with the Windows logo at the left bottom of the desktop.

2) Go to [All programs] →

[Okidata]→[OPOS]→[Printer OPOS Setup].

【For Windows XP / Windows Server 2008 R2 / Windows Server 2008 / Windows Server 2003 / WEPOS/POS Ready 2009】

Go to [Start]→[All programs] →

[Okidata]→[OPOS]→[Printer OPOS Setup].

【For Windows 2000】

Go to [Start]→[Program] →

[Okidata]→[OPOS]→[Printer OPOS Setup].

*As for how to use setting program, refer to **2.6. Setting Program Usage.**

5. After setting, installation finishes.

Installation using batch file

* If OS is 32bit, use batch file of \Driver_x32 folder. If OS is 64bit, use batch file of \Driver_x64 folder

■ Installation method for Windows Vista or Windows 7

1. Copy driver folder ("\Driver_x64" or "\Driver_x32") onto an arbitrary folder.
2. The button that attaches the Windows logo under the left of a desktop screen is clicked.
3. Select [All Programs]->[Accessories]->[Command Prompt], and right-click in the icon of the [Command Prompt], click "Run as administrator (A)", and execute the command prompt by the administrator authority.
4. Change the current directory of the command prompt to the directory that copied by 1, and execute the batch file "Install.bat".
5. The Registry Editor and additional confirmation message to the registry is displayed, and click [yes]. The additional completion message to the registry is displayed when the addition to the registry is completed, and click [OK].
6. The driver is installed in the folder "C:\OPOS\Okidata\PT\".
7. Please execute "C:\OPOS\Okidata\PT\OPPrinterSetup.exe", and execute the setup.
   *As for how to use setting program, refer to **2.6. Setting Program Usage.**
8. After setting, installation finishes.

■ Installation methods except Windows Vista and Windows 7

1. Driver folder ("\Driver_x64" or "\Driver_x32") onto an arbitrary folder.
2. The button that attaches the Windows logo under the left of a desktop screen is clicked.
3. Select [All Programs]->[Accessories]->[Command Prompt], and execute the command prompt.
4. Change the current directory of the command prompt to the directory that copied by 1, and execute the batch file "Install.bat". (Use "Install_Win2k.bat" for Windows 2000.)
5. The Registry Editor and additional confirmation message to the registry is displayed, and click [yes]. The additional completion message to the registry is displayed when the addition to the registry is completed, and click [OK].
6. The driver is installed in the folder "C:\OPOS\Okidata\PT\".
7. Please execute "C:\OPOS\Okidata\PT\OPPrinterSetup.exe", and execute the setup.
   *As for how to use setting program, refer to **2.6. Setting Program Usage.**
8. After setting, installation finishes.

2.4. Uninstallation Procedure

Uninstallation when installing it with installer (Setup32.exe or Setup64.exe)

1. Follow the steps blow to display the uninstalling screen.

【For Windows 7 / Windows Vista】

・Click the button indicated with the Windows logo at the left bottom of the desktop.

・Go to [Control Panel]→[Uninstall program] or [Uninstall a program].

【For Windows Server 2008 R2】

・Go to [Start]→[Control Panel]→[Uninstall a program]

【For Windows Server 2008】

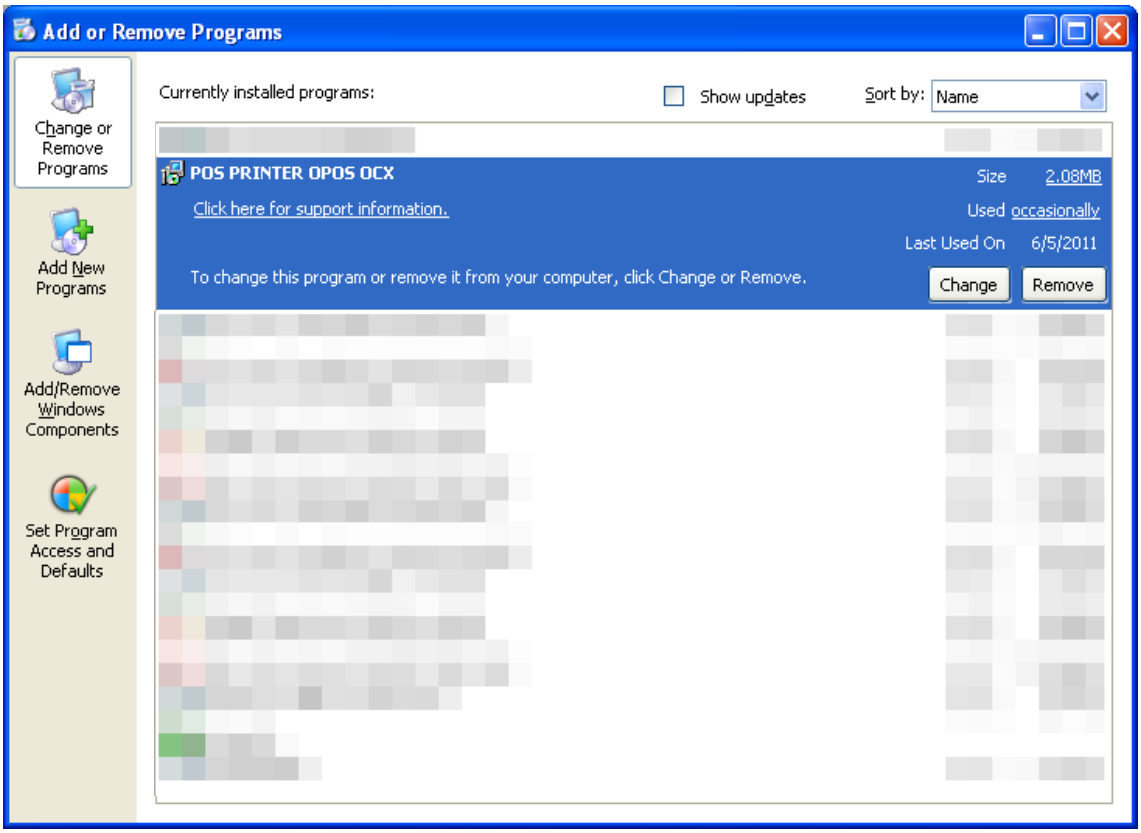
・Go to [Start]→[Control Panel]→[Programs and Functions].

【For Windows XP / Server 2003 / WEPOS】

・Go to [Start]→[Control Panel]→[Add or Remove Programs].

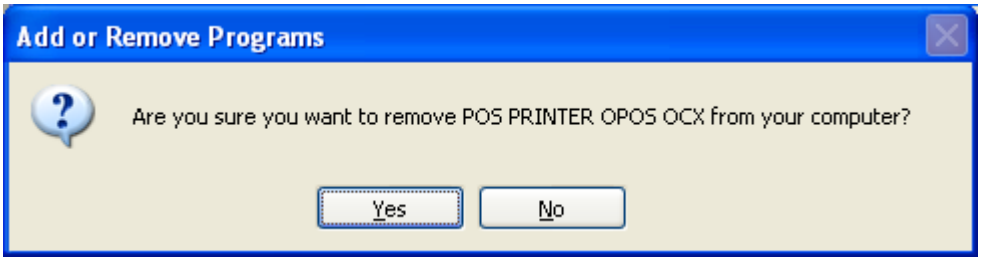
【For Windows 2000】

・Go to [Start]→[Settings]→[Control Panel]→[Add or Remove Programs].

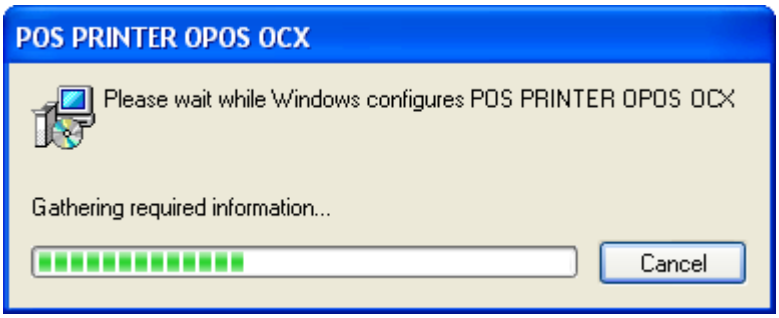


2. Select "POSPrinter OPOS OCX" and click "Uninstall" or "Remove".

3. The dialogue to confirm Uninstallation is displayed. Click [Yes].



4. Uninstallation is executed.



5. Sometimes there are some files that could not be deleted by uninstaller in Installed folder ([System Drive]:\OPOS\Okidata\PT), so please delete manually.

That is all of uninstallation.

Uninstallation when OPOS is installed with batch file

- Uninstallation method for Winodws Vista or Windows 7

1. The button that attaches the Windows logo under the left of a desktop screen is clicked.
2. Select [All Programs]->[Accessories]->[Command Prompt], and right-click in the icon of the [Command Prompt], click "Run as administrator (A)", and execute the command prompt by the administrator authority.
3. Execute batch file "uninstall.bat" with the folder of "C:\OPOS\Okidata\PT\".
4. Delete "C:\OPOS\Okidata\PT" folder manually.
5. It is an end.

- Uninstallation methods except Windows Vista and Windows 7

1. The button that attaches the Windows logo under the left of a desktop screen is clicked.
2. Execute batch file "uninstall.bat" with the folder of "C:\OPOS\Okidata\PT\".
3. Delete "C:\OPOS\Okidata\PT" folder manually.
4. It is an end.

2.5. Installation File List

File groups of POS Printer OPOS-OCX Control are arranged just like the following.

[System Drive]:\OPOS\Okidata\PT\

OPPOSPrinterCO.ocx	POS Printer Control Object
OPPOSPrinterSO.dll	POS Printer Service Object
OPCashDrawerCO.ocx	Drawer Control Object
OPCashDrawerSO.dll	Drawer Service Object
OPPrinterSetup.exe	Setting Program
oposdef.ini	OPOS display string definition file

[System Drive]:\OPOS\Okidata\PT\Log

Folder to output Log File

[System Drive]:\OPOS\Okidata\PT\include

Include-File-Group defined by OPOS-OCX Control.

Opos.h	OPOS Shared-Definition-Constant Header File
OposPtr.h	OPOS POS Printer Definition-Constant Header File
OposCash.h	OPOS Drawer Definition-Constant Header File
FitPtr.h	Particular Definition-Constant Header File for OPOS Printer.
OPOSPRN.BAS	OPOS Definition-Constant Standard Module (Shared-Definetion-Constant, Printer and Drawer Definition-Constant) (for Visual Basic)
OposFIT.BAS	OPOS Printer Particular Definition Constant Module (for Visual Basic)

[System Drive]:\WINDOWS\system32\
(64bit OS: [System Drive]:\Windows\SysWOW64\)

OPSharedPort.exe	Shared Port Program
SxJcp32.dll	DLL used to communicate with printer in LAN

The module to be installed has set property according to the following standard.

Item	Remarks
File Version	OPOS Version (for each release, the third and the forth characters are incremental numbers of version) In case of files except CO/SO, successive increment from 1.0.0.1
Remarks	- In case of CO; "POSPrinter [or Cash Drawer] OPOS Control Object Module". - In case of SO; "POSPrinter [or Cash Drawer] OPOS Service Object Module". - The other cases; "[exe file name part without extension] XXX (any wording)".
Copyright	"Copyright (C) 2011 Oki Data Corporation"
Comment	(Not written specially; If necessary, written specially)
Special Built Information	(Not written specially)
Private Built Information	(Not written specially)
Company	" Oki Data Corporation"
Language	"English (U.S.)"
Trademark	(Not written specially)
Formal File Name	(Object File Name of the subject)
Product Version	Same as File Version
Product Name	"OPOS POSPrinter" or "OPOS Cash Drawer"
Inside name	(Formal File Name without extension)

2.6. Setting Program Usage

Operation Conditions

OPOS must be installed

Screen and function

Setting Program is executed as the following procedure.

1. Execution

If the [User Account Control] dialog box is displayed,

[Windows 7 / Server 2008 R2]

click [Yes].

[Windows Vista / Server 2008]

click [Allow].

【For Windows 7 / Windows Vista】

1) Click the button indicated with the Windows logo  at the left bottom of the desktop.

2) Go to [All programs] →

[Okidata]→[OPOS]→[Printer OPOS Setup].

【For Windows XP / Windows Server 2008 R2 / Windows Server 2008 / Windows Server 2003 / WEPOS/ POS Ready 2009】

Go to [Start]→[All programs] →

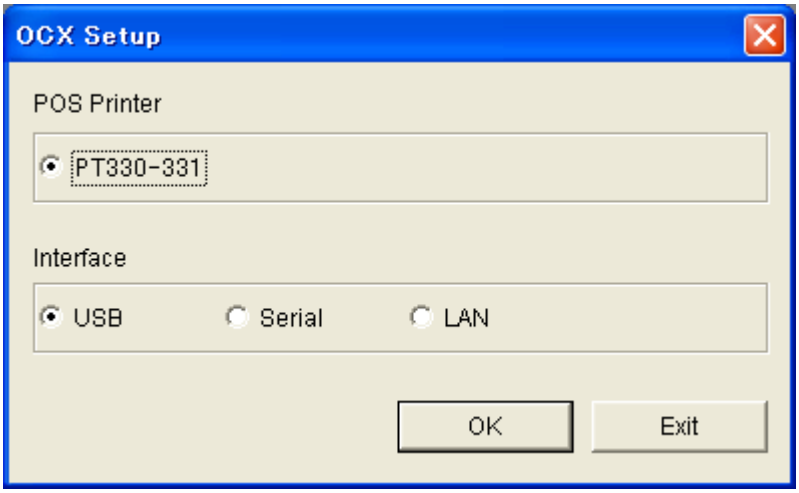
[Okidata]→[OPOS]→[Printer OPOS Setup].

【For Windows 2000】

Go to [Start]→[Program] →

[Okidata]→[OPOS]→[Printer OPOS Setup].

2. Selecting Printer and Interface

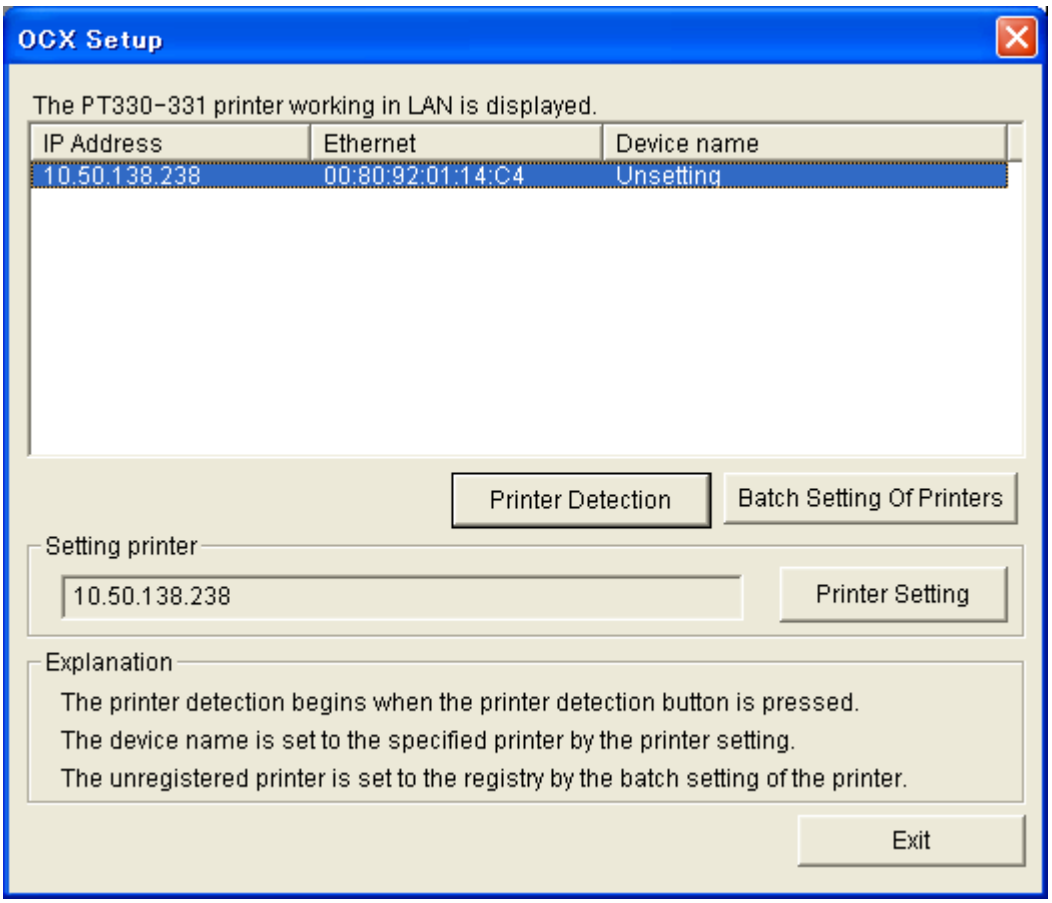


Select “POS Printer” and “Interface”.

When LAN interface is selected, Searching printers screen is displayed.

When other interface is selected, POS printer detail setting screen is displayed.

3. Searching printers



- Printer Detection

When "Printer Detection" is pressed, the list of IP address, MAC address and Device name of the found POS printer is displayed. When Device name is "Unsetting", Printer is disabled on OCX control. There are 2 solutions to enable Printer on OCX control. One is to set all Unsetting POS printers to default setting by "Batch Setting Of Printers", and another one is to entry Unsetting Printers one by one by "Printer Setting".

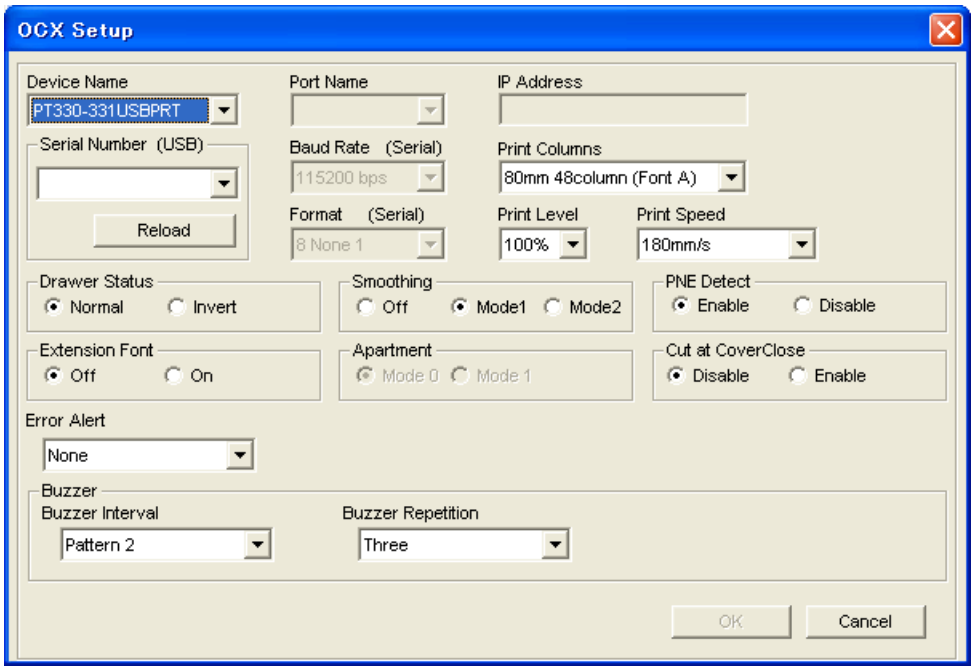
- Batch Setting Of Printers

When "Batch Setting Of Printers" button is pressed, OCX gives "Device name" to every single "Unsetting" printer. In that case, registry setting value will be the same as "5. Registry Used by OCX ". IP address and Device name is set depending on each printer's setting.

- Printer Setting

When POS printer is selected in the list, "Setting printer" shows own IP address, then press "Printer Setting" button for moving to POS printer detail setting screen page. Double click on a printer in the list for jumping to POS printer detail setting screen page directly.

4. POS printer detail setting



When an error occurred with OK button pushed, the following messages are displayed. Confirm an error factor, and setup again.

<An error factor>

A cable is not connected.

The printer is not switched on.

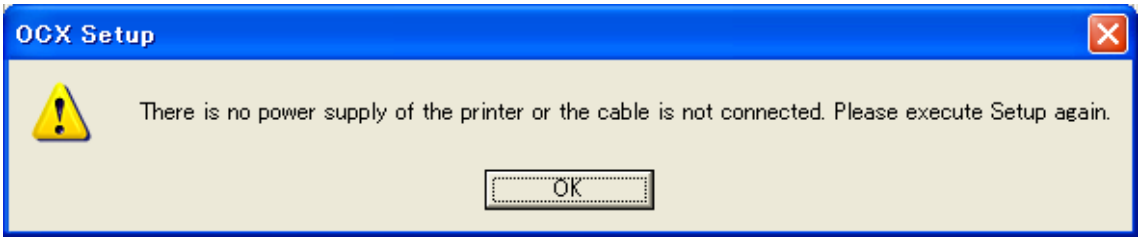
A cover opens.

There is not paper.

A port is already used in others.

A communication condition does not accord with a printer. (Serial connection)

Serial number input is wrong. (USB connection)



1) Device Name

Specify the device to be set which may differ depending on the OCX installed. The items to be displayed is as follows:

Device Name	Interface
PT330-331SERPRT	For Serial Interface
PT330-331USBPRT	For USB Interface
PT330-331SER2PRT	For Second Serial Interface
PT330-331USB2PRT	For Second USB Interface
PT330-331LANPRT_xxx.xxx.xxx.xxx	For LAN Interface (xxx.xxx.xxx.xxx : IP Adress.)

2) Port Name

For XXXSERPRT or XXXSER2PRT, the port can be selected from "COM1" to "COM9". The port number other than above can be entered into the Dropdown list. "COM1" is set as the default setting after installation.

3) IP Adress

IP address of the set device is displayed.

4) Serial Number

For XXXUSBPRT or XXXUSB2PRT, select the serial number of the printer from "Serial Number" by all means.

When the serial number of the printer is set by "123456", please select "123456" from "Serial Number".

5) Baud Rate

Perform the baud rate setting. This setting is available only when XXXSERPRT or XXXSER2PRT is selected in the “Port Name”. Set the baud rate set in the Printer.

6) Format

Perform the communication format. This setting is available only when XXXSERPRT or XXXSER2PRT is selected in the “Port Name”. Set the communication format set in the Printer.

The communication protocol is fixed to XON/XOFF.

* Special attention is required to perform the following settings:

1. Specifying 7bit Even (7bit Even 1) is not available from 8bit None (8 None 1) setting. To perform this setting, specify Even (8 Even 1) first, and then specify 7bit Even.
2. Specifying 7bit Odd (7bit Odd 1) is not available from 8bit None (8 None 1) setting. To perform this setting, specify Odd (8 Odd 1) first, and then specify 7bit Even.
3. To specify 8bit None (8 None 1) from 7bit Even (7 Even 1), specify 8bit Even (8 Even 1) first, and then specify 7bit Even.
4. To specify 8bit None (8 None 1) from 7bit Odd (7 Odd 1), specify 8bit Even (8 Odd 1) first, and then specify 7bit Odd.

7) Print Columns

Set the width of receipt and the number of the characters to be set in a single line.

8) Drawer Status

With some drawer, the drawer open/close event from the Drawer OCX may be reversed. To acquire the event properly and avoid such behavior, change the setting to "Invert."

9) Smoothing

Select whether smoothing processing is performed when the RecLetterQuality property is set to TRUE while printing double byte or more is specified for the font printing. When Mode1 or Mode2 is selected, smoothing processing is performed. With smoothing processing, the quality of double byte printing is improved, but printing performance may be declined slightly.

10) PNE Detect

Set whether to notify the paper near end.

11) Print Level

Specifies the printing thickness.

The default's printing level of black is taken as 100 %. The percentage can be set from 70 to 130 by 10 %.

12) Print Speed

Specifies the printing speed. It can be set to the numbers form 100mm/s to 220mm/s by 10mm/s. The default is set to 220mm/s.

13) Extension Font

Set whether to use Extension font.

This is valid only for the Extension font model.

"Off" uses normal size font.

"On" uses Extension font, which is larger than normal size font.

14) Apartment

The method of initialization and the termination of DLL is specified.

15) Cut at CoverClose

When "Enable" is selected, cut operation is executed when cover is closed.

When "Disable" is selected, cutting operation is not executed when cover is closed.

16) Error Alert

It is a warning method with the buzzer when the error occurs.

When "None" is selected, the buzzer doesn't beep when the printer error occurs.

When "One Time" is selected, the buzzer beeps only once when the printer error occurs.

When "Continuous" is selected, the buzzer keeps beeping from the occurrence of the error to release.

17) Buzzer Interval

Specifies the beeping interval of the buzzer.

18) Buzzer Repetition

Specifies the repetition frequency of the buzzer.

19) Cancel, OK buttons

When the OK button is pressed, the settings are written into the registry and the command to change the settings is sent to the Printer. To run this program, connect the printer and turn on the power beforehand. (If the printer is not connected, the power is not turned on or the printer is in the error status, the settings are not set in the Printer and only the registry is set.)

20) Reload buttons

When the Reload button is pressed, the serial number of the printer connected with USB is acquired again.

Operation Conditions

Registry Setting

The value to write into the registry may differ depending on the printer to be set.

Change the value as follows according to the printer in use. When the setting is applicable to the particular OCX, it is stated as so.

	Device Name	
	Serial Interface	**USB Interface**
POS Printer (first)	PT330-331SERPRT	PT330-331USBPRT
Drawer 1 (first)	PT330-331SERDR1	PT330-331USBDR1
Drawer 2 (first)	PT330-331SERDR2	PT330-331USBDR2
POS Printer (second)	PT330-331SER2PRT	PT330-331USB2PRT
Drawer 1 (second)	PT330-331SER2DR1	PT330-331USB2DR1
Drawer 2 (second)	PT330-331SER2DR2	PT330-331USB2DR2

	Device Name
	LAN Interface
POS Printer (IP Adress : xxx.xxx.xxx.xxx)	PT330-331LANPRT_xxx.xxx.xxx.xxx

Port Name (Serial)

The value specified is directly written into the following registry:

- HKEY_LOCAL_MACHINE\SOFTWARE\OLEforRetail\ServiceOPOS\POSPrinter\[*Printer*]\port
- HKEY_LOCAL_MACHINE\SOFTWARE\OLEforRetail\ServiceOPOS\CashDrawer\[*Drawer1*]\port
- HKEY_LOCAL_MACHINE\SOFTWARE\OLEforRetail\ServiceOPOS\CashDrawer\[*Drawer2*]\port

SerialNumber (USB)

Default Port of the USB is set by "USB", and the set value adds it after "USB" and writes in it at the following registry. (Please set the serial number of the printer to connect)

(An example: When the serial number is set to "123456", "USB123456" is written in the registry.)

- HKEY_LOCAL_MACHINE\SOFTWARE\OLEforRetail\ServiceOPOS\POSPrinter\[*Printer*]\port
- HKEY_LOCAL_MACHINE\SOFTWARE\OLEforRetail\ServiceOPOS\CashDrawer\[*Drawer1*]\port
- HKEY_LOCAL_MACHINE\SOFTWARE\OLEforRetail\ServiceOPOS\CashDrawer\[*Drawer2*]\port

Baud Rate, Format (For Serial only)

Based on the setting values, the combination of Baud Rate and Format is written to the registry. For example, when Baud Rate = "19200" and Format = "8 None 1", "19200,N,8,1,x" is written into the value. When Format = "7 Odd 1", "19200.O,7,1,x", and when Format = "8 Even 1", "19200,E,8,1,x" are written into the following values:

- HKEY_LOCAL_MACHINE\SOFTWARE\OLEforRetail\ServiceOPOS\POSPrinter\[*Printer*]\Protocol
- HKEY_LOCAL_MACHINE\SOFTWARE\OLEforRetail\ServiceOPOS\CashDrawer\[*Drawer1*]\Protocol
- HKEY_LOCAL_MACHINE\SOFTWARE\OLEforRetail\ServiceOPOS\CashDrawer\[*Drawer2*]\Protocol

IPAddress(LAN)

IP Address of the printer connected with the network is set.

- HKEY_LOCAL_MACHINE\SOFTWARE\OLEforRetail\ServiceOPOS\POSPrinter\[*Printer*]\IPAddress

Print Columns

Based on the setting values set, the number of the characters per line, available line character width and paper width are set in the following values:

- HKEY_LOCAL_MACHINE\SOFTWARE\OLEforRetail\ServiceOPOS\POSPrinter\[*Printer*]\RecLineChars
- HKEY_LOCAL_MACHINE\SOFTWARE\OLEforRetail\ServiceOPOS\POSPrinter\[*Drawer1*]\RecLineCharsList
- HKEY_LOCAL_MACHINE\SOFTWARE\OLEforRetail\ServiceOPOS\POSPrinter\[*Drawer2*]\RecLineWidth

The combinations are listed in the following table:

Print Columns Setting Value	48	42	35	32
RecLineChars	48	42	35	32
RecLineCharsList	48,57,72	42,51,64	35,42,52	32,38,48
RecLineWidth	576	512	420	384

Drawer Status

When the value is set to Normal, "0" is written into the following registry, and when the value is set to Invert, "1" is written.

- HKEY_LOCAL_MACHINE\SOFTWARE\OLEforRetail\ServiceOPOS\POSPrinter\[*Printer*]\InvertDrawerStatus

Smoothing

When the value is set to Mode1, "1" is written into the following registry , and when the value is set to Mode2, "2" is written into the following registry, and when the value is set to OFF, "0" is written.

- HKEY_LOCAL_MACHINE\SOFTWARE\OLEforRetail\ServiceOPOS\POSPrinter\[*Printer*]\Smoothing

PNE Detect

When the value is set to Enable, "E" is written into the following registry, and when the value is set to Disable, "D" is written.

- HKEY_LOCAL_MACHINE\SOFTWARE\OLEforRetail\ServiceOPOS\POSPrinter\[*Printer*]\PNESense

Print Level

The value specified is directly written into the following registry:

- HKEY_LOCAL_MACHINE\SOFTWARE\OLEforRetail\ServiceOPOS\POSPrinter\[*Printer*]\PrintLevel

Print Speed

The value specified is directly written into the following registry:

- HKEY_LOCAL_MACHINE\SOFTWARE\OLEforRetail\ServiceOPOS\POSPrinter\[*Printer*]\PrintSpeed

Batch Print

When the value is set to Enable, "E" is written into the following registry, and when the value is set to Disable, "D" is written.

- HKEY_LOCAL_MACHINE\SOFTWARE\OLEforRetail\ServiceOPOS\POSPrinter\[*Printer*]\BatchPrint

Extension Font

When the value is set to Off, "0" is written into the following registry, and when the value is set to On, "1" is written.

- HKEY_LOCAL_MACHINE\SOFTWARE\OLEforRetail\ServiceOPOS\POSPrinter\[*Printer*]\ExtensionFont

AnkCharsSpacing and RecLineSpacing

If "Extension Font" value is set to On, "2" is written in "Ank.CharsSpacing" and "34" is written in "RecLineSpacing".
If "Extension Font" value is set to Off, "0" is written in "Ank.CharsSpacing" and "30" is written in "RecLineSpacing".

- HKEY_LOCAL_MACHINE\SOFTWARE\OLEforRetail\ServiceOPOS\POSPrinter\[*Printer*]\AnkCharsSpacing
- HKEY_LOCAL_MACHINE\SOFTWARE\OLEforRetail\ServiceOPOS\POSPrinter\[*Printer*]\RecLineSpacing

Apartment(LAN)

When the value is set to Mode0, "0" is written into the following registry, and when the value is set to Mode1, "1" is written.

- HKEY_LOCAL_MACHINE\SOFTWARE\OLEforRetail\ServiceOPOS\POSPrinter\[*Printer*]\Apartment

Cut at CoverClose

When the value is set to Enable, "E" is written into the following registry, and when the value is set to Disable, "D" is written.

- HKEY_LOCAL_MACHINE\SOFTWARE\OLEforRetail\ServiceOPOS\POSPrinter\[*Printer*]\CutAtCoverClose

Error Alert

When the value is set to "None", "N" is written into the following registry, and when the value is set to "OneTime", "O" is written, and when the value is set to "Continuous", "C" is written.

- HKEY_LOCAL_MACHINE\SOFTWARE\OLEforRetail\ServiceOPOS\POSPrinter\[*Printer*]\Error Alert

Buzzer Interval

The setting pattern number is written into the following registry.

- HKEY_LOCAL_MACHINE\SOFTWARE\OLEforRetail\ServiceOPOS\POSPrinter\[*Printer*]\BuzzerInterval

Buzzer Repetition

The setting repetition frequency is written into the following registry.

- HKEY_LOCAL_MACHINE\SOFTWARE\OLEforRetail\ServiceOPOS\POSPrinter\[*Printer*]\BuzzerRepetition

3. Using OPOS Control

3.1. Common

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

1. **Open** method: Called to link the control object to the service object.
2. **ClaimDevice** method: Called to enable exclusive access to the device. For the device of exclusive use, this method is required, and foe the device of sharable use, it is optional.
3. **DeviceEnabled** property: Set to **TRUE** to operate the device.
4. Use the device. (Each property, method, event)
5. **DeviceEnabled** property: Set to **FALSE** to disable the device.
6. **ReleaseDevice** method: Called to clear exclusive access to the device.
7. **Close** method: Called to release the service object from the control object.

For details of other usage, refer to "OPOS-APG V1.13" document.

3.2. POS Printer

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

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

The POS printer is the device to be used exclusively.

3.3. Drawer

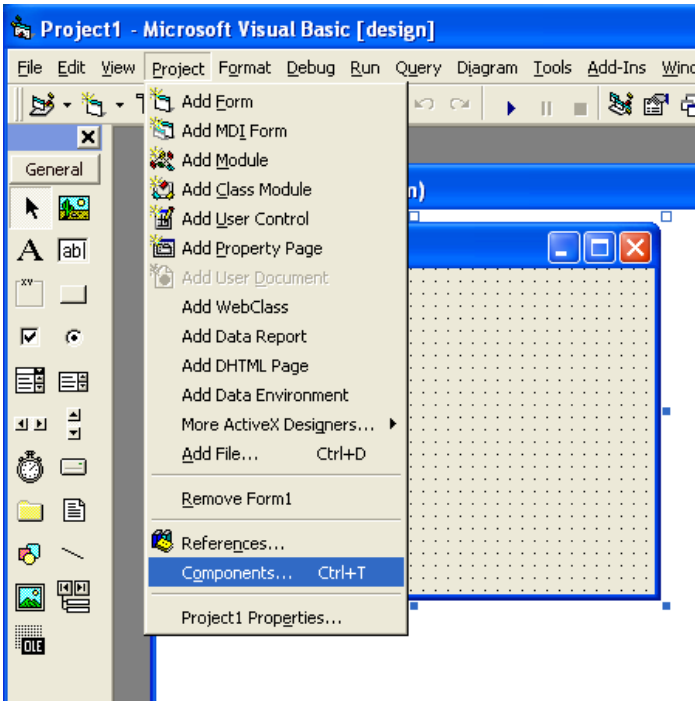
The Drawer can be used in the same way as the POS printer, but all features are executable without executing the ClaimDevice method. However, when exclusive permission is acquired for particular application by the OpenDevice method, the OpenDevice method cannot be executed by the application enabled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

3.4. How to Implement the OPOS Control

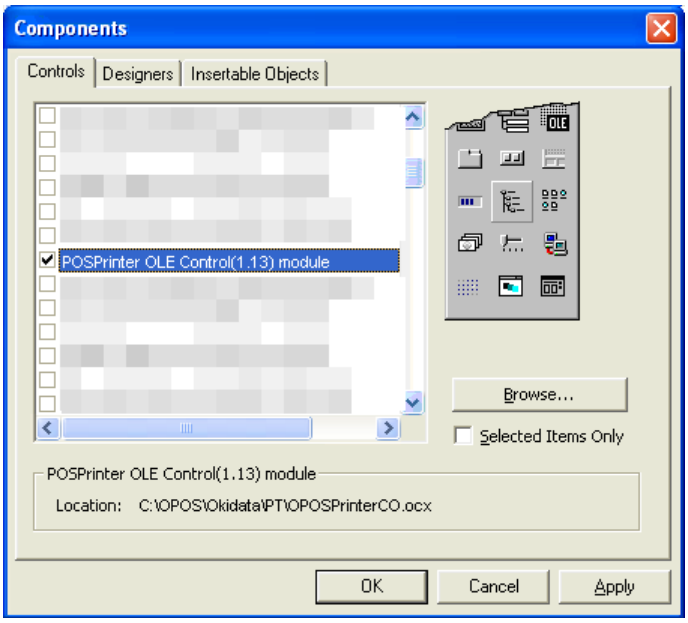
The following steps describe how to implement the OPOS-OCX control in the application.

To Implement Using Visual Basic 6.0

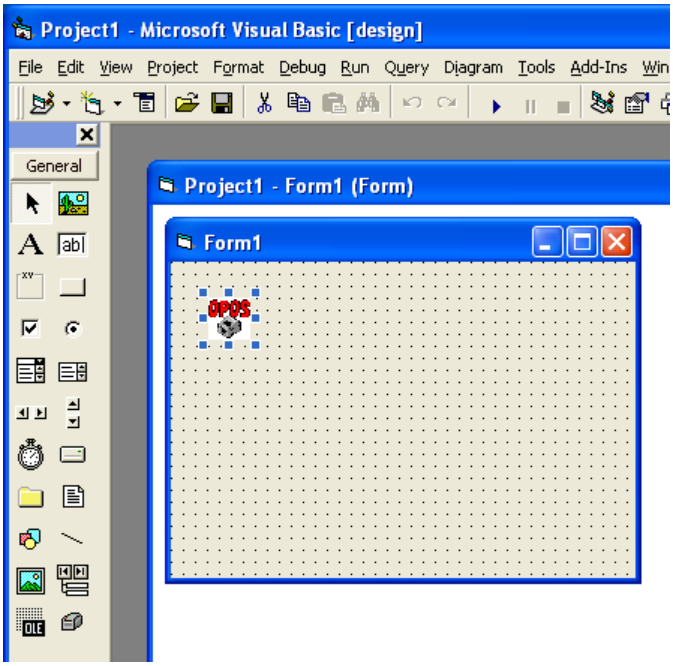
1. Start Visual Basic 6.0 and select the type of the project to be created. In this example, [Standard EXE] is selected. After selection, select [Project] from the menu, and select [Component].



2. When the Component dialog box is displayed, select the [Control] tab -> [POSPrinter OLE Control (1.13) module], and press the [OK] button.



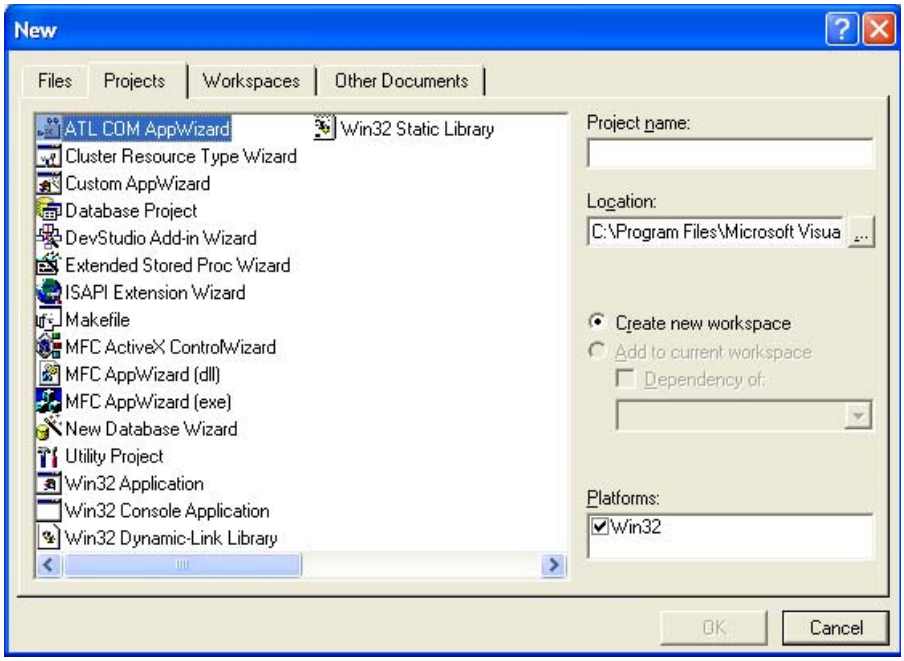
3. Paste the OCX displayed in the toolbar on the form. After this procedure, use it in the same manner as normal OCX.



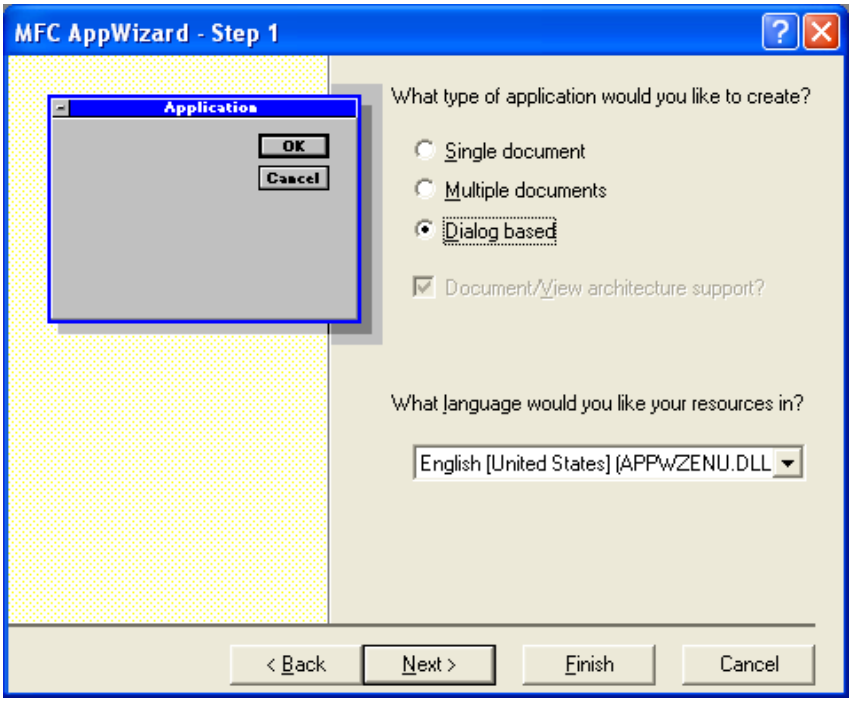
4. For the Drawer, repeat the steps to select [CashDrawer OLE Control (1.13) module].

To Implement Using Visual C++ 6.0

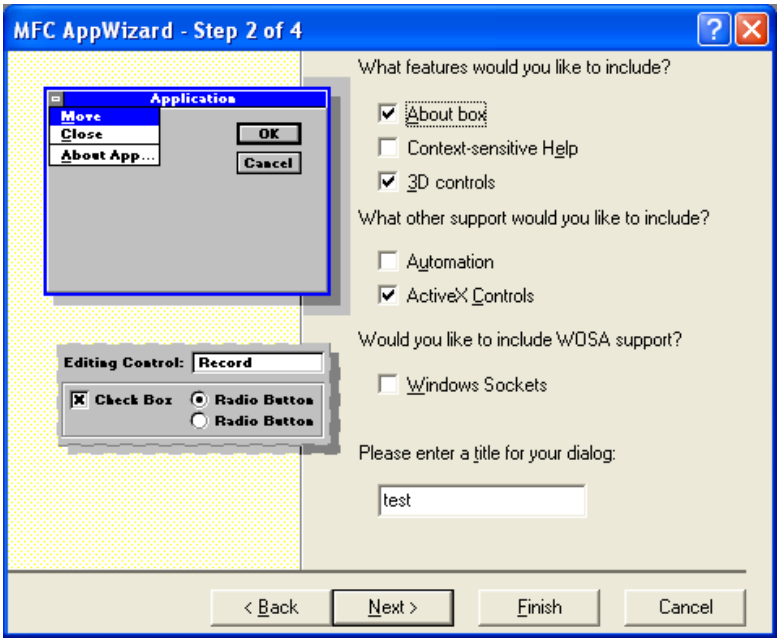
1. Start Visual C++ 6.0, select [Create New] -> [MFC AppWizard(exe)], and press [OK].



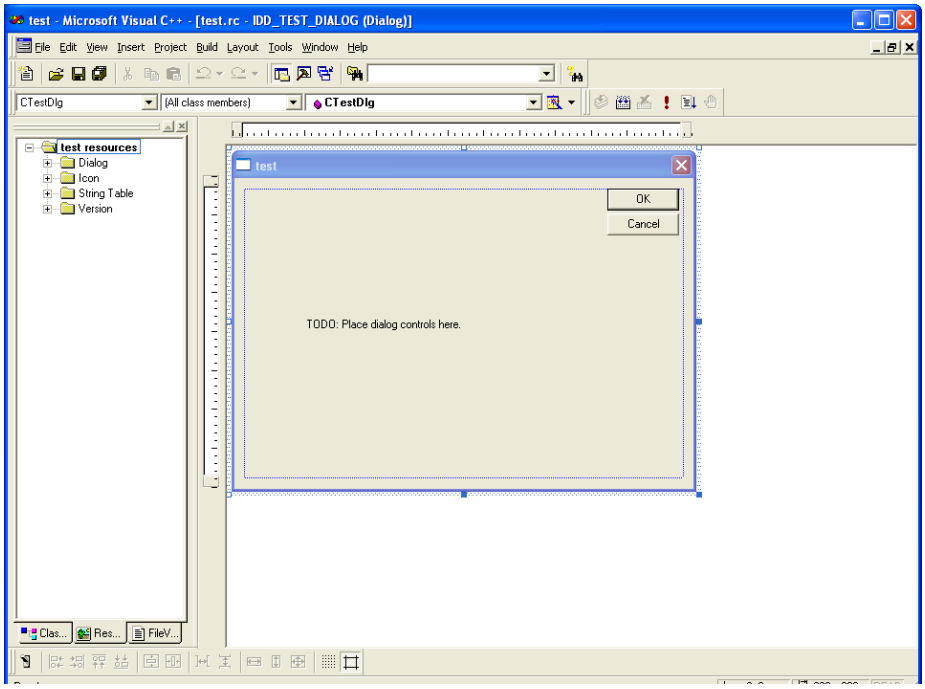
2. In the MFC AppWizard Step 1, select [Dialog Base] and press the [Next] button.



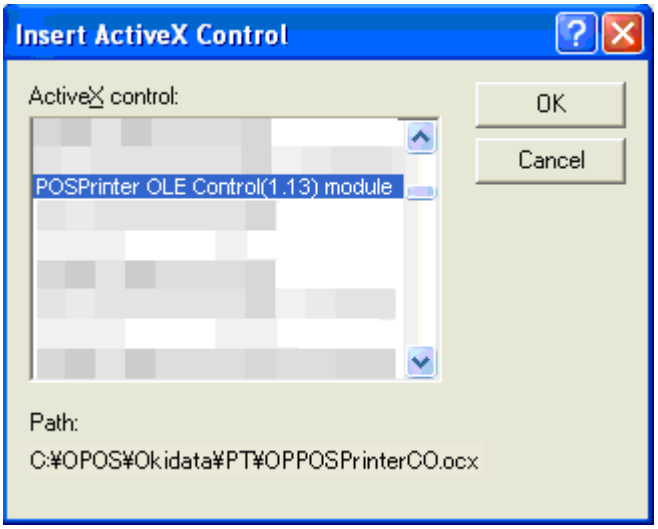
3. Select the ActiveX Control checkbox to be supported. Change other settings according to the type of the project to be created. Select [Next] and perform appropriate setting as required for the rest of the instruction.



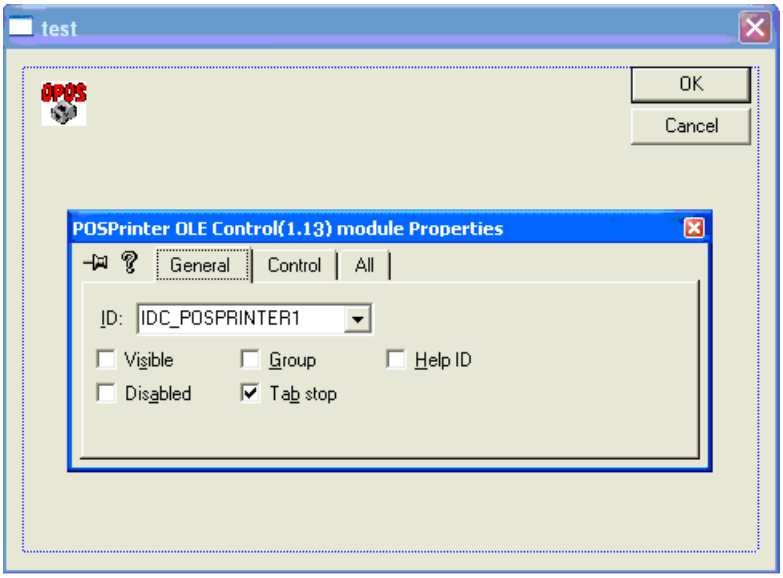
4. When n the project is created, select [Resource View] to display the main dialog box.



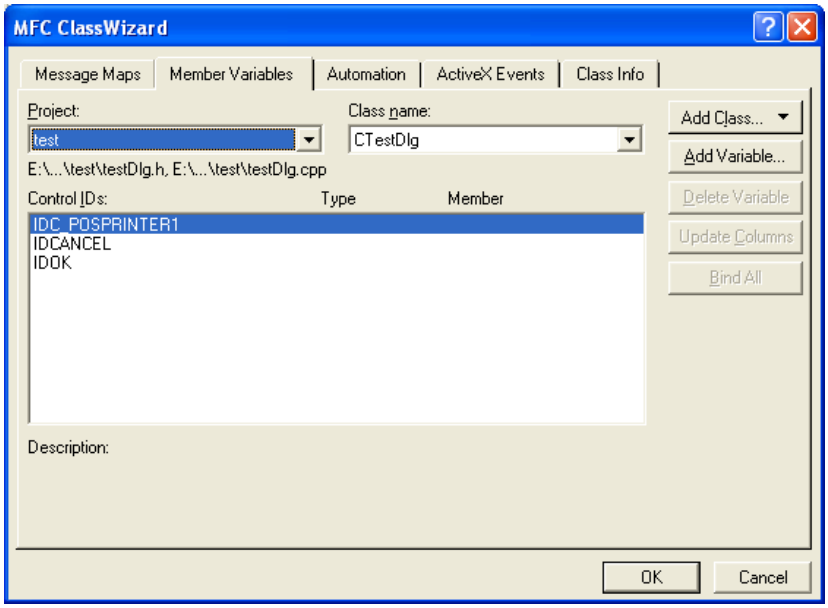
5. On the dialog box, right-click to display the pop-up menu, and select [Insert ActiveX Control]. When the dialog box is displayed, select [POSPrinter OLE Control (1.13) module] and select [OK].



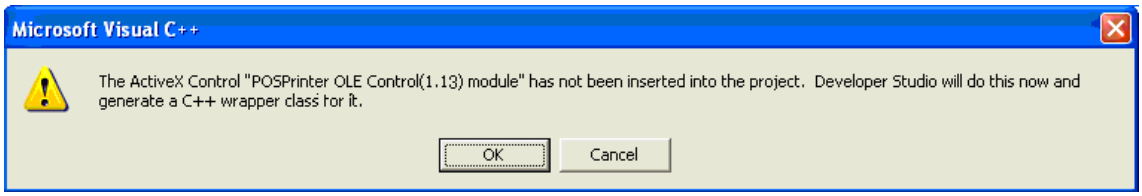
6. When OCX is inserted on the dialog box, select the control and right-click it. From the pop-up menu, select [Property]. Check the ID of the OCX. Clear the [Visible] and [Tab Stop] check boxes.

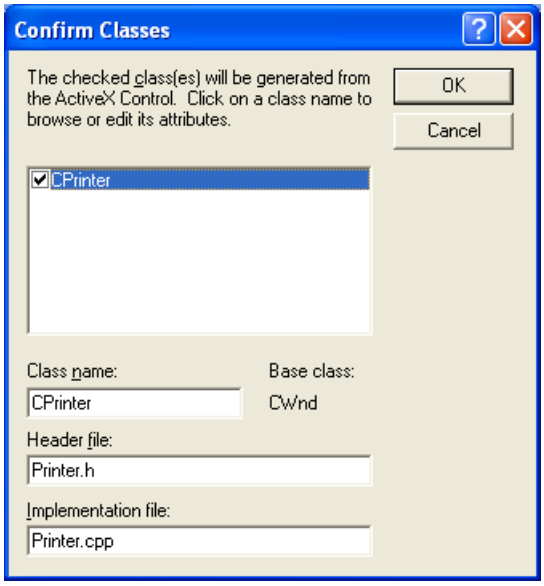


7. Close [Property], repeat the step 6 and select [ClassWizard]. Select the [Member Variable] tab, select the control ID checked in the step 6, and press [Add Variable].

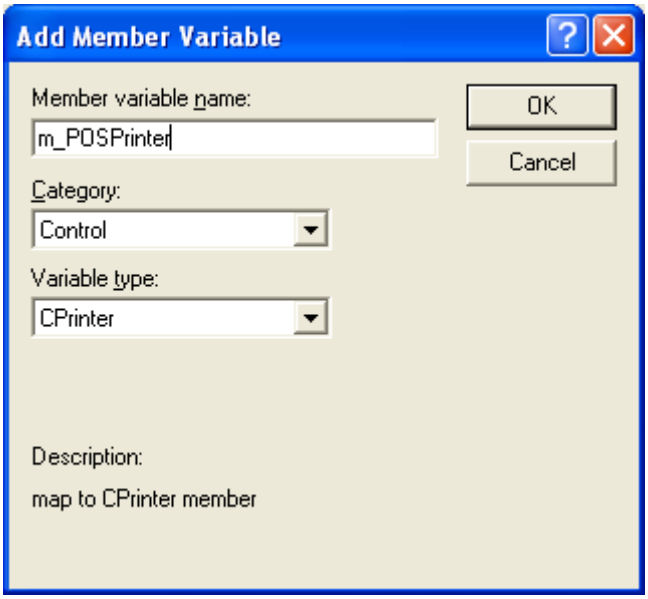


8. When the following dialog box is displayed, press [OK].





9. Set the member variable corresponding to the OPOS control, and exit Class Wizard. Use this member variable for the properties and methods with the OCX.
   Example: m_POSPrinter.Open("PT330-331SERPRT")



10. For the Drawer, repeat the steps to select [CashDrawer OLE Control (1.13) module].

4. OPOS Interface Specifications (Printer)

4.1. List

Properties

Common	Type	Access	May Use After	Initial Value, Conditions
BinaryConversion	Long	R/W	Open	OPOS_BC_NONE(0) Made writable after Open.
CapCompareFirmwareVersion	Boolean	R	Open	TRUE
CapPowerReporting	Long	R	Open	OPOS_PR_STANDARD(1)
CapStatisticsReporting	Boolean	R	Open	FALSE
CapUpdateFirmware	Boolean	R	Open	TRUE
CapUpdateStatistics	Boolean	R	Open	FALSE
CheckHealthText	String	R	Open	""
Claimed	Boolean	R	Open	FALSE
DeviceEnabled	Boolean	R/W	Open & Claim	FALSE Made writable after Open and Claim.
FreezeEvents	Boolean	R/W Open	FALSE Open	FALSE Made writable after Open.
OpenResult	Long	R	None	0
OutputID	Long	R	Open	1
PowerNotify	Long	R/W	Open	OPOS_PN_DISABLED(0) Made writable after Open, and unwritable after Enabled.
PowerState	Long	R	Open	OPOS_PS_UNKNOWN(2000)
ResultCode	Long	R	--	0
ResultCodeExtended	Long	R	Open	0
State	Long	R	--	OPOS_S_CLOSED
ControlObjectDescription	String	R	--	"POS Printer OPOS Control Object"
ControlObjectVersion	Long	R	--	1013XXX

Common	Type	Access	May Use After	Initial Value, Conditions
ServiceObjectDescription	String	R	Open	" POS Printer OPOS Service Object"
ServiceObjectVersion	Long	R	Open	1013XXX
DeviceDescription	String	R	Open	"OP 1 Station Thermal POSPrinter"
DeviceName	String	R	Open	"OP 1 Station Thermal POSPrinter"

Specific	Type	Access	May Use After	Initial Value, Conditions
CapCharacterSet	Long	R	Open	PTR_CCS_KANJI (11)
CapConcurrentJrnRec	Boolean	R	Open	FALSE
CapConcurrentJrnSlp	Boolean	R	Open	FALSE
CapConcurrentPageMode	Boolean	R	Open	FALSE
CapConcurrentRecSlp	Boolean	R	Open	FALSE
CapCoverSensor	Boolean	R	Open	TRUE
CapMapCharacterSet	Boolean	R	Open	TRUE
CapTransaction	Boolean	R	Open	TRUE
CapJrnPresent	Boolean	R	Open	FALSE
CapJrn2Color	Boolean	R	Open	FALSE
CapJrnBold	Boolean	R	Open	FALSE
CapJrnDhigh	Boolean	R	Open	FALSE
CapJrnDwide	Boolean	R	Open	FALSE
CapJrnDwideDhigh	Boolean	R	Open	FALSE
CapJrnEmptySensor	Boolean	R	Open	FALSE
CapJrnItalic	Boolean	R	Open	FALSE
CapJrnNearEndSensor	Boolean	R	Open	FALSE
CapJrnUnderline	Boolean	R	Open	FALSE
CapJrnCartridgeSensor	Long	R	Open	0
CapJrnColor	Long	R	Open	0

Specific	Type	Access	May Use After	Initial Value, Condition
CapRecPresent	Boolean	R	Open	TRUE
CapRec2Color	Boolean	R	Open	The initial value may vary according to the contents of the registry.
CapRecBarCode	Boolean	R	Open	TRUE
CapRecBitmap	Boolean	R	Open	TRUE
CapRecBold	Boolean	R	Open	TRUE
CapRecDhigh	Boolean	R	Open	TRUE
CapRecDwide	Boolean	R	Open	TRUE
CapRecDwideDhigh	Boolean	R	Open	TRUE
CapRecEmptySensor	Boolean	R	Open	TRUE
CapRecItalic	Boolean	R	Open	FALSE
CapRecLeft90	Boolean	R	Open	TRUE
CapRecNearEndSensor	Boolean	R	Open	The initial value may vary according to the contents of the registry.
CapRecPapercut	Boolean	R	Open	TRUE
CapRecRight90	Boolean	R	Open	TRUE
CapRecRotate180	Boolean	R	Open	TRUE
CapRecStamp	Boolean	R	Open	FALSE
CapRecUnderline	Boolean	R	Open	TRUE
CapRecCartridgeSensor	Long	R	Open	0
CapRecColor	Long	R	Open	0
CapRecMarkFeed	Long	R	Open	0
CapRecPageMode	Boolean	R	Open	FALSE
CapRecRuledLine	Boolean	R	Open	FALSE

Specific	Type	Access	May Use After	Initial Value, Condition
CapSlpPresent	Boolean	R	Open	FALSE
CapSlpFullslip	Boolean	R	Open	FALSE
CapSlp2Color	Boolean	R	Open	FALSE
CapSlpBarCode	Boolean	R	Open	FALSE
CapSlpBitmap	Boolean	R	Open	FALSE
CapSlpBold	Boolean	R	Open	FALSE
CapSlpDhigh	Boolean	R	Open	FALSE
CapSlpDwide	Boolean	R	Open	FALSE
CapSlpDwideDhigh	Boolean	R	Open	FALSE
CapSlpEmptySensor	Boolean	R	Open	FALSE
CapSlpItalic	Boolean	R	Open	FALSE
CapSlpLeft90	Boolean	R	Open	FALSE
CapSlpNearEndSensor	Boolean	R	Open	FALSE
CapSlpRight90	Boolean	R	Open	FALSE
CapSlpRotate180	Boolean	R	Open	FALSE
CapSlpUnderline	Boolean	R	Open	FALSE
CapSlpBothSidesPrint	Boolean	R	Open	FALSE
CapSlpCartridgeSensor	Long	R	Open	0
CapSlpColor	Long	R	Open	0
CapSlpPageMode	Boolean	R	Open	FALSE
CapSlpRuledLine	Boolean	R	Open	FALSE

Specific	Type	Access	May Use After	Initial Value, Condition
AsyncMode	Boolean	R/W	Open	FALSE Made writable after Open.
CartridgeNotify	Long	R/W	Open	PTR_CN_DISABLED(0) Unwritable
CharacterSet	Long	R/W	Open, Claim & Enable	998 Made writable after Enabled
CharacterSetList	String	R	Open	"101,102,103,437,850,851,852,857,858,860,863,864,865,866,869,932,998,1252,28592,28597"
CoverOpen	Boolean	R	Open, Claim & Enable	FALSE

Specific	Type	Access	May Use After	Initial Value, Condition
ErrorLevel	Long	R	Open	0
ErrorStation	Long	R	Open	0
ErrorString	String	R	Open	“”
FontTypefaceList	String	R	Open	“”
FlagWhenIdle	Boolean	R/W	Open	FALSE Made writable after Open
MapCharacterSet	Boolean	R/W	Open	TRUE
MapMode	Long	R/W	Open	PTR_MM_DOTS (1) Made writable after Open
PageModeArea	String	R	Open	“”
PageModeDescriptor	Long	R	Open	0
PageModeHorizontalPosition	Long	R/W	Open	0
PageModePrintArea	String	R/W	Open	“”
PageModePrintDirection	Long	R/W	Open	0
PageModeStation	Long	R/W	Open	0
PageModeVerticalPosition	Long	R/W	Open	0
RotateSpecial	Long	R/W	Open	PTR_RP_NORMAL (1) Made writable after Open

Specific	Type	Access	May Use After	Initial Value, Condition
JrnLineChars	Long	R/W	Open, Claim & Enable	0 Unwritable
JrnLineCharsList	String	R	Open	“”
JrnLineHeight	Long	R/W	Open, Claim & Enable	0 Unwritable
JrnLineSpacing	Long	R/W	Open, Claim & Enable	0 Unwritable
JrnLineWidth	Long	R	Open, Claim & Enable	0
JrnLetterQuality	Boolean	R/W	Open, Claim & Enable	FALSE Unwritable
JrnEmpty	Boolean	R	Open, Claim & Enable	FALSE
JrnNearEnd	Boolean	R	Open, Claim & Enable	FALSE

Specific	Type	Access	May Use After	Initial Value, Condition
JrnCartridgeState	Long	R	Open, Claim & Enable	0 Unwritable
JrnCurrentCartridge	Long	R/W	Open, Claim & Enable	0 Unwritable

Specific	Type	Access	May Use After	Initial Value, Condition
RecLineChars	Long	R/W	Open, Claim & Enable	The initial value may vary according to the contents of the registry Made writable after Open.
RecLineCharsList	String	R	Open	The initial value may vary according to the contents of the registry.
RecLineHeight	Long	R/W	Open, Claim & Enable	The initial value may differ according to the contents of the registry Unwritable
RecLineSpacing	Long	R/W	Open, Claim & Enable	The initial value may vary according to the contents of the registry Made writable after Open.
RecLineWidth	Long	R	Open, Claim & Enable	The initial value may vary according to the contents of the registry
RecLetterQuality	Boolean	R/W	Open, Claim & Enable	TRUE Made writable after Open.
RecEmpty	Boolean	R	Open, Claim & Enable	FALSE
RecNearEnd	Boolean	R	Open, Claim & Enable	FALSE
RecSidewaysMaxLines	Long	R	Open, Claim & Enable	The initial value may vary according to the contents of the registry
RecSidewaysMaxChars	Long	R	Open, Claim & Enable	The initial value may vary according to the xml contents.
RecLinesToPaperCut	Long	R	Open, Claim & Enable	The initial value may vary according to the xml contents.
RecBarCodeRotationList	String	R	Open	“0,R90,L90,180”
RecCartridgeState	Long	R	Open, Claim &	PTR_CART_UNKNOWN (268435456)

Specific	Type	Access	May Use After	Initial Value, Condition
			Enable	
RecCurrentCartridge	Long	R/W	Open, Claim & Enable	0 Unwritable
RecBitmapRotationList	String	R	Open	“0,R90,L90,180”

Specific	Type	Access	May Use After	Initial Value, Condition
SlpLineChars	Long	R/W	Open, Claim & Enable	0 Unwritable
SlpLineCharsList	String	R	Open	""
SlpLineHeight	Long	R/W	Open, Claim & Enable	0 Unwritable
SlpLineSpacing	Long	R/W	Open, Claim & Enable	0 Unwritable
SlpLineWidth	Long	R	Open, Claim & Enable	0
SlpLetterQuality	Boolean	R/W	Open, Claim & Enable	FALSE Unwritable
SlpEmpty	Boolean	R	Open, Claim & Enable	FALSE
SlpNearEnd	Boolean	R	Open, Claim & Enable	FALSE
SlpSidewaysMaxLines	Long	R	Open, Claim & Enable	0
SlpSidewaysMaxChars	Long	R	Open, Claim & Enable	0
SlpMaxLines	Long	R	Open, Claim & Enable	0
SlpLinesNearEndToEnd	Long	R	Open, Claim & Enable	0
SlpBarCodeRotationList	String	R	Open	""
SlpPrintSide	Long	R	Open, Claim & Enable	0

Specific	Type	Access	May Use After	Initial Value, Condition
SlpCartridgeState	Long	R	Open, Claim & Enable	0
SlpCurrentCartridge	Long	R/W	Open, Claim & Enable	0 Unwritable
SlpBitmapRotationList	String	R	Open	""

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

List of Methods

Common	Initialization
Open	--
Close	Open
ClaimDevice	Open
ReleaseDevice	Open, Claim
ClearOutput	Open, Claim & Enable *1
CheckHealth	Open, Claim & Enable
CompareFirmwareVersion	Open, Claim & Enable
DirectIO	Open, Claim & Enable *1
ResetStatistics	Open, Claim & Enable
RetrieveStatistics	Open, Claim & Enable
UpdateFirmware	Open, Claim & Enable
UpdateStatistics	Open, Claim & Enable

Specific	Initialization
PrintNormal	Open, Claim & Enable
PrintTwoNormal	Open, Claim & Enable
PrintImmediate	Open, Claim & Enable
BeginInsertion	Open, Claim & Enable
EndInsertion	Open, Claim & Enable
BeginRemoval	Open, Claim & Enable
EndRemoval	Open, Claim & Enable
CutPaper	Open, Claim & Enable
RotatePrint	Open, Claim & Enable
PrintBarcode	Open, Claim & Enable
PrintBitmap	Open, Claim & Enable
TransactionPrint	Open, Claim & Enable
ValidateData	Open, Claim & Enable
SetBitmap	Open, Claim & Enable
SetLogo	Open, Claim & Enable
ChangePrintSide	Open, Claim & Enable
MarkFeed	Open, Claim & Enable
ClearPrintArea	Open, Claim & Enable
PageModePrint	Open, Claim & Enable
PrintMemoryBitmap	Open, Claim & Enable

List of Events

Event	Initialization
DirectIOEvent	Open, Claim & Enable
ErrorEvent	Open, Claim & Enable *1
OutputCompleteEvent	Open, Claim & Enable
StatusUpdateEvent	Open, Claim & Enable

*1: Items required for initialization are different from "OLE for Retail POS Application Programmer's Guide Instructions The 1.13 Version".

4.2. Printing Data and Escape Sequence

This OCX supports the following Escape Sequence.

1) Escape Sequence which operates only when assigned time

Name	Data	Contents
Paper Cut	ESC |#P	It cuts the receipt paper. The character '#' is the character string of ASCII decimal number which shows the percentage of required cutting. It is possible to omit '#'. When the value is between '1' to '99', partial cutting is performed. When the value is '100' or omitted, full cutting is performed. When the value is other than any value between '1' to '100', it is ignored In addition to that, if there are data buffered to POS Printer, (in the case that POS Printer does not print even though printing request is done) it cannot do cutting. To execute cutting, it must be at the head of the line. It is also unavailable in 90 degrees rotating to the left or to the right by **RotatePrint** Method, and back in operation after clearing 90 degrees rotating to the left or to the right.
Feed and Paper Cut	ESC |#fP	It cuts the receipt paper after feeding the paper for the lines of **RecLinesToPaperCut**. Character '#' is defined by "Paper Cut" Escape Sequence. In addition to that, if there are data buffered to POS Printer, (in case that POS Printer does not print even though printing request is done) it cannot do cutting. To execute cutting, it must be at the head of the line. It is also unavailable in 90 degrees rotating to the left or to the right by **RotatePrint** Method, and back in operation after clearing 90 degrees rotating to the left or to the right.
Feed, Paper Cut, and Stamp	ESC |#sP	It is not supported.

Name	Data	Contents
Print bitmap	ESC |#B	It prints the bitmap saved by **SetBitmap** Method. '#' is the Bitmap number and supports 20 bitmap printing '1' to '20'. It is possible to change printing quality by changing **RecLetterQuality** property value in printing. As for handling printing quality, it is same as **PrintBitmap** Method. When '#' is omitted, it is handled as character string. Printing is available during 90 degrees rotating to the left or to the right by the **RotatePrint** method, but printing may not be performed properly because printing area of the bitmap size is not calculated. When '#' is omitted, it is regarded character string data starting with the character "B". When the number that is not stored in the **SetBitmap** method, the print command is issued to the printer, but printing is not performed.
Print top logo	ESC |tL	It prints the top logo saved by **SetLogo** Method.
Print bottom logo	ESC |bL	It prints the bottom logo saved by **SetLogo** Method.
Fire stamp	ESC |sL	It is not supported.
Feed line	ESC |#lF	Feeds the paper forward by lines. The character '#' is replaced by an ASCII decimal string telling the number of lines to be fed. If '#' is omitted, then one line is fed. '#' supports the values from '1' to '255'. (If the value is smaller than '1', the command is not executed, and if the value is larger than '255', the command is executed regarding that '255' is assigned.) If print data is not presence, line feed operation is executed according to the amount of line feed, and if print data is presence, the height of the print data is fed. If the value specified for "#" exceeds 35.4 in (approx. 900 mm), the command is executed feeding the paper by 35.4 in (approx. 900 mm).. In 90 degrees rotating to the left or to the right by **RotatePrint** Method, it prints next printing location after Returns of feed-assigned lines.

Name	Data	Contents
Feed unit	ESC |#uF	Feeds the paper forward by the units defined with **MapMode**. The character '#' is replaced by an ASCII decimal string telling the number of units to be fed. If '#' is omitted, then one unit for each **MapMode** is fed. MapMode = PTR_MM_DOTS(1) '#' supports the values from '1' to '127'. (If the value is smaller than '1', the command is not executed, and if the value is larger than '127', the command is executed regarding that '127' is assigned.) MapMode = PTR_MM_TWIPS(2) '#' supports the values from '1' to '903'. (If the value is smaller than '1', the command is not executed, and if the value is larger than '903', the command is executed regarding that '903' is assigned.) MapMode = PTR_MM_ENGLISH(3) '#' supports the values from '1' to '627'. (If the value is smaller than '1', the command is not executed, and if the value is larger than "627', the command is executed regarding that "627' is assigned.) MapMode = PTR_MM_METRIC(4) '#' supports the values from '1' to '1594'. (If the value is smaller than '1', the command is not executed, and if the value is larger than '1594', the command is executed regarding that '1594' is assigned.) The line feed setting of the printer does not affect the amount of line feed. It is executed on the halfway of the line, and when the specified amount of line feed is less than 1 line, then 1 line is fed In 90 degrees rotating to the left or to the right by **RotatePrint** Method, it will print after the interval of the value of next printing location assigned by unit feed.
Feed reverse	ESC |#rF	It is not supported.

Name	Data	Contents
Send embedded data	ESC |#rE	The successive character string of "#E" is passed to the device without any change. The character '#' is replaced by an ASCII decimal string specifying the number of bytes following the escape sequence to be passed directly to the device. If '#' is omitted, it is not regarded as the escape sequence, and handled as print data. Character string cannot output a control code and the cord of 80H - FFH as expected. In this case, set it in either of OPOS_BC_NIBBLE(1), OPOS_BC_DECIMAL(2) by a **BinaryConversion** property, and set printing data. When the print data specified by '#' is not set after the escape sequence is specified, available print data is sent. (Example: When ESC|2E"a" is specified, only "a" is sent because the character string is set only for one byte.) In rotate printing 90 degrees to the left or to the right by the **RotatePrint** method, the data column specified by Send embedded data is not counted as the character string, the width cannot be calculated. Adjust the printing width by inserting empty space and so on.
Barcode printing (Refer to the next page)	ESC |#rR	Prints the barcode. The character '#' is replaced by an ASCII decimal string and the number of characters following the R to use in the definition of the characteristics of the barcode to be printed. See details below. The barcode may be printed during rotate printing 90 degrees to the left or to the right by the **RotatePrint** method, but printing may not be performed normally because the print area is not calculated by the specified barcode width. When the other character string data specified exceeds the barcode width, printing is executed. In the **RotatePrint** method, please use it without specifying **PTR_RP_BARCODE**. Even if the **RotateSpecial** property is specified, the rotation print is not done. The available width that can be set by the parameter is up to the value of the **RecLineWidth** property and is not affected by the **RotateSpecial** property.

Starting with Release 1.13, the application can use the ESC|#R escape sequence to print barcodes in-line with other print commands. The character '#' is the number of characters following the R to use in the definition of the characteristics of the barcode to be printed.
In the data following the R, other lower case letters and numbers are used to identify different values. The same value definitions as defined for the printBarCode method headers and definitions are used for the various barcode values. Converting to string the values from the definitions are consistent.
The attribute symbols are defined as follows:

s	symbology
h	height
w	width
a	alignment
t	human readable text position
d	start of data
e	end of sequence

The attributes must appear in the order specified in the above list. (It cannot be omitted)
Using a basic UPCA, center aligned, with bottom text, 200 dots height and 400 dots wide, the command is as follows:
ESC|33Rs101h200w400a-2t-13d123456789012e

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

```
const LONG PTR_BCS_UPCA                 = 101;    // Digits
const LONG PTR_BC_CENTER                = -2;
const LONG PTR_BC_TEXT_BELOW            = -13;
```

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

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

2) Escape Sequence Operating in Printing

It has characteristics that are remembered until explicitly changed.

Name	Data	Contents
Font typeface selection	ESC |#fT	It is not supported.

3) Escape Sequence Operating at the Time of Printing

It has the characteristics that are reset at the end of each print method, by an explicit reset (where applicable), or by a “Normal” sequence.

Name	Data	Contents
Bold	ESC |bC	Prints in bold. If ‘!’ is specified then bold is disabled.
Underline	ESC |#uC	Prints with underline. The character ‘#’ is replaced by an ASCII decimal string telling the thickness of the underline in printer dot units. Underlines of one dot or two dots are only supported. If it is omitted, underline of one dot is printed.
Italic	ESC |iC	It is not supported.
Alternate color	ESC |#rC	It is not supported.
Red	ESC |rC	It prints with the second color of the receipt. Printing is possible only when "Printing Color" of Printer Setting is "Two Colors". When Color = mono is set in the setting program, printing is not affected by specifying this escape sequence.
Reverse video	ESC |rvC	Prints in a reverse video format. If ‘!’ is specified then reverse video is disabled.
Shadowing	ESC |#sC	It is 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	It prints characters, enlarging them horizontally. The character '#'is the ASCII decimal character string which shows horizontally enlarging rate. It supports form one to eight times. If "#" is omitted, it prints in life-size.
Scale vertically	ESC |#vC	It prints characters, enlarging them vertically. The character '#' is the ASCII decimal character string which shows vertically enlarging rate. It supports form one to eight times. At the time of "#" abbreviation, it prints in life-size.
RGB Color	ESC |#fC	It is not supported.
Center	ESC |cA	It aligns the text, from there after, in the center. It is not available unless you assign at the head of the line. It is unavailable in 90 degrees rotating to the left or to the right by RotatePrint Method.
Right justify	ESC |rA	It aligns the text, from there after, to the right. It is not available unless you assign at the head of the line. It is unavailable in 90 degrees rotating to the left or to the right by RotatePrint Method.
Normal	ESC |N	It regains POS Printer attribute to the normal state. It is impossible to regain from Centering or Align Right, unless you assign at the head of the row.

Name	Data	Contents
SubScript	ESC |tbC	It is not supported.
SuperScript	ESC |tpC	It is not supported.

4.3. Common Properties

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

We have two kinds of properties: Read-Only and Read/Write. If a property is for writing, R/W will be shown at the side of each property.

Only if return value has a special meaning, it will be shown. As for error, in case that access is done without satisfying initializing condition, refer to 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 original 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 values of **BinaryConversion** are as follows.

Value	Meaning
OPOS_BC_NONE (0)	Data is not converted and one character of Bstring is contained for one byte. (Default)
OPOS_BC_NIBBLE(1)	Each byte is converted to two characters. (By this option, conversion between Binary and ASCII is the fastest.) Each data byte is converted just like below. The first character = 0x30 + 7-4 byte of data byte The second character = 0x30 + 3-0 byte of data byte Example: 154=0x9A (byte value) is converted to 0x39 0x3A (character) (character string "9:"). So this conversion way is deferent from usual hexadecimal ASCII character conversion to convert byte value (154=0x9A) to code (0x39 0x41)(character string "9A").
OPOS_BC_DECIMAL(2)	Each byte is converted to three characters. VAL(*string*) is used with three characters for converting from ASCII to Binary. RIGHT("^^"+STR(*byte*),3) is used to form three ASCII characters from each byte. '^' means space character. Exampl1: 154=0x9A (Byte Value) becomes 0x31 0x35 0x34(Characters)(=Character string "154"). Example2: 8 (byte Value) becomes 0x30 0x30 0x38(Character)(=Character 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 responsibilities to convert the character string to the format assigned to **BinaryConversion,** before setting property and delivering method parameter.

This property is initialized to OPOS_BC_NONE(0) by **Open** Method.

Return Value

When this property is set, either of the following values will be contained in **ResultCode** Property

Value	Meaning
OPOS_SUCCESS(0)	It succeeds s in Property setting.
OPOS_E_ILLEGAL(106)	Illegal value is assigned.

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 to **TRUE** by the **Open** method.

CapPowerReporting Property

Syntax

LONG CapPowerReporting;

Remarks

It identifies power reporting ability. The value to show power reporting ability is just the following.

Value	Meaning
OPOS_PR_STANDARD(1)	SO can judge two kinds of power states and can report it. (ONLINE and OFF_OFFLINE)

This property is initialized by **Open** Method.

CapStatisticsReporting Property

Syntax

BOOL CapStatisticsReporting;

Remarks

This property is initialized to **FALSE** by the **Open** method. Statistics reporting is not supported.

CapUpdateFirmware Property

Syntax

BOOL CapUpdateFirmware;

Remarks

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

CapUpdateStatistics Property

Syntax

BOOL CapUpdateStatistics;

Remarks

This property is initialized to **FALSE** by the **Open** method. Statistics reporting is not supported.

CheckHealthText Property

Syntax

BSTR CheckHealthText;

Remarks

It keeps the **CheckHealth** Method result called just before. The following are examples of the check.

- In case of Internal "Internal HCheck: Successful", "Internal HCheck: OFF/OFFLINE"
- In case of External "External HCheck: Not Supported"
- In case of Interactive "Interactive HCheck: Not Supported"

This value is initialized to null character before the first **CheckHealth m**ethod call.

Claimed Property

Syntax

BOOL Claimed;

Remarks

TRUE: The device is claimed for exclusive access.

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

Claimed Property value is initialized to **FALSE** by **Open** Method.

ControlObjectDescription Property

Syntax

BSTR ControlObjectDescription;

Remarks

" POS Printer OPOS Control Object" is set.

Identifies the Control Object. It is a character string identifying the Control Object and the company that produced it and always readable.

This property is readable at any time.

ControlObjectVersion Property

Syntax

LONG ControlObjectVersion;

Remarks

"1013XXX" is set. Holds the Control Object version number. Following three version levels are specified:

Version Level	Description
Major	The "millions" place. Holds the OPOS major version level.
Minor	The "thousands" place. Holds the OPOS minor version level. This is always set to 10 since this OPOS control conforms to OPOS version 1.13.
Build	The "units" place. Updated when corrections are made to the Control Object.

This property is always readable. (XXX varies depending on the time the Control Object is distributed.)

DeviceDescription Property

Syntax

BSTR DeviceDescription;

Remarks

"OP 1 Station Thermal POSPrinter" is set.

This property is a character string identifying the device, and holds the device and any pertinent information about it

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

DeviceEnabled Property R/W

Syntax

BOOL DeviceEnabled;

Remarks

TRUE:

The device is made to enable (Operation state). If converted to **TRUE,** it is made to enable.

FALSE:

The device is made to disable. If converted to **FALSE**, it is made to disable.

Before this device is used, application must set this property **TRUE.**

Also, while **DeviceEnable** is **TRUE**, Device Connection State (**PowerReporting**) is reported. This property is initialized to **FALSE** by **Open** Method.

Return Value

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

Value	Meaning
OPOS_SUCCESS(0)	It succeeds in Property setting.
OPOS_E_NOTCLAIMED(103)	In order to make exclusive use device enable, it must acquire exclusive access right, before that.
OPOS_E_NOHARDWARE(107)	POS Printer is OFF or OFFLINE or the cable is not connected. Clear the problem, and then execute the property again. (* In the case of USB interface, even if a POS printer is connected, this error occurs when a serial number set by registry is different from a serial number set by a POS printer.) *In case of LAN interface model, same error occurs that if both of POS printer's IP addresses and the registry's are different, even the POS printer is connected in network.
OPOS_E_FAILURE(111)	The connection to the device is failed. Carry out once again after having confirmed whether there is a connection port and whether a connection port is not used by other programs. For USB interface, confirm two unsetting printers or more are not connected at the same time, and confirm it is not a printer used in the other port (for instance, when XXXUSBPRT is used, XXXUSB2PRT). The mistake might be found in the setting of plural connection. Set up one again.
OPOS_E_TIMEOUT(112)	Connection to the POS Printer could not be established. There is the possibility of cover open or running out of paper.
OPOS_E_BUSY(113)	Setting the property is failed because processing . Set the property after processing.
The others	Refer to **ResultCode** Property.

DeviceName Property

Syntax

BSTR DeviceName;

Remarks

"OP 1 Station Thermal POSPrinter" is set up.

This property shows the device and the any pertinent information about it. This is a short version of **DeviceDescription** property and should be limited to 30 characters.

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

FreezeEvents Property R/W

Syntax

BOOL FreezeEvents;

Remarks

In case of **TRUE**, Event is not reported from Control.

Event is kept by Control till the freeze is terminated.

In case of **FALSE**, if Event is reported from Control. If there is any Event kept during freeze, that Event is reported when **FreezeEvents** are converted to **FALSE**.

If interrupt by Event is undesirable, Application can select Event freeze. In case that **ErrorEvent** is frozen, **State** Property becomes **OPOS_S_BUSY(3)**. In such case, Control cannot be closed. In this case, cancel the event frozen by **ClearOutput** Method, or set **TRUE** and execute **Close** Method after **ErrorEventis** originated.

This property is initialized to **FALSE** by **Open** Method.

Return Value

At the time of this property setting, the following values will be contained to **ResultCode** Property.

Value	Meaning
OPOS_SUCCESS(0)	It succeeds in property setting.

OpenResult Property

Syntax

LONG OpenResult;

Remarks

It keeps the latest result by **Open** Method. The values of **OpenResult** are just the following.

Value	Meaning
OPOS_SUCCESS(0)	Open is succeeded.
EOPEN_ALREADYOPEN(301)	It is already opened.
EOPEN_REGBADNAME(302)	There is no assigned Device Name Key in the registry, or Device Name is not assigned.
EOPEN_REGPROGID(303)	Default value of Device Name Key is not readable, or it could not convert Programmatic ID kept there to effective Class ID.
EOPEN_CREATE(304)	It could not generate Service Object Instance, or could not obtain IDispatch Interface.
EOPEN_BADIF(305)	Service Object does not support one or more methods required by assignment version. It is possible that Device Name setting differs from POS Printer OPOS Service Object.

This property is initialized by **Open** Method.

OutputID Property

Syntax

LONG OutputID;

Remarks

It has the identifier to uniquely identify asynchronous request. (Asynchronous Respond Method Call when **AsyncMode** Property is set to **TRUE**.)

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

Output ID Number is numbered cyclically between 1 and 65535.

PowerNotify Property R/W

Syntax

LONG PowerNotify;

Remarks

This is the Power Notify Function type set by Application.

The values to show Power Notify Function are just the following.

Value	Meaning
OPOS_PN_DISABLED(0)	Control does not give any power notification to Application. **StatusUpdateEvent** concerning Power Notification is not given and any setting is not done to **PowerState** Property. (Default Value)
OPOS_PN_ENABLED(1)	When **DeviceEnabled** is set to **TRUE**, Control does **StatusUpdateEvent** notification and **PowerState** Property update.

PowerNotify Property can be set during Device is disable, which means when **DeviceEnabled** Property is **FALSE**.

Return Value

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

Value	Meaning
OPOS_SUCCESS(0)	It succeeds in property setting.
OPOS_E_ILLEGAL(106)	One of the following happens: Device is already enabled. Property setting value is illegal.
The others	Refer to **ResultCode** items.

PowerState Property

Syntax

LONG PowerState;

Remarks

While **PowerNotify** is **OPOS_PN_ENABLED(1)**, Present Device Power State is set up.

The values to show power state are just the following.

Value	Meaning
OPOS_PS_UNKNOWN(2000)	It cannot judge the device power state according to one of the following reasons. (Default Value) When **PowerNotify** = **OPOS_PN_DISABLED(0),** Power Notify Function is disable. When **DeviceEnabled** = **FALSE**, Power State Monitor does not operate till Device becomes enable.
OPOS_PS_ONLINE(2001)	Device power is ON and at the same time is ready.
OPOS_PS_OFF_OFFLINE(2004)	Device power is OFF, or is not connected to the body. As for recovery way, refer to **ResultCode** Property, OPOS_E_NOHARDWARE(107).

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

When **PowerNotify** is set to OPOS_PS_ENABLED(1) and **DeviceEnabled** is **TRUE**, then this property is updated as the SO detects power condition changes.

ResultCode Property

Syntax

LONG ResultCode;

Remarks

Each property sets this property. It is set when it obtains property, or when it set writable property.

This property is always readable. It returns **OPOS_E_CLOSED(101)** till it calls up **Open** Method.

The result code values are just the following.

Value	Meaning
OPOS_SUCCESS(0)	Normal Operation
OPOS_E_CLOSED(101)	You try to access the closed device.
OPOS_E_NOTCLAIMED(103)	You try to access Exclusive Use Device for which you need to obtain exclusive access right before using method/property setting process.
OPOS_E_NOSERVICE(104)	Control cannot communicate to Service Object. Perhaps, you must correct Set Up Error or Configuration Error.
OPOS_E_DISABLED(105)	It cannot be executed when Device is disabled.
OPOS_E_ILLEGAL(106)	You try to execute unavailable operation to Device or operation not supported. Or you use unavailable parameter value.
OPOS_E_NOHARDWARE(107)	POS Printer is OFF or OFFLINE.
OPOS_E_FAILURE(111)	Even though Device is connected to System, Power is ON, and it is ONLINE, Device cannot execute the requested process.
OPOS_E_TIMEOUT(112)	The Service Object waiting for the response from Device does time-out, or the Control waiting for the response from Service Object does time-out.
OPOS_E_BUSY(113)	Present SO state cannot accept this request. For example, when asynchronous Output is executed, some methods cannot accept it.
OPOS_E_EXTENDED(114)	Particular Error State occurs. Error State Code can be confirmed with **ResultCodeExtended** Property

ResultCodeExtended Property

Syntax

LONG ResultCodeExtended;

Remarks

When **ResultCode** is **OPOS_E_EXTENDED(114)**, particular-to-class error information value described in device class explanation to this property is set.

When **ResultCode** is another value**,** Service Object can set particular-to-SO value to this property. These values have meaning only when Application processes them adding particular-to-SO values.

This OPOS Control has the following values.

ResultCodeExtended	Constant Name(Contents)	Details
201	OPOS_EPTR_COVER_OPEN	Printer Cover is open. To recover from Error, close the cover. In this case, you don't have to make OPOS disable, **ReleaseDevice**, **Close**. If Cover opens in printing, after closing cover, the not-yet-printed data will be printed. However, there is the possibility of interrupted printing space among the printing data.
203	OPOS_EPTR_REC_EMPTY	Run-out-of-receipt-paper. To recover from error, feed receipt paper. At this time, there is no need to make OPOS disable, and **ReleaseDevice**, **Close**. If run-out-of-paper happens in printing, the printing data not yet printing will be printed after feeding receipt paper.
206	OPOS_EPTR_TOOBIG	Bitmap is too big for the Printer to process. Check the bitmap file.
207	OPOS_EPTR_BADFORMAT	The format is not correct as a bitmap file. There is a possibility that the file name is not correct or the contents of the file are corrupt.

ResultCodeExtended	Constant Name(Contents)	Details
281	OPOS_FIRMWARE_BAD_FILE	The format is not correct as a firmware file. There is a possibility that the file name is not correct or the contents of the file are corrupt.
10001	OPOS_FIT_EPTR_BLACKMARK	Black Mark Sensor Error happens. There is the possibility that paper cannot respond to Black Mark. At such time, there is no need to make OPOS disable, and ReleaseDevice, Close. Change the receipt to the one responding to Black Mark. *It does not occur when Black Mark Error Sensor is not on. (Check Black Mark Enable.)
10003	OPOS_FIT_EPTR_FATAL	Fatal error occurs in Printer. There is no recover way. Do OPOS **Close**.
10006	OPOS_FIT_EPTR_OVERHEAT	Printer Header Overheat Error happened. To recover from the error, leave it for a while until the head temperature cools down and the error does not occur again. At such time, there is no need to make OPOS disable, and **ReleaseDevice**, **Close**. If Head Overheat occurs in printing, the printing data not yet printing will be printed after the head temperature recovery. However, there is the possibility of interrupted space of printing of the printing data.
10008	OPOS_FIT_EPTR_CUTTERJAM	Cutter Jam Error happened. To recover from the error, open the cover and remove the cause (such as the jammed paper), and close the cover. If the cover does not open, follow procedure on Printer UsersManual.

ServiceObjectDescription Property

Syntax

BSTR ServiceObjectDescription;

Remarks

" POS Printer OPOS Service Object" is set.

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

ServiceObjectVersion Property

Syntax

LONG ServiceObjectVersion;

Remarks

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

State Property

Syntax

LONG State;

Remarks

It shows the present state of Control.

Value	Meaning
OPOS_S_CLOSED(1)	Control is closed. (Default)
OPOS_S_IDLE(2)	Control is in normal state and not busy.
OPOS_S_BUSY(3)	Control is in normal state and busy because it is outputting.
OPOS_S_ERROR(4)	Error is reported, and if you want to restart usual I/O, Application must return Control into normal state.

This property is always readable.

4.4. Common Methods

The following section describes the methods provided commonly to the POS printer.

CheckHealth Method

Syntax

LONG CheckHealth (LONG *Level*);

Level parameter shows the type of health check executed with device. The following values can be assigned.

Value	Meaning
OPOS_CH_INTERNAL(1)	Performs on line check. The result is set to the **CheckHealthText** property as follows: When the POS Printer is connected to POS and the power is turned on, "Internal HCheck: Successful" is set in the **CheckHealth** property. When the POS Printer is not connected to POS, or the power is turned off, "Internal HCheck: OFF/OFFLINE" is set in the **CheckHealth** property. Returns OPOS_SUCCESS(0).
OPOS_CH_EXTERNAL(2)	It is not supported with this OCX.. "Internal HCheck: NOT Supported" is set in the **CheckHealth** property. Returns OPOS_ILLEGAL(106).
OPOS_CH_INTERACTIVE(3)	It is not supported with this OCX.. "Internal HCheck: NOT Supported" is set in the **CheckHealth** property. Returns OPOS_ILLEGAL(106).

Remarks

It is called at the time of device condition test. The result of this method is contained in **CheckHealthText** Property. **CheckHealth** Method is always synchronous.

Return Value

Any of the following values will be returned and be contained in **ResultCode**.

Value	Meaning
OPOS_SUCCESS(0)	It shows that Health Check Procedure starts properly and that, when confirmed that, the device is working properly. However you cannot decide about its properness until you see the result of the test.
OPOS_E_ILLEGAL(106)	*Level* parameter not supported is assigned.
OPOS_E_NOHARDWARE(107)	After assigning and executing OPOS_CH_INTERNAL(1), it is Offline.
OPOS_E_BUSY(113)	It cannot be executed while Output is ON.
The others	Refer to the items of **ResultCode**.

ClaimDevice Method

Syntax

LONG ClaimDevice (LONG *Timeout*);

Timeout Parameter shows maximum waiting time (in ms.) till it acquires exclusive access right.

In case of zero, even if it cannot acquire the Device Exclusive Access Method, it returns the result immediately.

If **OPOS_FOREVER(-1)** is set, Method waits as long as till it can acquire exclusive access right.

Remarks

This method is called when exclusive access is required to device.

POS Printer Device cannot be used without acquiring exclusive access right.

In case of success, **Claimed** Property is set to **TRUE**.

If **ClaimDevice** Method is executed, it establishes connection to POS Printer Device, and confirms the process possibility. If the process is possible, it requires particular data and **ClaimDevice** Method, and ends normally.

Return Value

One of the following values will be returned and contained to **ResultCode** Property.

Value	Meaning
OPOS_SUCCESS(0)	Exclusive access right is confirmed. **Claimed** Property is **TRUE.** If this application has already accessed the device exclusively, it is returned, too.
OPOS_E_ILLEGAL(106)	Unavailable ***Timeout*** Parameter is assigned.
OPOS_E_TIMEOUT(112)	Other application tries to access the device exclusively and waits for being released. But ***Timeout*** Time (in ms.) has passed. Or, POS Printer Device does not become POS-Printer-Device-Process-Possible State even after ***Timeout*** Time (in ms.) passes.
OPOS_E_EXTENDED(114)	Refer to the **ResultCodeExtended** property.

ClearOutput Method

Syntax

LONG ClearOutput ();

Remarks

It is called when to clear all the output devices buffered by asynchronous issue of each method of **PrintNormal**, **CutPaper**, **RotatePrint**, **PrintBarCode**, **PrintBitmap**, **TransactionPrint.** Also, it does release when it is in rotation mode or in batch processing mode by **RotatePrint** Method or **TransactionPrint** Method.

Pending Output Error Event (which is in case that **FreezeEvents** are wanting for being set to **FALSE**) is also cleared.

Return Value

Following values will be returned and contained to **ResultCode** Property.

Value	Meaning
OPOS_SUCCESS(0)	Output is cleared.
OPOS_E_FAILURE(111)	Device is accessed exclusively by other process.
The others	Refer to items of **ResultCode**.

Close Method

Syntax

LONG Close ();

Remarks

It is called when to release Device and its resource.

If **DeviceEnabled** Property is **TRUE,** Device is forced to disable.

If **Claimed** Property is **TRUE,** at first, exclusive access will be released.

Don't execute at the time of Event Processing. (within Event Handler)

Return Value

One of the following values will be returned and contained to **ResultCode** Property.

Value	Meaning
OPOS_SUCCESS(0)	Device is disabled and closed.
OPOS_E_BUSY(113)	Asynchronous processing is under way.
The others	Refer to items of **ResultCode**.

CompareFirmwareVersion Method

Syntax

LONG CompareFirmWareVersion (String *FirmWareFileName***, Long** *result***);**

Parameter	Meaning
FirmWareFileName	Holds the name of the firmware file whose version is to be compared against the firmware version of the device.
result	Returns the result of the comparison.. 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. OPOS_CFV_FIRMWARE_UNKNOWN(5) Indicates that a relationship between the two firmware versions could not be determined.

Remarks

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

The version of the firmware is represented by the value of first six successive numbers found in the *FirmWareFileName* parameter. For example, when "01rm012345.hx5" is set in the *FirmWareFileName* parameter, "012345" is recognized as the version of the firmware file to be compared. The version of the POS printer can be acquired by sending the command to the POS printer every time this method is executed.

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

When the first four places are equal, then last two places are compared as the values. When the version of the firmware file is smaller, OPOS_CFV_FIRMWARE_OLDER (1) is set. When the version of the POS printer is smaller,

OPOS_CFV_FIRMWARE_NEWER(3) is set. When those are equal,

OPOS_CFV_FIRMWARE_SAME(2) is set.

Return Value

One of the following values will be returned and contained to **ResultCode** Property.

Value	Meaning
OPOS_SUCCESS(0)	Compare firmware successful.
OPOS_E_BUSY(113)	Specified argument is invalid as the firmware file name.
Other Values	Refer to **ResultCode**.

DirectIO Method

Syntax

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

Remarks

In this OPOS, synchronous/asynchronous transmission of binary character string, and conversional synchronous/asynchronous transmission of hexadecimal character string are supported.

Synchronous transmission of binary character string

Command	OPOS_FIT_DIO_BIN_SYNC(0) OPOS_FIT_DIO_BIN_IMMIDIATE(1) OPOS_FIT_DIO_BIN_REALTIME(2)
pData	Not in use
pString	IN Binary character string
Function	It synchronously transmits binary character string to POS Printer, not influenced by present **BinaryConvertion** (Equivalent to OPOS_BC_NONE transmission). With this command, it becomes possible to directly transmit command to POS Printer Binary Character String is the following format. &H00 - &Hff(&HFF) The following are set when to transmit Partial Cut Command (1BH 6DH) to the printer side. "&H1B&H6D" (Small letters in the alphabet except "&H" are acceptable) It ignores the characters not admittable as HEX. Example: When "ABCD&H00EFG" is transmitted, only "0x00" is transmitted to Printer side.

Conversional synchronous transmission of hexadecimal character string

Command	OPOS_FIT_DIO_HEX_SYNC(5)
pData	Not yet used
pString	IN hexadecimal character string
Function	It synchronously transmits hexadecimal character string to POS Printer, not influenced by present **BinaryConvertion** (Equivalent to OPOS_BC_NONE transmission). With this command, it becomes possible to directly transmit command to POS Printer One byte character is displayed in 2 byte character. &H00 -> '00' Characters ranging from '0' – '9', 'A' – 'F', and 'a' – 'f' are available. If there is any character except the above, it returns OPOS_E_ILLEGAL1(106) without transmitting the command.

Asynchronous transmission of binary character string

Command	OPOS_FIT_DIO_BIN_ASYNC(6)
pData	Not yet used
pString	IN Binary character string
Function	It asynchronously transmits binary character string to POS Printer, not influenced by present **BinaryConversion** (Equivalent to OPOS_BC_NONE transmission). With this command, it becomes possible to directly transmit command to POS Printer. After execution **OutputCompleteEvent** is up. Binary character string is just like the following format. &H00 - &Hff(&HFF) The following are set when to transmit Partial Cut Command (1BH 6DH) to the printer side. "&H1B&H6D" (Small letters in the alphabet except "&H" are acceptable) It ignores characters not admittable as HEX. Example: When "ABCD&H00EFG" is transmitted, only "0x00" is transmitted to Printer side.

Conversional asynchronous transmission of hexadecimal character string

Command	OPOS_FIT_DIO_HEX_ASYNC(7)
pData	Not yet used
pString	IN hexadecimal character string
Function	It asynchronously transmits hexadecimal character string to POS Printer, not influenced by present **BinaryConvertion** (Equivalent to OPOS_BC_NONE transmission). With this command, it becomes possible to directly transmit command to POS Printer. After execution **OutputCompleteEvent** is up. One byte character is displayed in 2 byte character. &H00 -> '00' Characters ranging from '0' - '9', 'A' - 'F', and 'a' - 'f' are available. If there is any character except the above, it returns OPOS_E_ILLEGAL1(106) without transmitting the command.

Setting of error correction level of QR code

Command	OPOS_FIT_DIO_SET_QRERRORLV(20)
pData	IN error correction level of QR code
pString	Not yet used
Function	Specifies error correction level of QR code. The default value is “0”. The default value is set when **DeviceEnabled**=True is executed.

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

Getting of error correction level of QR code

Command	OPOS_FIT_DIO_GET_QRERRORLV(21)
pData	OUT error correction level of QR code
pString	Not yet used
Function	The error correction level of the QR code is returned. Error correction level (0-3) of the QR code is set to pData.

Setting of error correction level of Micro QR code

Command	OPOS_FIT_DIO_SET_MICROQRERRORLV(22)
pData	IN error correction level of Micro QR code
pString	Not yet used
Function	Specifies error correction level of Micro QR code. The default value is “0”. The default value is set when **DeviceEnabled**=True is executed.

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

Getting of error correction level of Micro QR code

Command	OPOS_FIT_DIO_GET_MICROQRERRORLV(23)
pData	OUT error correction level of Micro QR code
pString	Not yet used
Function	The error correction level of the Micro QR code is returned. Error correction level (0-2) of the Micro QR code is set to pData.

For all the calling after Enable, if these values except command are set to ***Command***, it returns OPOS_E_ILLEGAL(106) And **DirectIO** Method is buffered in **TransactionPrint** and **RotatePrint.** In this case, as for synchronous/asynchronous transmission of **DirectIO** Method, its synchronous/asynchronous execution depends on synchronous/asynchronous execution of **TransactionPrint** Method and **RotatePrint** Method

Value	Meaning
OPOS_E_ILLEGAL(106)	This method is unavailable.
OPOS_E_NOHARDWARE(107)	POS Printer is OFF or OFFLINE.
OPOS_E_FAILURE(111)	OPOS Control is in error state. After releasing error state, execute it.
OPOS_E_BUSY(113)	It cannot be executed because it is outputting.
OPOS_E_EXTENDED(114)	**ResultCodeExtended** = OPOS_EPTR_COVER_OPEN(201): POS Printer cover is open. (Only when synchronous method is executed.) **ResultCodeExtended** = OPOS_EPTR_REC_EMPTY(203): Run out of receipt paper . (Only when synchronous method is executed.) **ResultCodeExtended** = OPOS_FIT_EPTR_BLACKMARK(10001): Black Mark Error occurs. (Only when synchronous method is executed.) **ResultCodeExtended** = OPOS_FIT_EPTR_FATAL (10003): Fatal error occurs. (Only when synchronous method is executed.) **ResultCodeExtended** = OPOS_FIT_EPTR_OVERHEAT (10006): Head Overheat Error occurs. (Only when synchronous method is executed.)
The others	Refer to the items of **ResultCode**.

Open Method

Syntax

LONG Open (BSTR *DeviceName*);

DeviceName Parameter assigns the name of the device which opens.

Set up the following values according to interface of the POS Printer connected to.

- Serial Interface "PT330-331SERPRT", "PT330-331SER2PRT"
- USB Interface "PT330-331USBPRT", "PT330-331USB2PRT"
- LAN Interface "PT330-331LANPRT_[IP Address]"

*In case of LAN interface model, enter IP address of the POS printer in [IP address] part, in the next part of "Device name". Setting Program (ref. 2. 6. Setting Program Usage) helps to find each enabled Device name.

Example: Open ("PT330-331LANPRT_192.168.1.1")

Remarks

It is called to open the device.

When the **Open** method is successful, the Common properties and other Specific-to-Class Properties are initialized.

Return Value

Following values are returned and contained to **ResultCode** Property.

Value	Meaning
OPOS_SUCCESS(0)	It succeeds in Open.
OPOS_E_NOSERVICE(104)	It cannot be connected to corresponding Service Object.
OPOS_E_ILLEGAL(106)	The applicable Control has already opened.
OPOS_E_NOEXIST(109)	It cannot find assigned DeviceName.
OPOS_E_FAILER(111)	It failed to initialize OPOS. This error occurs when OCX cannot find SxJcp32.dll. When Registry has Apartment = "0" setting, and also another process is using Open Method for Printer with LAN interface connection, same error will occur.

ReleaseDevice Method

Syntax

LONG ReleaseDevice ();

Remarks

Call this device when to release exclusive access of Device.

If **DeviceEnabled** Property is **TRUE** and exclusive device, Device is made to disable. Don't execute at the time of Event Processing. (within Event Handler)

Return Value

One of the following values will be returned and contained to **ResultCode** Property.

Value	Meaning
OPOS_SUCCESS(0)	Exclusive Access is released. **Claimed** Property become **FALSE.**
OPOS_E_ILLEGAL(106)	Application doesn't have exclusive access right to applicable Device.
OPOS_E_BUSY(113)	Asynchronous processing is under way.
The others	Refer to the explanation of **ResultCode** Property.

ResetStatistics Method

Syntax

LONG ResetStatistics (BSTR *StatisticsBuffer***);**

Remarks

This method is not supported.

Return Value

Value	Meaning
OPOS_E_ILLEGAL(106)	This method is not supported.

RetrieveStatistics Method

Syntax

LONG RetrieveStatistics (BSTR* *pStatisticsBuffer***);**

Remarks

This method is not supported.

Return Value

Value	Meaning
OPOS_E_ILLEGAL(106)	This method is not supported.

UpdateFirmwareMethod

Syntax

LONG UpdateFirmware (BSTR *FirmwareFileName***);**

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

Remarks

This method updates the firmware of a device with the version of the firmware contained or defined in the file specified by the *FirmwareFileName* parameter regardless of whether that firmware's version is newer than, older than, or the same as the version of the firmware already in the device..

When this method is invoked, the Service Object should check that the specified firmware file exists. If so, this method should return immediately and the remainder of the update firmware process should continue asynchronously. The Service Object should notify the application of the status of the update firmware process by firing **StatusUpdateEvents** with values of OPOS_SUE_UF_PROGRESS(2100) + an integer between 1 and 100 indicating the completion percentage of the update firmware process. For application convenience, the **StatusUpdateEvent** value OPOS_SUE_UF_COMPLETE(2200) is defined to be the same value as OPOS_SUE_UF_PROGRESS(2100) + 100. If an error is detected during the asynchronous portion of an update firmware process, one of the following **StatusUpdateEvents** will be fired:

After downloading the firmware to the POS printer, when the firmware version acquired from the file name and the version acquired from the POS printer are compared (same processing as the **CompareFirmware** method). If inconsistency is found,

OPOS_SUE_UF_FAILED_DEV_OK(2201) is notified instead of

OPOS_SUE_UF_COMPLETE(2200).

Return Value

Value	Meaning
OPOS_SUE_UF_FAILED_DEV_OK(2201)	The update firmware process failed but the device is still operational.

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

Value	Meaning
OPOS_SUCCESS(0)	The method is executed successfully.
OPOS_E_NOEXIST(109)	The file specified by *FirmwareFileName* does not exist
OPOS_E_EXTENDED(114)	**ResultCodeExtended**= POS_EFIRMWARE_BAD_FILE(281): The specified firmware file or files exist, but one or more are either not in the correct format or are corrupt. (When the extension is other than "hx5", this error occurs)

UpdateStatistics Method

Syntax

LONG UpdateStatistics (BSTR *StatisticsBuffer***);**

Remarks

This method is not supported.

Return Value

Value	Meaning
OPOS_E_ILLEGAL(106)	This method is not supported.

4.5. Specific Property

AsyncMode Property R/W

Syntax

BOOL AsyncMode;

Remarks

TRUE: Printing methods of **PrintNormal, CutPaper, PrintBarCode, PrintBitmap, RotatePrint, TransactionPrint** are executed asynchronously.

FALSE: Method is executed synchronously.

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

CapCharacterSet Property

Syntax

LONG CapCharacterSet;

Remarks

It shows printable character setting of POS printer.

This property has one of the following values:

Value	Meaning
PTR_CCS_KANJI(11)	Character setting supports cord page 932. It supports single-byte katakana's between 0xA1 and 0xDF, and all the ASCII characters between 0x20 and 0x7F. Also, it supports the Sift JIS Code characters which are defined by the first JIS standard level and the second JIS standard level.

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

CapCoverSensor Property

Syntax

BOOL CapCoverSensor;

Remarks

TRUE: POSPrinter has "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 the **CharacterSetList** property.

CapRec2Color Property

Syntax

BOOL CapRec2Color;

Remarks

TRUE: It is possible to print receipts in two colors. (*If the printing color is set up "Mono" by Setting up Program, Setting-up will be **FALSE**: Impossible to print in two colors.)

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

CapRecBarCode Property

Syntax

BOOL CapRecBarCode;

Remarks

TRUE: It is possible to print bar-codes on receipts.

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

CapRecBitmap Property

Syntax

BOOL CapRecBitmap;

Remarks

TRUE: It is possible to print bitmap on receipts.

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

CapRecBold Property

Syntax

BOOL CapRecBold;

Remarks

TRUE: It is possible to have Bold attribute with receipts.

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

CapRecCartridgeSensor Property

Syntax

LONG CapRecCartridgeSensor:

Remarks

0: The feature of the receipt cartridge sensor is not supported.

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

CapRecColor Property

Syntax

LONG CapRecColor:

Remarks

0: The feature of the receipt color printing 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** method.

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

LONG CapRecItalic:

Remarks

FALSE: The receipt cannot print italic characters.

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

CapRecLeft90 Property

Syntax

BOOL CapRecLeft90;

Remarks

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

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

CapRecMarkFeed Property

Syntax

LONG CapRecMarkFeed:

Remarks

0: The feature of handling mark sensed paper is not supported.

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

CapRecNearEndSensor Property

Syntax

BOOL CapRecNearEndSensor;

Remarks

TRUE: The receipt has a low paper sensor.

FALSE: The low paper sensor does not work.

This property is initialized by the **Open** method If PNESense in the setting program is set to Enabled, it is initialized to **TRUE**, and if Disabled, it is initialized to **FALSE**.

CapRecPageMode Property

Syntax

BOOL LONG CapRecPageMode:

Remarks

FALSE: The printer is not capable of supporting Page Mode for the receipt station.

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

CapRecPapercut Property

Syntax

BOOL CapRecPapercut;

Remarks

TRUE: The receipt can perform paper cuts.

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

CapRecPresent Property

Syntax

BOOL CapRecPresent;

Remarks

TRUE: It is possible to print receipts.

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

CapRecRight90 Property

Syntax

BOOL CapRecRight90;

Remarks

TRUE: It is possible to have 90-Degree-Rotaion-to-the-Right attribute of receipts.

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

CapRecRotate180 Property

Syntax

BOOL CapRecRotate180;

Remarks

TRUE: It is possible to have 180-Degree-Rotaion attribute of receipts.

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

CapRecStamp Property

Syntax

BOOL CapRecStamp;

Remarks

FALSE: It is impossible to print stamp on receipts.

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

CapRecUnderline Property

Syntax

BOOL CapRecUnderline;

Remarks

TRUE: It is possible to have Underline attribute of receipts.

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

CapTransaction Property

Syntax

BOOL CapTransaction;

Remarks

TRUE: Batch processing of POS Printer is valid.

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

CartridgeNotify Property R/W

Syntax

LONG CartridgeNotify;

Remarks

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

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

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

Return Value

One of the following values is contained in **ResultCode** Property at the time of this property setting up.

Value	Meaning
OPOS_E_ILLEGAL(106)	This property cannot be set.
Other Values	Refer to **ResultCode.**

CharacterSet Property R/W

Syntax

LONG CharacterSet;

Remarks

It sets up the characters for printing.

This property is initialized when Device is first enabled after **Open** Method.

One of the following values is set up in this property.

Value	Meaning
101	It selects MIK Character Set.
102	It selects PC866(Cynillic #2) Character Set.
103	It selects Thai code 18.
437	It selects PC437 (USA: Standard Europe) Character Set.
850	It selects PC850 (Multilingual) Character Set.
851	It selects PC851 (Greece - obsolete) Character Set.
852	It selects PC852 (Latin2) Character Set.
857	It selects PC857 (Turkish) Character Set.
858	It selects PC858 (Euro) Character Set.
860	It selects PC860 (Portuguese) Character Set.
863	It selects PC863 (Canadian-French) Character Set.
864	It selects PC864 (Arabic without BOX DRAWINGS below 20) Character Set.
865	It selects PC865 (Nordic) Character Set.
866	It selects PC866 (Cyrillic #2) Character Set.
869	It selects PC869 (Greece) Character Set.
932	Windows Code Page; Japanese Version Shift-JIS.
PTR_CS_ASCII (998)	It set up ASCII Character. It supports ASCII Characters between 0x20 and 0x7F. The constant value is 998.
1252	It selects WPC1252 Character Set.
2859	It selects ISO8859-2 (1999 Latin Alphabet No.2) Character Set.
28597	It selects ISO8859-7 (1987 LatinGreek Alphabet) Character Set.

Return Value

One of the following values is contained in **ResultCode** Property at the time of this property setting up.

Value	Meaning
OPOS_SUCCESS(0)	It succeeds in setting up this property.
OPOS_E_ILLEGAL(106)	Illegal values were used.
The others	Refer to the items of **ResultCode.**

CharacterSetList Property

Syntax

BSTR CharacterSetList;

Remarks

It is Character String of Character Setting up Number. In Japanese version, "101,102,103,437,850,851,852,857,858,860,863,864,865,866,869,932,998,1252,28592,28597" are set up. One of the values is set by Installer. This property is initialized by the **Open** method.

CoverOpen Property

Syntax

BOOL CoverOpen;

Remarks

TRUE: POS Printer Cover is open.

FALSE: POS Printer Cover is closed.

This property is initialized while Device is made to enable, and keeps the present state.

ErrorLevel Property

Syntax

LONG ErrorLevel;

Remarks

It shows seriousness of error condition.

One of the following values is set up in this property.

Value	Meaning
PTR_EL_NONE(1)	It is not error state.
PTR_EL_RECOVERABLE(2)	Recoverable error occurs. (at the time of Cover Open Error, Receipt End Error, Head Hot Error, Black Mark Error or Power Discontinuity Error)
PTR_EL_FATAL(3)	Unrecoverable Error happens. (only in case of fatal error)

This property is set up by Control before **ErrorEvent** is notified. If the error is deleted, this property converts into PTR_EL_NONE(1).

ErrorStation Property

Syntax

LONG ErrorStation;

Remarks

Holds the POS printer (PTR_S_RECEIPT(2)) in printing when an error is detected.

This property is set up before **ErrorEvent** is notified.

When the power is turned off (or the cable is disconnected), "0" is set.

ErrorString Property

Syntax

BSTR ErrorString;

Remarks

It keeps particular-to-vender description of present error.

This property is set up by Control before **ErrorEvent** is notified. If this description is not used, null character string is set up in property. When the error is deleted, this property is converted to null character string.

The following wordings are set up by POS Printer.

- When the cover opens. "Cover Open"
- When receipt paper ended. "Paper End"
- When the head is hot. "Head Hot"
- When fatal error occurs "Fetal Error"
- When Cutter jam Error occurs "Cutter Jam Error"
- Power Discontinuity (Offline) "Power Off or Offline"

FlagWhenIdle Property R/W

Syntax

BOOL FlagWhenIdle;

Remarks

TRUE: If POS Printer Control is in idle state, it notifies **StatusUpdateEvent**.

FALSE: This event is not notified.

If this status event is notified, **FlagWhenIdle** is automatically reset to **FALSE.**

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

If, **State** Property is already OPOS_S_IDLE(2) and when **FlagWhenIdle** Property is set up to **TRUE, StatusUpdateEvent** is immediately notified. Accordingly, Application can use this event without worrying about the time difference between asynchronous output end and setting up of this flag.

This property is initialized to **FALSE** by **Open** Method.

Following value is contained in **ResultCode** Property at the time of this property setting up.

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

FontTypefaceList Property

Syntax

BOOL FontTypefaceList;

Remarks

An empty string is set. It indicates that only the default typeface is supported.

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

MapCharacterSet Property

Syntax

BOOL MapCharacterSet;

Remarks

TRUE: On outputting data, the Service Object maps the characters transferred by the application to the character set selected in the **CharacterSet** property for printing data.

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

MapMode Property R/W

Syntax

LONG MapMode;

Remarks

It shows mapping mode of the Printer. It defines measuring units used by other properties, such as line heights and line spacing.

It supports the following map modes. The values inside the parentheses are the values calculated in dots per each unit.

Value	Meaning
PTR_MM_DOTS(1)	Width of POS Printer dot 0.125mm (1 dot)
PTR_MM_TWIPS(2)	1/1440 of one inch (7.0866 dot)
PTR_MM_ENGLISH(3)	0.001 inch (4.921 dot)
PTR_MM_METRIC(4)	0.01 mm (12.5 dot)

RecLineSpacing, RecLineWidth, and **RecLineHeight** change, too, if you set up **MapMode**.

At the first enablement after **Open** Method, It is initialized to PTR_MM_DOTS(1).

Return Value

Following values are contained in **ResultCode** Property at the time of this property setting up.

Value	Property
OPOS_SUCCESS(0)	The property was set successfully.
OPOS_E_ILLEGAL(106)	Illegal mapping mode is assigned.

PageModeArea Property

Syntax

BSTR PageModeArea;

Remarks

This property is not supported by this OCX.

PageModeDescriptor Property

Syntax

LONG PageModeDescriptor;

Remarks

This property is not supported by this OCX.

PageModeHorizontalPosition Property

Syntax

LONG PageModeHorizontalPosition;

Remarks

This property is not supported by this OCX.

PageModePrintArea Property

Syntax

BSTR PageModePrintArea;

Remarks

This property is not supported by this OCX.

PageModeStation Property

Syntax

LONG PageModeStation;

Remarks

This property is not supported by this OCX.

PageModeVerticalPosition Property

Syntax

LONG PageModeVerticalPosition;

Remarks

This property is not supported by this OCX.

RecBarCodeRotationList Property

Syntax

BSTR RecBarCodeRotationList;

Remarks

This character string shows the possible direction of receipt bar-code rotation.

"0, R90, L90, 180" are set up.

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

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

RecBitmapRotationList Property

Syntax

BSTR RecBitmapRotationList;

Remarks

This character string shows the directions in which a receipt bitmap may be rotated..

"0, R90, L90, 180" can be set.

This property is initialized by the **Open** method. The string consists of rotation strings separated by commas. The 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

Indicates the status of the currently selected Receipt cartridge (ink, ribbon or toner).

Since the POS printer is the thermal printer, this is fixed to the following value.

Value	Meaning
PTR_CART_UNKNOWN(268435456)	The device does not support the feature of notifying the cartridge state.

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

RecCurrentCartridge Property R/W

Syntax

LONG RecCurrentCartridgeState;

Remarks

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

Return Values

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

Value	Meaning
OPOS_E_ILLEGAL(106)	Specifying cartridge state is invalid.
Other Values	Refer to the **ResultCode** section.

RecEmpty Property

Syntax

BOOL RecEmpty;

Remarks

TRUE: Receipt paper is run out

FALSE: Enough receipt paper

This property is initialized while Device is enabled, and keeps present condition

RecLetterQuality Property R/W

Syntax

BOOL RecLetterQuality;

Remarks

TRUE: It prints in high quality printing mode.

FALSE: It prints in normal printing mode.

This property is initialized to **TRUE,** at the time of initial enablement of Device after **Open** Method.

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

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

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

In case of bitmap printing in Escape Sequence, in high quality printing mode, printing is in normal resolution, and in normal printing mode, printing is in.1/2 resolution. (double-width and double-height) The method follows **RecLetterQuality** in the same way. (*When Registry is set up, if Smoothing setting is off, it does not do smoothing processing even at the time of setting up to **TRUE**.)

Return Value

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

It is the number of the half-size characters, per line of receipts.

Printing is done in the following font, according to the assigned number of half-size characters per line of receipt.

Characters per Line	Printing Font (WidthxHeight)
Printer Setting: Paper width 80 mm (576 dots)	
48 (double-width 24)	12x24 dots (font A)
57 (double-width 28)	10x24 dots (font B)
72 (double-width 36)	8x16 dots (font C)
Printer Setting: Paper width 80 mm (512 dots)	
42 (double-width 21)	12x24 dots (font A)
51 (double-width 25)	10x24 dots (font B)
64 (double-width 32)	8x16 dots (font C)
Printer Setting: Paper width 58 mm (420 dots)	
35 (double-width 17)	12x24 dots (font A)
42 (double-width 21)	10x24 dots (font B)
52 (double-width 26)	8x16 dots (font C)
Printer Setting: Paper width 58 mm (384 dots)	
32 (double-width 16)	12x24 dots (font A)
38 (double-width 19)	10x24 dots (font B)
48 (double-width 24)	8x16 dots (font C)

If this value changed into supported line character width, the character width is set up to the assigned value. If it cannot support exact width, it is set up to the nearest value in supported line width and at the same time larger value than supported line width. (For example, when to set up paper width to 80 mm and to set 40 for Printer, Service Object will select character size of "48".) If it cannot support character width, Error will return.

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

Return Value

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)	Illegal line character width is assigned.

RecLineCharsList Property

Syntax

BSTR RecLineCharsList;

Remarks

It is character string which includes line character width supported by receipt.

This property is initialized to the following values by **Open** Method according to Printer paper width and setting.

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

RecLineHeight Property R/W

Syntax

LONG RecLineHeight;

Remarks

It is the receipt print line height. It is written in the unit defined by **MapMode**.

If **RecLineChars** is converted, **RecLineHeight** will be updated to default line height of the selected width.

The value of **RecLineHeight** will be initialized to the default line height of POS Printer by **Open** Method.

The following are the applicable values. (*As for the value of Property, the value of **MapMode** Property is PTR_MM_DOTS(1))

Characters per Line	Value of RecLineHight property
Printer Setting: Paper width 80 mm (576 dots)	
48	24
57	24
72	16
Printer Setting: Paper width 80 mm (512 dots)	
42	24
51	24
64	16
Printer Setting: Paper width 58 mm (420 dots)	
35	24
42	24
52	16
Printer Setting: Paper width 58 mm (384 dots)	
32	24
38	24
48	16

Return Value

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

Value	Meaning
OPOS_E_ILLEGAL(106)	This property setting is not possible. It can be only acquired.

RecLineSpacing Property R/W

Syntax

LONG RecLineSpacing;

Remarks

It is the spacing of each single-high print line, including both the printed line height plus the white space between each pair of lines. This property is written by the unit defined by **MapMode**.

When **RecLineChars** is converted, if the new **RecLineHeight** is larger than the value assigned by **RecLineSpacing,** the same value as that of **RecLineHeight** will be set up.

The value of **RecLineSpacing** will be initialized to the default line space of POS Printer after **Open** Method.

The setting is possible between **16(dot)** and **127(dot)**.

Return Value

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)	Property setting range is illegal.
The others	Refer to **PrintNormal** Method.

RecLinesToPaperCut Property

Syntax

LONG RecLinesToPaperCut;

Remarks

It holds the number of lines that must be advanced before the receipt paper is cut. These lines are ones before reaching the paper cut mechanism.

This property is updated by the change of **RecLineChars** Property or **RecLineSpacing** Property.

RecLineWidth Property

Syntax

LONG RecLineWidth;

Remarks

It is the width of a line of **RecLineChars** and is written in the unit defined by **MapMode**.

This property is initialized after **Open** Method.

The following values are set up according to printer paper width.

Printer Paper Width	Value
Paper width 80 mm	576
Paper width 80 mm	512
Paper width 58 mm	420
Paper width 58 mm	384

RecNearEnd Property

Syntax

BOOL RecNearEnd;

Remarks

TRUE: Receipt paper is low.

FALSE: Receipt paper is not low.

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

RecSidewaysMaxChars Property

Syntax

LONG RecSidewaysMaxChars;

Remarks

In case of Sideways Mode (Printing with 90 degree rotation to the left or to the right), it is the maximum number of the half-size characters per line. Since the width of 90 degrees rotating to the left or to the right is declined to half when the CapRec2Color property is TRUE (2 colors), the number of printable characters is half of the value.

The following values are taken.

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

RecSidewaysMaxlLines Property

Syntax

LONG RecSidewaysMaxLines;

Remarks

In case of Sideways Mode (Printing with 90 degree rotation to the left or to the right), it is the maximum number of the lines printed.

It is the value obtained by dividing **RecLineWidth** Property by **RecLineSpacing** Property. However, if the remainder of the divided value is equal to, or larger than **RecLineHeight** Property (font height), the value will be the one equal to the divided value plus +1. Accordingly, the property will change by changing **RecLineSpacing** Property.

However, exceptionally in case of fontC (*Refer to **RecLineChars** Property), calculation is done to the value **(RecLineWidth** – 7(dot)), against the above condition.

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

RotateSpecial Property R/W

Syntax

LONG RotateSpecial;

Remarks

It shows the bar code rotation direction.

This property is initialized to PTR_RP_NORMAL(1) with **Open** Method.

This property has one of the following values:

Value	Meaning
PTR_RP_NORMAL(1)	Bar code can be printed to the normal direction.
PTR_RP_RIGHT90(257)	Bar code can be printed with the 90 degree rotation to the right.
PTR_RP_LEFT90(258)	Bar code can be printed with the 90 degree rotation to the left.
PTR_RP_ROTATE180(259)	Bar code can be printed with the 180 degree rotation (upside-down).

Return Value

One of the following values is contained in **ResultCode** Property at the time of this property setting up.

Value	Meaning
OPOS_SUCCESS(0)	The property was set successfully.
OPOS_E_ILLEGAL(106)	Illegal property value is assigned.

*The following specific POS printer properties are not supported.

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

4.6. Exclusive-Use Methods

BeginInsertion Method

Syntax

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

Remarks

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

Return Value

One of the following values are returned and placed in the **ResultCode** property:

Value	Meaning
OPOS_E_ILLEGAL(106)	The POS printer does not have the slip.
Other Values	Refer to the **ResultCode** section.

BeginRemoval Method

Syntax

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

Remarks

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

Return Value

One of the following values are returned and placed in the **ResultCode** property:

Value	Meaning
OPOS_E_ILLEGAL(106)	The POS printer does not have the slip.
Other Values	Refer to the **ResultCode** section.

ChangePrintSide Method

Syntax

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

Remarks

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

Return Value

One of the following values are returned and placed in the **ResultCode** property:

Value	Meaning
OPOS_E_ILLEGAL(106)	The POS printer does not have the slip.
Other Values	Refer to the **ResultCode** section.

ClearPrintArea Method

Syntax

LONG ClearPrintArea ();

Remarks

This method is not supported by this OCX. OPOS_E_ILLEGAL(106) is returned.

CutPaper Method

Syntax

LONG CutPaper (LONG *Percentage***);**

Percentage Parameter indicates the percentage of the paper to be cut. When the value is between '1' to '99', partial cutting is performed. When the value is '100', full cutting is performed.

When the value is other than '1' to '100', OPOS_E_ILLEGAL(106) is returned.

Remarks

This method is called when to cut receipt paper.

This method is executed synchronously if **AsyncMode** is **FALSE** and asynchronously if **AsyncMode** is **TRUE.** When **PrintNormal** Method or **PrintImmediate** Method is called, paper cutting can be done using Escape Sequence of paper cutting, too. In addition to that, if POS Printer has buffered data (even though printing is requested, POS Printer does not print), it cannot cut paper. In order to cut receipt paper, it must be at the head of each line.

Return Value

One of the following values is contained in **ResultCode** Property at the time of this property setting up.

Value	Meaning
OPOS_SUCCESS(0)	Method ends properly.
OPOS_E_ILLEGAL(106)	Illegal percentage is assigned.
OPOS_E_NOHARDWARE(107)	POS Printer is OFF or OFFLINE.
OPOS_E_FAILURE(111)	OPOS Control is in error state. Execute after deleting error state.
OPOS_E_BUSY(113)	It cannot execute because it is outputting.
OPOS_E_EXTENDED(114)	**ResultCodeExtended** = OPOS_EPTR_COVER_OPEN(201): POS Printer cover is open . (It returns only when **AsyncMode** is **FALSE**.) **ResultCodeExtended** = OPOS_EPTR_REC_EMPTY(203): It runs out of receipt paper. (It returns only when **AsyncMode** is **FALSE**.) **ResultCodeExtended** = OPOS_FIT_EPTR_BLACKMARK(10001): Black Mark error occurs. (It returns only when **AsyncMode** is **FALSE**.) **ResultCodeExtended** = OPOS_FIT_EPTR_FATAL (10003): Fatal error occurs. (It returns only when **AsyncMode** is **FALSE**.) **ResultCodeExtended** = OPOS_FIT_EPTR_OVERHEAT (10006): Head Heat up error occurs. (It returns only when **AsyncMode** is **FALSE**.)
The others	Refer to the items of **ResultCode**.

EndInsertion Method

Syntax

LONG EndInsertion ();

Remarks

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

Return Value

One of the following values are returned and placed in the **ResultCode** property:

Value	Meaning
OPOS_E_ILLEGAL(106)	The POS printer does not have the slip.
Other Values	Refer to the **ResultCode** section.

EndRemoval Method

Syntax

LONG EndRemoval ();

Remarks

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

Return Value

One of the following values are returned and placed in the **ResultCode** property:

Value	Meaning
OPOS_E_ILLEGAL(106)	The POS printer does not have the slip.
Other Values	Refer to the **ResultCode** section.

MarkFeed Method

Syntax

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

Remarks

This method is not supported by this OCX.

Return Value

One of the following values are returned and placed in the **ResultCode** property:

Value	Meaning
OPOS_E_ILLEGAL(106)	The feature of handling mark sensed paper is not supported. (Refer to the **CapRecMarkFeed** Property.)
Other Values	Refer to the **ResultCode** section.

PageModePrint Method

Syntax

LONG PageModePrint (LONG Control);

Remarks

This method is not supported by this OCX. OPOS_E_ILLEGAL(106) is returned.

PrintBarCode Method

Syntax

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

Parameter	Description
Station	PTR_S_RECEIPT(2) is assigned.
Data	Bar-code character string format of the data depend on **BinaryConversion** Property value. Refer to **BinaryConversion** Property for further details.
Symbology	Usable bar-code type (Refer to the values below)
Height	Bar code height, which is written in the unit defined by **MapMode** It is possible to set from 1 to 255dot. In case of PDF417 printing with upright/upside-down, it is possible to set from 1 to 831(dot). Or, in case of 90 degree rotation to the right or to the left, it is possible to set from 12 to the value of **RecLineWidth** Property (dot). Specify the range of 1-16 for width of the module for the QR code and micro QR code of two dimension bar code. OPOS_E_ILLGAL(106) is notified when other values are specified.
Width	Bar code width, which is written in the unit defined by **MapMode**.
Alignment	Bar code location Refer to the value below.
TextPosition	Position of character strings. Refer to the value below.

Values of the *Symbology* Parameter of this release are just the following

Value	Label type
PTR_BCS_UPCA(101)	UPC-A
PTR_BCS_UPCE(102)	UPC-E
PTR_BCS_EAN8(103)	EAN 8 (= JAN 8)
PTR_BCS_JAN8(103)	JAN 8 (= EAN 8)
PTR_BCS_EAN13(104)	EAN 13 (= JAN 13)
PTR_BCS_JAN13(104)	JAN 13 (= EAN 13)
PTR_BCS_ITF(106)	Interleaved 2 of 5
PTR_BCS_Codabar(107)	Codabar(NW-7)
PTR_BCS_Code39(108)	Code 39
PTR_BCS_Code93(109)	Code 93
PTR_BCS_Code128(110)	Code 128
PTR_BCS_PDF417(201)	PDF 417
PTR_BCS_QRCODE(204)	QR code (two dimension barcode)
PTR_BCS_UQRCODE(205)	Micro QR code (two dimension barcode)

The values of *Alignment* Parameter are just the following.

Value	Meaning
PTR_BC_LEFT(-1)	Align with the left-most print column. (Because it is executed to Printing data, it is to align right toward POS Printer Printing direction in case of 180 degree rotation printing.)
PTR_BC_CENTER(-2)	Align in the center of the station. In the case of two dimension barcode, this parameter is not supported in 90 degrees rotating to the left or to the right. it works as PTR_BC_LEFT(-1).
PTR_BC_RIGHT(-3)	Align with the right-most print column. (Because it is executed to Printing data, it is to align left toward POS Printer Printing direction in case of 180 degree rotation printing.) In the case of two dimension barcode, this parameter is not supported in 90 degrees rotating to the left or to the right. It works as PTR_BC_LEFT(-1).
The others	Distance from the left-most print column to the start of the bar code. It is written in the unit defined by **MapMode**. OPOS_E_ILLEGAL(106) will return, in the case that actual bar code printing width, calculated by OPOS from the bar code width assigned by *Width* Parameter plus the distance from the left-most print column exceeds **RecLineWidth** Property value. However, in the case that PTR_RP_RIGHT90(257), PTR_RP_LEFT90(258) are assigned in **RoatateSpecial** Property, PTR_BC_LEFT(-1) is considered to be assigned and printing is done. In the case of PDF417, this parameter is not supported in 90 degrees rotating to the left or to the right. it works

as PTR_BC_LEFT(-1).

TextPosition Parameter values are just the following.

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

Remarks

This method is called when to print bar codes with assigned POS Printer.

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

Following are printable bar code conditions per each *Symbology.*

<MONO>

Symbology	Each printable character kind	Upright/Upside-down mode		90 degree rotation to the left/right	
		Character string length	Width (dots)	Character string length	Width (dots)
PTR_BCS_UPCA *1	10 kinds ('0'-'9')	11-12	95-RecLineWidthValue	11-12	95-1662
PTR_BCS_UPCE *1		11-12	51-RecLineWidthValue	11-12	51-1662
PTR_BCS_JAN8 *1		7-8	67-RecLineWidthValue	7-8	67-1662
PTR_BCS_JAN13 *1		12-13	95-RecLineWidthValue	12-13	95-1662
PTR_BCS_CODE39 *1	43 kinds ('0'-'9', 'A'-'Z', space, '$', '%', '+', '-', '.', '/') (Start/Stop character of '*' is automatically added.)	1-34	47-RecLineWidthValue	1-101	47-1662
PTR_BCS_ITF *1	10 kinds ('0'-'9')	2-62	27-RecLineWidthValue	2-182	27-1662
PTR_BCS_CODABAR *1	20 kinds ('0'-'9', 'A'-'D', '$', '+', '-', '.', '/', ':')	3-47	41-RecLineWidthValue	3-138	41-1662
PTR_BCS_CODE93 *1	128 kinds (0x00-0x7F) (Lower stage is for two characters)	1-59 1-29	46-RecLineWidthValue	1-88 1-44	46-1662
PTR_BCS_CODE128 *1	Code Set A: 0x00 - 0x5F Code Set B 0x20 - 0x7F Code Set C 0x00 - 0x63 However the characters including "{" are exception. For details refer to later.	3-51	46-RecLineWidthValue	3-74	46-1662

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

*1 When the setup value of Width is minimum value (exclude "0") or less, the following barcodes are printed by a recommended dot width.

PTR_BCS_UPCA, PTR_BCS_UPCE, PTR_BCS_JAN8, PTR_BCS_JAN13, PTR_BCS_CODE39, PTR_BCS_ITF, PTR_BCS_CODABAR, PTR_BCS_CODE93, PTR_BCS_CODE128

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

Symbology	Formula for calculating printing width
PTR_BCS_UPCA	Bar code width = 95 * dotNarrow
PTR_BCS_UPCE	Bar code width = 51 * dotNarrow
PTR_BCS_JAN8	Bar code width = 67 * dotNarrow
PTR_BCS_JAN13	Bar code width = 95 * dotNarrow
PTR_BCS_CODE39	Bar code width = 6 * dotNarrow + 3 * dotWide + 1 * dotNarrow + (6 * dotNarrow + 3 * dotWide + 1 dotNarrow)* Length + 6 * dotNarrow + 3 * dotWide (Length = Number of characters printed)
PTR_BCS_ITF	Bar code width = 4 * dotNarrow + (3 * dotNarrow + 2 * dotWide) * Length + 2 * dotNarrow + 1 * dotWide (Length = Number of characters printed)
PTR_BCS_CODABAR	Bar code width = (5 * dotNarrow + 2 * dotWide) * (Length – Wlen) + (4 * dotNarrow + 3 * dotWide) * Wlen + 1 * dotNarrow * (Length + 1) (Length = Number of characters printed) (Wlen = Number of characters among ":", "/", ".", "+", "A", "B", "C", "D")

PTR_BCS_CODE93	Bar code width = 9 * dotNarrow + 9 * dotNarrow * Wlen + (9 * dotNarrow) * 2 * (Length - Wlen) + 9 * 2 * dotNarrow + 10 * dotNarrow (Length = Number of characters printed) (Wlen =Number of characters among "0"-"9", "A"-"Z", " ", "$", "%", "+", "-", "/")
PTR_BCS_CODE128	Bar code width = 11 * dotNarrow * (Length + 1) + 13 * dotNarrow (Length = Number of characters printed – Special characters*[1]) *1: Number of characters which head is "{". If "{AA{BA" is assigned, Length = 6 – 2 = 4.
PTR_BCS_PDF417	Bar code width = ((17 * (C + 2)) + (17 + 18)) * X Bar code height = RYX C: Number of columns X: Nominal width of narrow element R: Number of rows Y: Row height * For number of rows and number of columns, the minimum value that input data can convert as the bar code is selected. For nominal width of narrow element and row height, after number of rows and number of columns are determined, maximum size that does not exceed the Width and Height parameters is selected.
PTR_BCS_QRCODE	Bar code width = Number of cells of vertical directions * Module width (1 – 16) Bar code height = Horizontal number of cells * Module width (1 – 16) * Width and height of the bar code are set to the maximum size that doesn't exceed the value of Width.

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

*Relation between dotNarrow and dotWide

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

Notes for Bar Code Printing

1. When to print CODE39, "*" (Start/Stop Character) is automatically added. So, there is no deed of setting up in Character.

2. When to assign ITF, even-number character must be assigned. If odd-number is assigned, OPOS_E_ILLEGAL(106) will return.

3. When to assign CODABAR, the head and the tail of the characters must be among "A" - "D". Accordingly, three or more than three characters (the head character plus any characters plus the tail character) must be assigned. In the other cases, OPOS_E_ILLEGAL(106) returns.

4. When to assign UPC-E, development is done according to the following list. UPC-A Left Code shows top characters (2-6), UPC-A Right Code shows 7th-11th characters. The shortened code is actually printed as UPC-E. If the UPC-A top character assigned is except 0 or, characters not based on the following list is assigned, OPOS_E_ILLEGAL(106) returns

   Example 05810000226 -> Converted to c58226.

   09859363583 -> OPOS_E_ILLEGAL returns.

Manufacturer Code Left Code for UPC-A					**Item Code Right Code for UPC-A**					**Shortened Code**					
F1	**F2**	**F3**	**F4**	**F5**	**A1**	**A2**	**A3**	**A4**	**A5**	**Z1**	**Z2**	**Z3**	**Z4**	**Z5**	**Z6**
0-9	0-9	0	0	0	0	0	0-9	0-9	0-9	F1	F2	A3	A4	A5	0
0-9	0-9	1	0	0	0	0	0-9	0-9	0-9	F1	F2	A3	A4	A5	1
0-9	0-9	2	0	0	0	0	0-9	0-9	0-9	F1	F2	A3	A4	A5	2
0-9	0-9	3-9	0	0	0	0	0	0-9	0-9	F1	F2	F3	A4	A5	3
0-9	0-9	0-9	1-9	0	0	0	0	0	0-9	F1	F2	F3	F4	A5	4
0-9	0-9	0-9	0-9	1-9	0	0	0	0	5-9	F1	F2	F3	F4	F5	A5

5. When to print CODE128, set up characters as followed.

   1. One of "{A", "{B", "{C" must be assigned as the head of the bar code. Following that, each of CODE A, CODE B, CODE C must be set up.
   2. When to assign Function Code, assign "{1", "{2", "{3", or "{4". Each is to assign FNC1, FNC2, FNC3, or FNC4. For further information, in CODE C, only FUNC1is available. If you assign except FUNC1 in CODE C, OPOS_E_ILLEGAL(106) returns.
   3. When to print "{" in CODE B, assign "{{".
   4. When to set up SHIFT, assign "{S". After that, code set of one character sifts like CODE A <- -> CODE B. If you assign in CODE C, OPOS_E_ILLEGAL(106) returns.

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

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)

Character to Print			Character to Print		
CODE-A	CODE-B	CODE-C	CODE-A	CODE-B	CODE-C
>	>	30(1EH)	DC3(13H)	s	83(53H)
?	?	31(1FH)	DC4(14H)	t	84(54H)
@	@	32(20H)	NAK(15H)	u	85(55H)
A	A	33(21H)	SYN(16H)	v	86(56H)
B	B	34(22H)	ETB(17H)	w	87(57H)
C	C	35(23H)	CAN(18H)	x	88(58H)
D	D	36(24H)	EM(19H)	y	89(59H)
E	E	37(25H)	SUB(1AH)	z	90(5AH)
F	F	38(26H)	ESC(1BH)	{ "{{"	91(5BH)
G	G	39(27H)	FS(1CH)	|	92(5CH)
H	H	40(28H)	GS(1DH)	}	93(5DH)
I	I	41(29H)	RS(1EH)	~	94(5EH)
J	J	42(2AH)	US(1FH)	DEL	95(5FH)
K	K	43(2BH)			96(60H)
L	L	44(2CH)			97(61H)
M	M	45(2DH)			98(62H)
N	N	46(2EH)			99(63H)
O	O	47(2FH)	Following are used by assigning "{"		
P	P	48(30H)	FNC 3 "{3"	FNC 3 "{3"	
Q	Q	49(31H)	FNC 2 "{2"	FNC 2 "{2"	
R	R	50(32H)	SHIFT "{S"	SHIFT "{S"	
S	S	51(33H)	CODE C "{C"	CODE C "{C"	
T	T	52(34H)	CODE B "{B"	CODE A "{A"	CODE B "{B"
			FNC 4 "{4"	FNC 4 "{4"	CODE A "{A"
			FNC 1 "{1"	FNC 1 "{1"	FNC 1 "{1"

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

Symbology	***TextPosition*** **PTR_BC_TEXT_NONE**	***TextPosition*** **PTR_BC_TEXT_ABOVE** **PTR_BC_TEXT_BELOW**
PTR_BCS_UPCA	Printable	Width=95 - 189 OPOS_E_ILLEGAL
PTR_BCS_UPCE	Printable	Width=51 - 101 OPOS_E_ILLEGAL
PTR_BCS_JAN8	Printable	Width=67 - 133 OPOS_E_ILLEGAL
PTR_BCS_JAN13	Printable	Width=95 - 189 OPOS_E_ILLEGAL
PTR_BCS_CODE39	Printable	Printable
PTR_BCS_ITF	Printable	Printable
PTR_BCS_Codabar	Printable	Printable
PTR_BCS_CODE93	* 1	* 1
PTR_BCS_CODE128	* 2	* 2

*1: Relation between *width* and character number which become OPOS_E_ILLEGAL is within the following range.

37 + 9 * wlen + 18(len - wlen) <= *width* < 74 + 18 * wlen + 36(len - wlen)

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

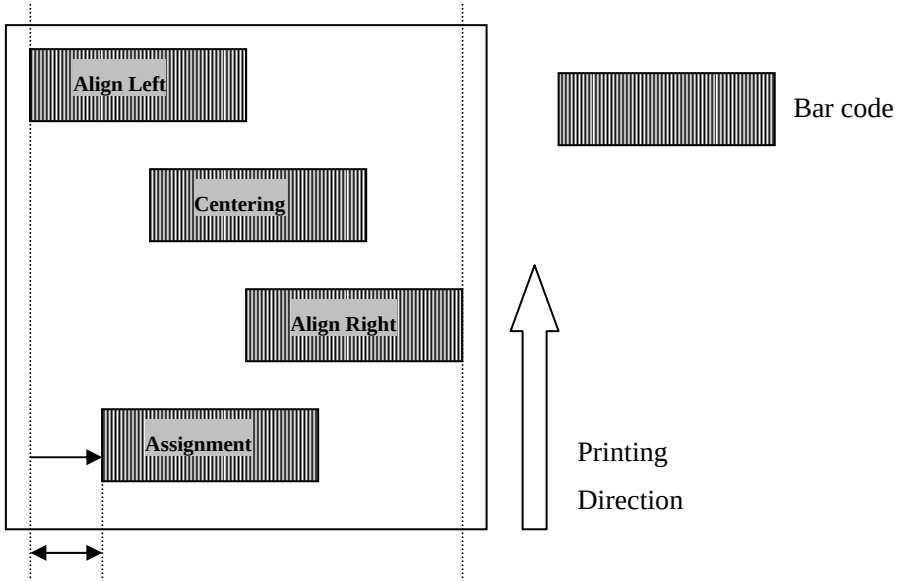
*2: Relation between width and character number which become OPOS_E_ILLEGAL is within the following range.

24 + 11 * len <= width < 48 + 22 * len

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

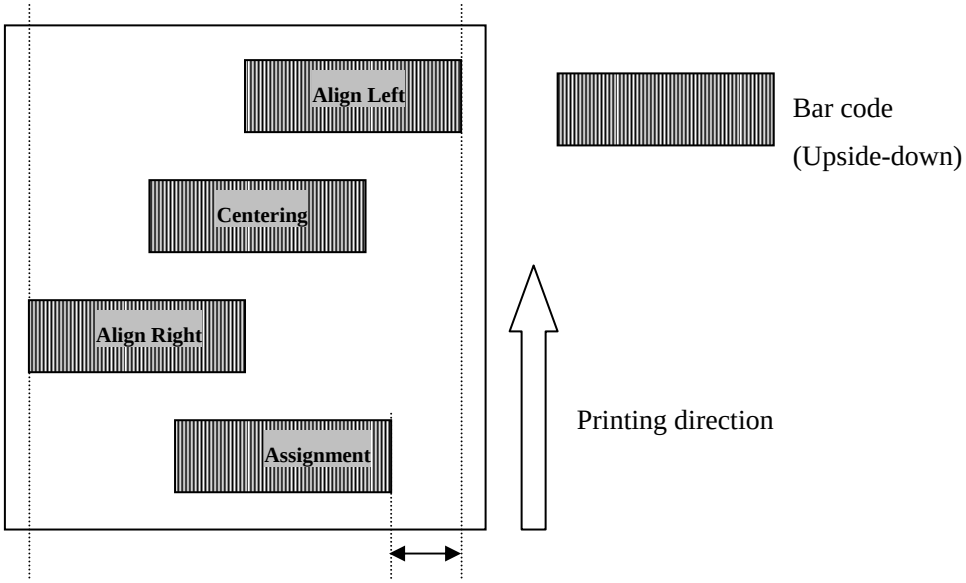
Bar Code Rotation Printing Using Rotate Special

Printing-position-change by *Alignment* assignment, at the time of upright printing.



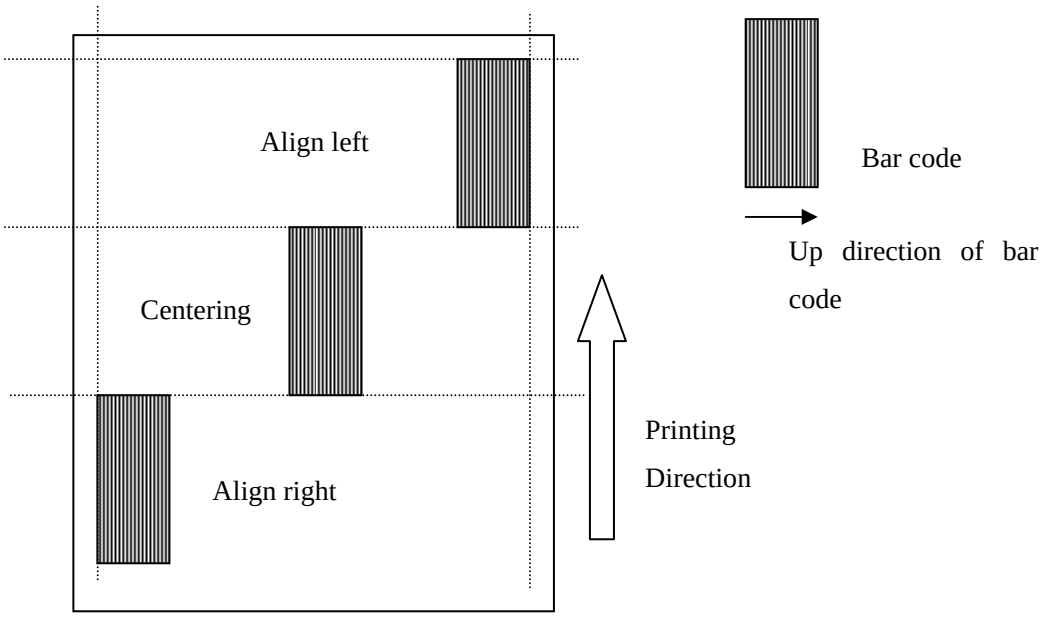
Position from the left most print column set by *Alignment*

Printing-position-change by *Alignment* assignment, at the time of upside-down printing.

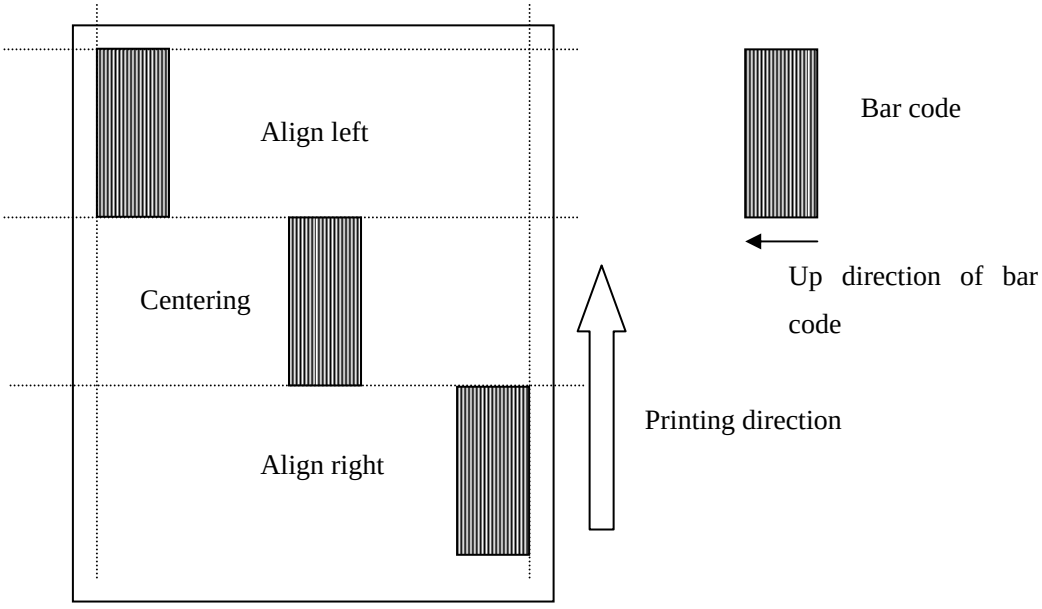


Position from the left most print column set y *Alignment*

Printing-position-change by *Alignment* assignment, at the time of 90-degree-rotation-to-the-right printing.



Printing-position-change by *Alignment* assignment, at the time of 90-degree-rotation-to-the-left printing.



*In case of two dimension barcode, all Aligment is fixed to PTR_BC_LEFT(-1) at the time of 90-degree-rotation-to-the-left/right printing.

Return Value

One of the following values is returned and contained in **ResultCode** Property

Value	Meaning
OPOS_SUCCESS(0)	Method finishes normally.
OPOS_E_ILLEGAL(106)	One of the following errors occurs. - No Station - Station does not support bar code printing. - Height or Width is 0 or too large. - Symbology is not supported. - There is a character not supported by Symbology. - Alignment is illegal value or too large. (When the total value of assigned Alignment value plus actual bar code printing width (the value calculated using the nearest value to Width) exceeds printable width, in case of Alignment absolute location assignment. - TextPosition is illegal value.
OPOS_E_NOHARDWARE(107)	POS Printer is OFF or OFFLINE.
OPOS_E_FAILURE(111)	OPOS Control is in error state. After deleting error, execute it .
OPOS_E_BUSY(113)	It cannot perform because it outputting.
OPOS_E_EXTENDED(114)	**ResultCodeExtended** = OPOS_EPTR_COVER_OPEN(201): POS Printer cover is open. (Only when **AsyncMode** is **FALSE,** it is returned.) **ResultCodeExtended** = OPOS_EPTR_REC_EMPTY(203): It runs out of paper. (Only when **AsyncMode** is **FALSE,** it is returned.) **ResultCodeExtended** = OPOS_FIT_EPTR_BLACKMARK(10001): Black Mark Error occurs. Only when **AsyncMode** is **FALSE,** it is returned.) **ResultCodeExtended** = OPOS_FIT_EPTR_FATAL (10003): Fatal error occurs. (Only when **AsyncMode** is **FALSE,** it is returned.) **ResultCodeExtended** = OPOS_FIT_EPTR_OVERHEAT (10006): Head overheat error occurs. (Only when **AsyncMode** is **FALSE,** it is returned.)
The others	Refer to the items of **ResultCode**.

PrintBitmap Method

Syntax

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

Parameter	Description
Station	It assigns PTR_S_RECEIPT(2)
FileName	Windows bitmap file name. The files must be uncompressed format. (Assign full pass or relative pass)
Width	Bitmap printing width. Refer to the following values.
Alignment	Bitmap printing position. Refer to the following values.

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

Value	Meaning
PTR_BM_ASIS(-11)	Prints the bitmap with one bitmap pixel per POS Printer dot.
The others	Bitmap width. It writes in the units defined by **MapMode**. Available values are from 1 to RecLineWidth Property value.

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

Value	Meaning
PTR_BM_LEFT(-1)	Align left
PTR_BM_CENTER(-2)	Centering
PTR_BM_RIGHT(-3)	Align right
The others	Distance from the left-most print column to the start of the bitmap. It is written in the units defined by **MapMode**. Total value with *Width* must not exceed restriction of *Width* Parameter. This parameter is not supported in 90 degrees rotating to the left or to the right. It works as PTR_BC_LEFT(-1).

Remarks

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

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

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

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

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

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

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

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

Return Value

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

Value	Meaning
OPOS_SUCCESS(0)	Method ends normally.
OPOS_E_ILLEGAL(106)	One of the following errors occurred. - No Station. - Station does not support bitmap printing. - Width is too large. - Alignment is illegal value or too large.
OPOS_E_NOHARDWARE(107)	POS Printer is OFF or OFFLINE.
OPOS_E_FAILURE(111)	OPOS Control is in error state. After deleting error, execute it .
OPOS_E_NOEXIST(109)	It could not find the file assigned by *FileName*.
OPOS_E_BUSY(113)	It cannot perform because it outputting.
OPOS_E_EXTENDED(114)	**ResultCodeExtended** = OPOS_EPTR_COVER_OPEN(201): POS Printer cover is open. (Only when **AsyncMode** is **FALSE,** it is returned.) **ResultCodeExtended** = OPOS_EPTR_REC_EMPTY(203): It runs out of paper. (Only when **AsyncMode** is **FALSE,** it is returned.) **ResultCodeExtended** = OPOS_EPTR_TOOBIG (206): Assigned bitmap is too large. Printable bitmap size is: Width (number of dot of **RecLineWidth** Property) Height (monochrome: 1662 dot, 2-clor setting: 831 dot) **ResultCodeExtended** = OPOS_EPTR_BADSYNTAX(207): Bitmap format is different from the assigned one. The assigned file is not bitmap file. When 2-color printing is set and data is the 24 bit bitmap. (When monochrome is specified, printing 24 bit bitmap is available.) **ResultCodeExtended** = OPOS_FIT_EPTR_BLACKMARK(10001): Black Mark Error occurs. (Only when **AsyncMode** is **FALSE,** it is returned.) **ResultCodeExtended** = OPOS_FIT_EPTR_FATAL(10003): Fatal error occurs. (Only when **AsyncMode** is **FALSE**, it is returned.) **ResultCodeExtended** = OPOS_FIT_EPTR_OVERHEAT (10006): Head overheat occurs. (Only when **AsyncMode** is **FALSE,** it is returned.)
The others	Refer to the items of **ResultCode**.

PrintImmediate Method

Syntax

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

Parameter	Description
Station	It assigns PTR_S_RECEIPT(2).
Data	Characters to be printed. It consists of printable characters, Escape Sequence, Carriage return (13 decimal), and New line/line feed (10 decimal). In details, refer to **BinaryConversion** Property.

Remarks

This method is called when to print *Data* with POS Printer. In executing asynchronous printing (**state**=OPOS_S_BUSY(3)), OPOS_E_BUSY(113) returns. During an error event (state=OPOS_S_ERROR(4)), OPOS_E_FAILURE(111) returns. It performs a reverse line feed in the case that characters per line of the text exceed maximum-characters-per-line.

The special character value within *Data* is as follows:

Value	Meaning
New line/Line Feed (10)	After printing the data in the buffer, it feeds to the next print line. (No need of carriage return for printing the line.)
Carriage Return(13)	If a Carriage Return immediately precedes a Line Feed, it is ignored. Carriage Return acts like a Line Feed. **ValidateData** Method is used to determine whether a Carriage Return without Line Feed is possible and whether a reverse line feed is required to support it.

Return Value

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

Value	Meaning
OPOS_SUCCESS(0)	Method ends normally.
OPOS_E_ILLEGAL(106)	No POS Printer assigned (except receipt)
OPOS_E_NOHARDWARE(107)	POS Printer is OFF or OFFLINE.
OPOS_E_FAILURE(111)	OPOS Control is in error state. After deleting error, execute it .
OPOS_E_BUSY(113)	It cannot perform because it outputting.
OPOS_E_EXTENDED(114)	**ResultCodeExtended** = OPOS_EPTR_COVER_OPEN(201): POS Printer cover is open. (Only when **AsyncMode** is **FALSE,** it is returned.) **ResultCodeExtended** = OPOS_EPTR_REC_EMPTY(203):It runs out of paper. (Only when **AsyncMode** is **FALSE,** it is returned.) **ResultCodeExtended** = OPOS_FIT_EPTR_BLACKMARK(10001): Black Mark Error occurs. Only when **AsyncMode** is **FALSE,** it is returned.) **ResultCodeExtended** = OPOS_FIT_EPTR_FATAL (10003): Fatal error occurs. (Only when **AsyncMode** is **FALSE,** it is returned.) **ResultCodeExtended** = OPOS_FIT_EPTR_OVERHEAT (10006): Head overheat error occurs. (Only when **AsyncMode** is **FALSE,** it is returned.)
The others	Refer to the items of **ResultCode**.

PrintMemoryBitmap Method

Syntax

LONG PrintMemoryBitmap (LONG *Station*, BSTR* *Data*, LONG *Type*, LONG *Width*, LONG *Alignment*);

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

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

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

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

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

Remarks

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

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

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

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

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

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

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

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

Return Value

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

Value	Meaning
OPOS_SUCCESS(0)	Method ends normally.
OPOS_E_ILLEGAL(106)	One of the following errors occurred. - No *Station*. - *Station* does not support bitmap printing. - *Width* is too large. - *Alignment* is illegal value or too large.
OPOS_E_NOHARDWARE(107)	POS Printer is OFF or OFFLINE.
OPOS_E_FAILURE(111)	OPOS Control is in error state. After deleting error, execute it.
OPOS_E_BUSY(113)	It cannot perform because it outputting.
OPOS_E_EXTENDED(114)	**ResultCodeExtended** = OPOS_EPTR_COVER_OPEN(201): POS Printer cover is open. (Only when **AsyncMode** is **FALSE,** it is returned.) **ResultCodeExtended** = OPOS_EPTR_REC_EMPTY(203): It runs out of paper. (Only when **AsyncMode** is **FALSE,** it is returned.) **ResultCodeExtended** = OPOS_EPTR_TOOBIG (206): Assigned bitmap is too large. Printable bitmap size is: Width (number of dot of **RecLineWidth** Property) Height (monochrome: 1662 dot, 2-clor setting: 831 dot) **ResultCodeExtended** = OPOS_EPTR_BADSYNTAX(207): Bitmap format is different from the assigned one. The assigned file is not bitmap file. **ResultCodeExtended** = OPOS_FIT_EPTR_BLACKMARK(10001): Black Mark Error occurs. (Only when **AsyncMode** is **FALSE,** it is returned.) **ResultCodeExtended** = OPOS_FIT_EPTR_FATAL(10003): Fatal error occurs. (Only when **AsyncMode** is **FALSE**, it is returned.) **ResultCodeExtended** = OPOS_FIT_EPTR_OVERHEAT (10006): Head overheat occurs. (Only when **AsyncMode** is **FALSE,** it is returned.)
The others	Refer to the items of **ResultCode**.

PrintNormal Method

Syntax

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

Parameter	Description
Station	It assigns PTR_S_RECEIPT(2).
Data	Characters to be printed It consists of printable characters, Escape Sequence, Carriage return (13 decimal), and New line/line feed (10 decimal). In details, refer to **BinaryConversion** Property.

Remarks

This method is called when to print *Data* with POS Printer. It performs a reverse line feed in the case that characters per line of the text exceed maximum-characters-per-line.

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

The special character value within *Data* is as follows:

Value	Meaning
New line/Line Feed (10)	After printing the data in the buffer, it feeds to the next print line. (No need of carriage return for printing the line.)
Carriage Return(13)	If a Carriage Return immediately proceeds a Line Feed, it is ignored. CarriageReturn acts like a Line Feed. **ValidateData** Method is used to determine whether a Carriage Return without Line Feed is possible and whether a reverse line feed is required to support it.

Return Value

One of the following values is returned and contained in **ResultCode** Property.

Value	Meaning
OPOS_SUCCESS(0)	Method ends normally.
OPOS_E_ILLEGAL(106)	No POS Printer assigned (except receipt)
OPOS_E_NOHARDWARE(107)	POS Printer is OFF or OFFLINE.
OPOS_E_FAILURE(111)	OPOS Control is in error state. After deleting error, execute it .
OPOS_E_BUSY(113)	It cannot perform because it is outputting.
OPOS_E_EXTENDED(114)	**ResultCodeExtended** = OPOS_EPTR_COVER_OPEN(201): POS Printer cover is open. (Only when **AsyncMode** is **FALSE,** it is returned.) **ResultCodeExtended** = OPOS_EPTR_REC_EMPTY(203): It runs out of paper. (Only when **AsyncMode** is **FALSE,** it is returned.) **ResultCodeExtended** = OPOS_FIT_EPTR_BLACKMARK(10001):

	Black Mark Error occurs. Only when **AsyncMode** is **FALSE,** it is returned.) **ResultCodeExtended** = OPOS_FIT_EPTR_FATAL (10003): Fatal error occurs. (Only when **AsyncMode** is **FALSE,** it is returned.) **ResultCodeExtended** = OPOS_FIT_EPTR_OVERHEAT (10006): Head overheat error occurs. (Only when **AsyncMode** is **FALSE,** it is returned.)
The others	Refer to the items of **ResultCode** and other items.

PrintTwoNormal Method

Syntax

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

Parameter	Description
Station	POS Printer station to be used.
Data1	Characters to be printed on the first station.
Data2	Characters to be printed on the second station. In details, refer to **BinaryConversion** Property.

Remarks

This method is called to print two character strings on two print stations simultaneously.

In this OPOS, this method is not supported because it is subject to slip printers.

Return Value

One of the following values is returned and contained in **ResultCode** Property, too.

Value	Meaning
OPOS_E_ILLEGAL(106)	No POS Printer assigned(except receipt)
The others	Refer to the items of **ResultCode**.

RotatePrint Method

Syntax

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

Parameter	Description
Station	PTR_S_RECEIPT(2) is assigned.
Rotation	Rotation Direction. Refer to the following values.

The *Rotation* value is as follows:

Value	Meaning
PTR_RP_RIGHT90(257)	Start of 90 degree rotation printing to the right (clockwise)
PTR_RP_LEFT90(258)	Start of 90 degree rotation printing to the left (counterclockwise)
PTR_RP_ROTATE180(259)	Start of 180 degree rotation printing (upside-down)
PTR_RP_NORMAL(1)	End of rotation printing

Remarks

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

If *Rotation* is PTR_RP_ROTATE180, upside-down printing mode starts. The data called by **PrintNormal** and **PrintImmediate RotatePrint** is printed upside-down till it is called at the *Rotation* Parameter setting of PTR_RP_NORMAL. The lines are printed in the order that they are sent to POS printer Control, and with the start of each line at the right margin of the printer. Printing methods of **PrintNormal** and **PrintImmediate** are used in upside-down printing mode.

When *Rotation* is PTR_RP_RIGHT90, PTR_RP_LEFT90, sideways printing mode starts. Data called by **PrintNormal** Method is buffered till it is called at the *Rotation* Parameter setting of PTR_RP_NORMAL. (In this case, the above method data is only buffered and printing does not start. At the same time, **AsyncMode** Property value does not affect the operation. In other words, **OutputID** is not assigned to the request and does not notify **OutputCompleteEvent,** either. In addition to this, each method succeeds, even if it is in error state in this case. For example, even though the power of POS Printer is off, error does not return in printing data buffering of **RotatePrint**.)

In case of sideways printing, width is automatically set to 0 to 1662dot for monochrome and 0 - 831dot for 2-color by the character data buffered by **PrintNormal** Method call. OPOS Control analyses character data buffered, and decides the width adjusting the maximum values of the width for all the lines (Refer to the following list). When BitMap print by the escape sequence and bar code print were appointed in letter data, the print of

BitMap and the bar code which do not fit into the width calculated by other letter data is not performed normally because inclusion of the width is not performed.
If the width of total character number exceeds 1662dot (831dot.for 2-color), printing width is 1662dot (831dot.for 2-color). The left data is printed, and it performs a reverse line feed within the page. In case, the width of character data is double size or more, by Escape sequence, the value is calculated by multiplying its multiple. (Example: In the case that it is assigned that font is A and that ANK character is double-width, the calculation result is 24 dots.)
If there is no buffered data, (when PrintNormal Method is not executed before) no printing is done.

Width per a character (dot)

Font (Refer to RecLineChars Property)	ANK	kanji
Font A	12 dot	24 dot
Font B	10 dot	20 dot
Font C	8 dot	16 dot

When **PrintBitmap** and **PrintMemoryBitmap** are issued in upside-down printing mode, bitmap is printed upside-down.
For the bitmap performed **SetBitmap** in upside-down printing mode, it is registered without upside-down.

When *Rotation* contains PTR_RP_BARCODE or PTR_RP_BITMAP, the barcode (printed with **PrintBarCode**) or the bitmap (printed with **PrintBitmap** or the 'ESC|#B' escape sequence) can be printed upside-down. Their directions of rotation are controlled by the **RecBarCodeRotationList** and **RecBitmapRotationList** properties respectively.
When *Rotation* contains PTR_RP_BARCODE, the contents of **RotateSpecial** are ignored.

When *Rotation* is PTR_RP_NORMAL, the rotation printing mode ends. If particular data is buffered by **PrintNormal** while sideways rotation printing mode is effective, the buffered data is printed. One whole block of rotated lines is treated just as one message.

When **ClearOutput** is called, the Rotation printing mode is terminated. Any buffered sideways printing line is deleted.

When the vertical length is specified by Escape Sequence with "n" times and print rotated, Print may be overlapped or exceed the paper. In this case, input "LF" code before Escape

Sequence to specify vertical length.

(Ex. To print the data 3 times length vertically, input "LF" code twice. To print the data "n" times length vertically, input "LF" code with the number of "n-1" times.)

Return Value

One of the following values is returned and contained in **ResultCode** Property, too.

Value	Meaning
OPOS_SUCCESS(0)	Method ends normally
OPOS_E_ILLEGAL(106)	No assigned POS Printer (except receipt) *Station* does not support assigned rotation. In different rotation mode, assign PTR_RP_NORMAL(1) or re-execute after clearing rotation printing with **ClearOutput.**
OPOS_E_NOHARDWARE(107)	POS Printer is OFF or OFFLINE.
OPOS_E_FAILURE(111)	OPOS Control is in error state. After deleting error, execute it.
OPOS_E_BUSY(113)	It cannot perform because it is outputting.
OPOS_E_EXTENDED(114)	**ResultCodeExtended** = OPOS_EPTR_COVER_OPEN(201): POS Printer cover is open. (Only when **AsyncMode** is **FALSE,** it is returned.) **ResultCodeExtended** = OPOS_EPTR_REC_EMPTY(203):It runs out of paper. (Only when **AsyncMode** is **FALSE,** it is returned.) **ResultCodeExtended** = OPOS_FIT_EPTR_BLACKMARK(10001): Black Mark Error occurs. Only when **AsyncMode** is **FALSE,** it is returned.) **ResultCodeExtended** = OPOS_FIT_EPTR_FATAL (10003): Fatal error occurs. (Only when **AsyncMode** is **FALSE,** it is returned.) **ResultCodeExtended** = OPOS_FIT_EPTR_OVERHEAT (10006): Head overheat error occurs. (Only when **AsyncMode** is **FALSE,** it is returned.)
The others	Refer to the items of **ResultCode**.

SetBitmap Method

Syntax

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

Parameter	Description
BitmapNumber	Number assigned to this bitmap. Valid values are from 1 to 20.
Station	Assign PTR_S_RECEIPT(2).
FileName	Windows bitmap file name. File must be uncompressed format. (To assign full pass or relative pass) If null number is set up, delete assigned *BitmapNumber* bit map from POSPrinter.
Width	Bitmap printing width. For value, refer to **PrintBitmap**.
Alignment	Location of bitmap printing. For value, refer to **PrintBitmp**.

Remarks

It is called when to save the information concerning bitmap soon to be printed.

Bitmap is printed by calling **PrintNormal** or **PrintImmediate** which has bitmap printing Escape Sequence inside printing data.

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

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

When 2-color printing is set, the bitmap data in 24 bit color cannot be registered.

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

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

monochrome data.

* When the bitmap registered to the parameter of the method by PTR_BM_LEFT(-1), PTR_BM_CENTER(-2) or PTR_BM_RIGHT(-3) is printed in 90 degrees rotating to the left or to the right by the **RotatePrint** method, the bit map is aligned left, centered or aligned right based on the printing standard of the normal direction (vertical).

Return Value

One of the following values is returned and placed in **ResultCode** Property.

Value	**Meaning**
OPOS_SUCCESS(0)	Method ends normally.
OPOS_E_ILLEGAL(106)	No assigned POS Printer (except receipt) - *BitmapNumber* is illegal value - *No* POS Printer (except receipt) - *Station* does not support bitmap printing. - Width is too large. - *Alignment* is illegal value or too large.
OPOS_E_NOEXIST(109)	Bitmap file assigned by *FileName* could not be found.
OPOS_E_FAILURE(111)	Bitmap data could not be transmitted to POS Printer. It is possible that printer cover is open, it run out of paper, or power of POS Printer is OFF.
OPOS_E_BUSY(113)	It cannot be executed because of device outputting.
OPOS_E_EXTENDED(114)	**ResultCodeExtended** = OPOS_EPTR_TOOBIG(206): Bitmap is too wide to print without conversion, or too large to convert **ResultCodeExtended** = OPOS_EPTR_BADFORMAT(207): Assigned file is not bitmap file or not-supported format.
The other	Refer to the item of **ResultCode**.

SetLogo Method

Syntax

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

Parameter	Description
Location	logo to set up, PTR_L_TOP(1), or PTR_L_BOTTOM(2)
Data	Characters to form logo. It consists of printable characters, Escape Sequence, Carriage Return (13 decimal), and New line/Line Feed (10 decimal)

Refer to **BinaryConversion** Property for further information.

Remarks

It is called when to save data character string as top logo or bottom logo.

Logo is printed by calling **PrintNormal** Method/**PrintImmediate** Method which includes escape sequence of top log/bottom logo within printing data.

Return Value

One of the following values is returned and contained in **ResultCode** Property.

Value	Meaning
OPOS_SUCCESS(0)	Method ends normally.
OPOS_E_BUSY(113)	It is not executed because it is outputting.
OPOS_E_ILLEGAL(106)	Illegal *Location* is assigned.
The others	Refer to the items of **ResultCode**.

TransactionPrint Method

Syntax

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

Parameter	Description
Station	Assign PTR_S_RECEIPT(2)
Control	Batch processing. Refer to the following values.

Following are *Control* values.

Value	Meani9ng
PTR_TP_TRANSACTION(11)	Start of batch processing
PTR_TP_NORMAL(12)	End of batch processing after printing buffered data.

Remarks

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

If *Control* is PTR_TP_TRANSACTION(11), it enters batch processing. Calls to the **PrintNormal**, **CutPaper**, **RotatePrint**, **PrintBarCode** and **PrintBitmap** methods after that, buffer printing data by Service Object until **TransactionPrint** is called by setting PTR_TP_NORMAL(12) to *Control* Parameter. (In this case, the above printing data of the above method is only buffered and printing does not start. Also, **AsyncMode** Property value does not affect the operation. In other words, the request does not assign **OutputID** and does not notify **OutputCompleteEvent**. In addition to that, each method succeeds, not affected by error state of POS Printer. For example, even though the power of the POS Printer is OFF, it does not return error at the time of calling of each method, during buffering of printing data by **TransactionPrint.**)

If *Control* is PTR_TP_NORMAL(12), it leaves batch processing. If data is buffered by the **PrintNormal**, **CutPaper**, **RotatePrint**, **PrintBarCode** and **PrintBitmap** methods, the data is to be printed. The whole batch processing is processed as one message. This method is executed synchronously if **AsyncMode** is **FALSE,** and asynchronously if **AsyncMode** is **TRUE**.

Batch processing mode is canceled by calling **ClearOutput.** The buffered printed lines are also deleted.

Be careful when to execute **RotatePrint** method. Until **TransactionPrint** is executed and it leaves batch processing mode, printing by the calling of **RotatePrint** Method with PTR_RP_RIGHT90(257), by the calling of **PrintNormal** Method, and by calling of **RotatePrint** Method with PTR_RP_ PTR_RP_NORMAL (1) is not done. Also, in case of calling of **RotatePrint** Method with PTR_RP_RIGHT90(257) and calling of **TransactionPrint** Method with PTR_TP_TRANSACTION(11), because buffering by **TransactionPrint** Method is prior to the others, it cannot print the data buffered during this time properly and without rotation printing. Accordingly, if **RotatePrint** Method is

executed, it must be done after **TransactionPrint** Method.

Return Value

One of the following values is returned and contained in **ResultCode** Property, too.

Value	Meaning
OPOS_SUCCESS(0)	Method ends normally.
OPOS_E_ILLEGAL(106)	No assigned POS Printer (except receipt)
OPOS_E_NOHARDWARE(107)	POS Printer is OFF or OFFLINE.
OPOS_E_FAILURE(111)	OPOS Control is in error state. After deleting error, execute it.
OPOS_E_BUSY(113)	It cannot perform because it is outputting. (Only when **AsyncMode** is **FALSE,** it is returned.)
OPOS_E_EXTENDED(114)	**ResultCodeExtended** = OPOS_EPTR_COVER_OPEN(201): POS Printer cover is open. (Only when **AsyncMode** is **FALSE,** it is returned.) **ResultCodeExtended** = OPOS_EPTR_REC_EMPTY(203): It runs out of paper. (Only when **AsyncMode** is **FALSE,** it is returned.) **ResultCodeExtended** = OPOS_FIT_EPTR_BLACKMARK(10001): Black Mark Error occurs. Only when **AsyncMode** is **FALSE,** it is returned.) **ResultCodeExtended** = OPOS_FIT_EPTR_FATAL (10003): Fatal error occurs. (Only when **AsyncMode** is **FALSE,** it is returned.) **ResultCodeExtended** = OPOS_FIT_EPTR_OVERHEAT (10006): Head overheat error occurs. (Only when **AsyncMode** is **FALSE,** it is returned.)
The others	Refer to the items of **ResultCode**.

ValidateData Method

Syntax

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

Parameter	Description
Station	PTR_S_RECEIPT(2) is assigned
Data	Data to be validated. It includes printable data and escape sequence. Refer to **BinaryConversion** Property for further information.

Remarks

It is called to validate whether the data sequence which includes one or more-than-one escape sequences **is** valid or not for assigned POS Printer, before calling **PrintNormal** Method and **PrintImmediate** Method.

This method does not do any printing, but is used to validate the POS Printer capability.

The escape sequence that returns OPOS_E_ILLEGAL(106) or that is not described in the case returning OPOS_E_FAILURE(111) is not determined and OPOS_SUCCESS(0) is always returned.

Return Value

One of the following values is returned and contained in **ResultCode** Property, too.

Value	Meaning
OPOS_SUCCESS(0)	Data is validated.
OPOS_E_ILLEGAL(106)	One or more-than-one escape sequences are out of the range, but Control can select valid alternatives. Or, subject station is not yet supported.
OPOS_E_FAILURE(111)	One or more-than-one escape sequences are not supported. There is no alternative to select.
The others	Refer to the items of **ResultCode**.

Cases which cause OPOS_E_ILLEGAL(106) returned.

Escape Sequence	Condition
Paper Cut/Feed and Paper Cut	'#' (percentage) is not supported. (Valid only when 1 to 100)
Underline	'#' (thickness) is not supported. (Valid only when 1 to 2)
Height Rate	'#' (rate) is not supported. (Valid only when 1 to 8)
Width Rate	'#' (rate) is not supported. (Valid only when 1 to 8)

Cases which cause OPOS_E_FAILURE(111) returned.

Escape Sequence	Condition
Feed and Cut and Stamp Printing	Not supported.
Stamp Printing	Not supported.
Bitmap Printing	'#' (Bitmap number) is out of range. (Valid only when 1 to 20)
Reverse Feed	Not supported.
Font Type Assignment	Not supported.
Italic	Not supported.
Custom	Not supported.
Red	Not supported when **CapRec2Color** is **FALSE**.
RGBColor	Not supported.
Shading	Not supported.
Color assignment	Not supported.
Subscript/Superscript	Not supported.

DrawRuled Line Method

Syntax

LONG DrawRuledLine (LONG *Station*, **BSTR** *PositionList*, **LONG** *LineDirection*, **LONG** *LineWidth*, **LONG** *LineStyle*, **LONG** *LineColor***);**

Remarks

This method is not supported by this OCX.

Return Value

Value	Meaning
OPOS_E_ILLEGAL(106)	This method is not supported

4.7. Event

DirectIOEvent Event

Syntax

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

Parameter	Description
EventNumber	Event Number. Specific value assigned by Service Object.
pData	Number value data pointer. Specific value changing according to event number and Service Object.
pString	Character string pointer. Specific value changing according to event number and Service Object. The format of this data depends on the value of **BinaryConversion** Property. Refer to **BinaryConversion** Property for further information.

Remarks

For direct transmission to Application, Service Object notifies it.

It becomes possible for Service Object to supply Application with Event not supported in Control Object.

In the case that the data other than normal status notification is received from the Printer, the following *EventNumber* notifies it, byte by byte.

In *EventNumber*, OPOS_FIT_DIO_NOT_ASB(=101) is set and the value of the decimal conversion of byte data (0 to 255) is ser in *pData*.

ErrorEvent Event

Syntax

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

Parameter	Description
ResultCode	Code which causes error event. Refer to the items of **ResultCode** for values.
ResultCodeExtended	Extension code which causes error event. Refer to the values below.
ErrorLocus	OPOS_EL_OUTPUT(1) is set up. Error occurs in asynchronous outputting.
pErrorResponse	Pointer to respond to error event. Refer to the values below.

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

Value	Meaning
OPOS_EPTR_COVER_OPEN(201)	POS Printer cover is open.
OPOS_EPTR_REC_EMPTY(203)	It runs out of paper.
OPOS_FIT_EPTR_FATAL (10003)	Fatal error occurs in POS Printer.
OPOS_FIT_EPTR_OVERHEAT (10006)	Head overheat occurred in POS Printer.
OPOS_FIT_EPTR_CUTTERJAM(10008)	A cutter jam error occurs.

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

Application sets one of the following values.

Value	Meaning
OPOS_ER_RETRY(11)	It tries its asynchronous processing again. It has already left error state.
OPOS_ER_CLEAR(12)	It deletes all the buffered data including asynchronous output. (It has same effect as **ClearOutput** Method.) It has already left error state.

Remarks

It will notify when OPOS Control state changes to error state, in executing method performable asynchronously.

OutputCompleteEvent Event

Syntax

void OutputCompleteEvent (LONG *OutputID*);

OutputID Parameter shows ID number of completed asynchronous-output-request.

Remarks

It notifies when the asynchronous-output-request started before, ends normally.

StatusUpdateEvent Event

Syntax

void StatusUpdateEvent (LONG *Status*);

Remarks

Status is set one of the following values.

Value	Meaning
PTR_SUE_COVER_OPEN(11)	POS Printer cover is open.
PTR_SUE_COVER_OK(12)	POS Printer cover is closed.
PTR_SUE_REC_EMPTY(24)	No receipt paper.
PTR_SUE_REC_NEAREMPTY(25)	Receipt paper is low.
PTR_SUE_REC_PAPEROK(26)	Receipt paper is ready.
PTR_SUE_REC_COVER_OPEN(62)	The cutter jam error occurred.
PTR_SUE_REC_COVER_OK(63)	The cutter jam error is recoverd.
PTR_SUE_IDLE(1001)	All the asynchronous output succeeds, or ends by being deleted. POS Printer's **State** is OPOS_S_IDLE(2) now. **FlagWhenIdle** Property must be **TRUE** in order to be notified by this event. And POS Printer Control automatically sets the property to **FALSE** before the event notifies it.
OPOS_SUE_POWER_ONLINE(2001)	Device is Power-ON and ready. (It notifies at the time of **PowerNotify** = OPOS_PN_ENABLED(1))
OPOS_SUE_POWER_OFF_OFFLINE(2004)	Device is Power-OFF or in OFF-LINE state. (It notifies at the time of **PowerNotify** = OPOS_PN_ENABLED(1))

OPOS_SUE_UF_PROGRESS(2100) + 1 to 100

(1 to 100 indicate the completion percentage)

Specifies the completion percentage of the firmware.

OPOS_SUE_UF_COMPLETE(2200)

The firmware is updated successfully.

OPOS_SUE_UF_FAILED_DEV_OK(2201)

The update firmware process failed but the device is still operational.

Remarks

It is notified when significant state change occurred on printer device side.

When device starts to enable, Control issues the first **StatusUpdateEvent** to let application know the device state.

Reference

CapPowerReporting Property and **PowerNotify** Property

5. OPOS Interface Specifications (Drawer)

5.1. List

List of Properties

Common	Type	Access	May Use After	Initial Value, Condition
BinaryConversion	Long	R/W	Open	OPOS_PR_NONE(0) Made writable after Open.
CapCompareFirmwareVersion	Boolean	R	Open	FALSE
CapPowerReporting	Long	R	Open	OPOS_PR_NONE(0)
CapStatisticsReporting	Boolean	R	Open	FALSE
CapUpdateFirmware	Boolean	R	Open	FALSE
CapUpdateStatistics	Boolean	R	Open	FALSE
CheckHealthText	String	R	Open	""
Claimed	Boolean	R	Open	FALSE
DeviceEnabled	Boolean	R/W	Open & claim	FALSE Made writable after Open and claim.
FreezeEvents	Boolean	R/W Open	Open	FALSE Made writable after Open.
PowerNotify	Long	R/W	Open	OPOS_PN_DISABLED(0) Unwritable
PowerState	Long	R	Open	OPOS_PS_UNKNOWN(2000)
ResultCode	Long	R	---	0
ResultCodeExtended	Long	R	Open	0
State	Long	R	---	1
ControObjectDescription	String	R	---	"CashDrawer OPOS Control Object"
ControlObjectVersion	Long	R	---	1013XXX
ServiceObjectDescription	String	R	Open	"CashDrawer OPOS Service Object"
ServiceObjectVersion	Long	R	Open	1013XXX
DeviceDescription	String	R	Open	"OP CashDrawer"
DeviceName	String	R	Open	The name set in the parameter at Open.

Specific	Type	Access	May Use After	Initial Value, Condition
CapStatus	Boolean	R	Open	The initial value is the value of the registry "CapStatus."
CapStatusMultiDrawerDetect	Boolean	R	Open	FALSE
DrawerOpened	Boolean	R	Open & Enable	FALSE

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

List of Methods

Common	Initialization
Open	---
Close	Open
ClaimDevice	Open
ReleaseDevice	Open & Claim
CheckHealth	Open & Enable
CompareFirmwareVersion	Open & Enable
DirectIO	Open
ResetStatistics	Open & Enable
RetrieveStatistics	Open & Enable
UpdateFirmware	Open & Enable
UpdateStatistics	Open & Enable

Specific	Initialization
OpenDrawer	Open & Enable
WaitForDrawerClose	Open & Enable

List of Events

Event	Initialization
DirectIOEvent	Open & Enable
StatusUpdateEvent	Open & Enable

5.2. Common Properties

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

We have two kinds of properties: Read-Only and Read/Write. If a property is for writing, R/W will be shown at the side of each property.

Only if return value has a special meaning, it will be shown. As for error, in case that access is done without satisfying initializing condition, refer to ResultCode property.

BinaryConversion Property R/W

Syntax

LONG BinaryConversion;

Remarks

The value of **BinaryConversion** is changeable to the following values, but it does not affect any method.

Value	Meaning
OPOS_BC_NONE (0)	Data is not converted and one character of Bstring is contained in one byte. (Default)
OPOS_BC_NIBBLE(1)	Each byte is converted to two characters.
OPOS_BC_DECIMAL(2)	Each byte is converted to three characters.

This property is initialized to OPOS_BC_NONE(0) by **Open** Method.

Return Value

When this property is set, either of the following values will be contained in **ResultCode** Property

Value	Meaning
OPOS_SUCCESS(0)	It succeeds in Property setting.
OPOS_E_ILLEGAL(106)	Illegal value is assigned.

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 to **TRUE** by the **Open** method

CapPowerReporting Property

Syntax

LONG CapPowerReporting;

Remarks

It identifies power reporting ability. The value to show power reporting ability is just the following.

Value	Meaning
OPOS_PR_NONE(0)	The power reporting ability does not work.

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

CapStatisticsReporting Property

Syntax

BOOL CapStatisticsReporting;

Remarks

This property is initialized to **FALSE** by the **Open** method. Statistics reporting is not supported.

CapUpdateFirmware Property

Syntax

BOOL CapUpdateFirmware;

Remarks

This property is initialized to **FALSE** by the **Open** method. The firmware can be updated.

CapUpdateStatistics Property

Syntax

BOOL CapUpdateStatistics;

Remarks

This property is initialized to **FALSE** by the **Open** method. Statistics reporting is not supported.

CheckHealthText Property

Syntax

BSTR CheckHealthText;

Remarks

It keeps the **CheckHealth** Method result called just before. The following are examples of the check.

- "Internal HCheck: Successful" (It succeeded in Internal check.)
- "External HCheck: Successful" (It succeeded in External check.)
- "External HCheck: Failure" (It failed in External check.)
- "InteractiveHCheck: Not Supported" (It is not supported.)

This value cannot be initialized before the first **CheckHealth** Method calling. (null character)

Claimed Property

Syntax

BOOL Claimed;

Remarks

TRUE: Exclusive Access right of device is acquired.

FALSE: Device is released in order to be shared to other applications.

In many cases, access event to Methods/property is possible after acquiring exclusive access right.

Claimed Property value is initialized to **FALSE** by **Open** Method.

ControlObjectDescription Property

Syntax

BSTR ControlObjectDescription;

Remarks

"Cash Drawer OPOS Control Object" is set.

Identifies the Control Object. It is a character string identifying the Control Object and the company that produced it and always readable.

ControlObjectVersion Property

Remarks

"1013XXX" is set. Holds the Control Object version number. Following three version levels are specified:

Version Level	Description
Major	The "millions" place. Holds the OPOS major version level.
Minor	The "thousands" place. Holds the OPOS minor version level. This is always set to 10 since this OPOS control conforms to OPOS version 1.10.
Build	The "units" place. Updated when corrections are made to the Control Object.

This property is always readable. (XXX varies depending on the time the Control Object is distributed.)

DeviceDescription Property

Syntax

BSTR DeviceDescription;

Remarks

"OP Cash Drawer" is set.

This property is a character string identifying the device, and holds the device and any pertinent information about it.

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

DeviceEnabled Property R/W

Syntax

BOOL DeviceEnabled;

Remarks

TRUE:

The device is made to enable (Operation state). If converted to **TRUE,** it is made to enable.

FALSE:

The device is made to disable. If converted to **FALSE**, if possible, it is made to disable physically.

Inputting after that is canceled and output operation is not possible.

Before this device is used, application must set this property **TRUE.**

This property is initialized to **FALSE** by **Open** Method.

Return Value

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

Value	Meaning
OPOS_SUCCESS(0)	It succeeds in Property setting.
OPOS_E_FAILRE(111)	It failed to access to device. Because it failed to open connection port, make sure whether any connection port is used form other program.
OPOS_E_NOHARDWARE(107)	POS Printer is OFF or OFFLINE or the cable is not connected. Clear the problem, and then execute the property again. (* In the case of USB interface, even if a POS printer is connected, this error occurs when a serial number set by registry is different from a serial number set by a POS printer.)

DeviceName Property

Syntax

BSTR DeviceName;

Remarks

One of "PT330-331SERDR1","PT330-331SERDR2", "PT330-331USBDR1","PT330-331USBDR2" is set.

This property shows the device and the any pertinent information about it. This is a short version of DeviceDescription and should be limited to 30 characters.

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

FreezeEvents Property R/W

Syntax

BOOL FreezeEvents;

Remarks

In case of **TRUE**, Event is not reported from Control.

Event is kept by Control till the freeze is terminated.

In case of **FALSE**, Event is reported from Control. If there is any Event kept during freeze, that Event is reported when **FreezeEvents** are converted to **FALSE**.

If interrupt by Event is undesirable, Application can select Event freeze.

This property is initialized to **FALSE** by **Open** Method.

Return Value

At the time of this property setting, the following value will be contained to **ResultCode** Property.

Value	Meaning
OPOS_SUCCESS(0)	It succeeds in property setting.

OpenResult Property

Syntax

LONG OpenResult;

Remarks

It keeps the latest result by **Open** Method. The values of **OpenResult** are just the following.

Value	Meaning
OPOS_SUCCESS(0)	Open is succeeded.
EOPEN_ALREADYOPEN(301)	It is already opened.
EOPEN_REGBADNAME(302)	There is no assigned Device Name Key in the registry.
EOPEN_REGPROGID(303)	Default value of Device Name Key is not readable, or it could not convert Programmatic ID kept there to effective Class ID.
EOPEN_CREATE(304)	It could not generate Service Object Instance, or could not obtain IDispatch Interface.
EOPEN_BADIF(305)	Service Object does not support one or more methods required by assignment version number.

This property is initialized by **Open** Method.

PowerNotify Property R/W

Syntax

LONG PowerNotify;

Remarks

This is the Power Notify Function type set by Application.

The values to show Power Notify Function are just the following.

Value	Meaning
OPOS_PN_DISABLED(0)	Control does not give any power notification to Application. **StatusUpdateEvent** concerning Power Notification is not given and any setting is not done to **PowerState** Property. (Default Value)

This property is initialized to OPOS_PN_DISABLED(0) by **Open** Method.

Return Value

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

Value	Meaning
OPOS_E_ILLEGAL(106)	It cannot set this property.

PowerState Property

Syntax

LONG PowerState;

Remarks

If it can be judged, Present Device Power State is set up. The value to show power state is just the following.

Value	Meaning
OPOS_PS_UNKNOWN(2000)	It cannot judge the device power state.

ResultCode Property

Syntax

LONG ResultCode;

Remarks

Each property sets this property. It is also set when it set writable property. This property is always readable. It returns **OPOS_E_CLOSED(101)** till it calls up **Open** Method.

The result code values are just the following.

Value	Meaning
OPOS_SUCCESS(0)	Normal Operation
OPOS_E_CLOSED(101)	You try to access the closed device.
OPOS_E_CLAIMED(102)	You tried to access to the device which other process accessed exclusively.
OPOS_E_NOSERVICE(104)	Control cannot communicate to Service Object. Perhaps, you must correct Set Up Error or Configuration Error.
OPOS_E_DISABLED(105)	It cannot be executed when Device is disabled.
OPOS_E_ILLEGAL(106)	You try to execute unavailable operation to Device or operation not supported. Or you use unavailable parameter value.
OPOS_E_NOHARDWARE(107)	POS Printer is OFF or OFFLINE.
OPOS_E_NOEXIST(109)	There is no file name (or other assigned value).
OPOS_E_FAILURE(111)	Device cannot execute the requested process.
OPOS_E_TIMEOUT(112)	The Service Object waiting for the response from Device does time-out, or the Control waiting for the response from Service Object does time-out.
OPOS_E_BUSY(113)	Present SO state cannot accept this request. For example, when asynchronous Output is executed, some methods cannot accept it.
OPOS_E_EXTENDED(114)	Particular Error State occurs. Error State Code can be confirmed with **ResultCodeExtended** Property

ResultCodeExtended Property

Syntax

LONG ResultCodeExtended;

Remarks

When **ResultCode** is **OPOS_E_EXTENDED(114)**, particular-to-class error information value described in device class explanation to this property is set.

When **ResultCode** is another value**,** Service Object can set particular-to-SO value to this property. These values have meaning only when Application processes them adding particular-to-SO values.

See also

ResultCode Property

ServiceObjectDescription Property

Syntax

BSTR ServiceObjectDescription;

Remarks

Character string which shows Service Object which supports device and the device manufacturer

" Cash Drawer OPOS Service Object" is set.

This property is initialized by **Open** Method

ServiceObjectVersion Property

Syntax

LONG ServiceObjectVersion;

Remarks

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

State Property

Syntax

LONG State;

Remarks

It shows the present state of Control.

Value	Meaning
OPOS_S_CLOSED(1)	Control is closed. (Default)
OPOS_S_IDLE(2)	Control is in normal state and not busy.

This property is always readable.

5.3. Common Methods

CheckHealth Method

Syntax

LONG CheckHealth (LONG *Level*);

Level parameter shows the type of health check executed with device. The following values can be assigned.

Value	Meaning
OPOS_CH_INTERNAL(1)	Health Check without using physical device is done. It always returns OPOS_SUCCESS.
OPOS_CH_EXTERNAL(2)	It does perfect test using device. If possible, it opens the drawer. If drawer is open successfully, OPOS_SUCCESS is returned. This method fails when exclusive access is done by other application.
OPOS_CH_INTERACTIVE(3)	It perform dialogue test. Not supported.

Remarks

It is called at the time of device condition test. The result of this method is contained in **CheckHealthText** Property. **CheckHealth** Method is always synchronous.

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

Return Value

Any of the following values will be returned and be contained in **ResultCode**.

Value	Meaning
OPOS_SUCCESS(0)	It shows that Health Check Procedure starts properly and that, when confirmed that, the device is working properly. However, you cannot decide about its properness until you see the result of the test.
OPOS_E_CLAIMED(102)	Other device is doing exclusive access.
OPOS_E_ILLEGAL(106)	*Level* parameter not supported is assigned.
OPOS_E_NOHARDWARE(107)	It failed in health check procedure. The POS Printer connected to the drawer is OFF or OFFLINE. It is only contained in the case of OPOS_CH_EXTERNAL(2) setting.
OPOS_E_TIMEOUT(112)	Connection is succeeded with printer connected to Drawer, but Drawer Open cannot be detected after timeout period. Only when OPOS_CH_EXTERNAL(2) is set and CapStatus proper is set TRUE, it will be stored.
The others	Refer to the items of **ResultCode**.

ClaimDevice Method

Syntax

LONG ClaimDevice (LONG *Timeout*);

Timeout Parameter shows maximum waiting time (in ms.) till it acquires exclusive access right.

In case of zero, even if it cannot acquire the Device Exclusive Access Method it returns the result immediately.

If **OPOS_FOREVER(-1)** is set, Method waits as long as till it can acquire exclusive access right.

Remarks

This method is called when exclusive access is required to device.

It is not requisite to acquire exclusive access right because drawer device is sharable device.

In case of success, **Claimed** Property is set to **TRUE**.

Return Value

One of the following values will be returned and contained to **ResultCode** Property.

Value	Meaning
OPOS_SUCCESS(0)	Exclusive access right is confirmed and proccessable device connection is assured. **Claimed** Property is **TRUE**. If this application has already accessed the device exclusively, it is returned, too.
OPOS_E_ILLEGAL(106)	Unavailable ***Timeout*** Parameter is assigned.
OPOS_E_TIMEOUT(112)	Other application tries to access the device exclusively and waits for being released. But ***Timeout*** Time (in ms.) has passed. Or, POS Printer Device does not become POS-Printer-Device-Process-Possible State even after ***Timeout*** Time (in milli-sec.) passes.
The others	Refer to the items of **Result Code.**

Close Method

Syntax

LONG Close ();

Remarks

It is called when to release Device and its resource.

If **DeviceEnabled** Property is **TRUE,** Device is forced to disable.

If **Claimed** Property is **TRUE,** at first, exclusive access will be released.

Don't execute at the time of Event Processing. (within Event Handler)

Return Value

One of the following values will be returned and contained to **ResultCode** Property.

Value	Meaning
OPOS_SUCCESS(0)	Device is disabled and closed.
The others	Refer to items of **ResultCode**.

CompareFirmwareVersion Method

Syntax

LONG CompareFirmWareVersion (String *FirmWareFileName***, Long** *result***);**

Remarks

This method is not supported.

Return Value

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

Value	Meaning
OPOS_E_ILLEGAL(106)	This method is not supported.

DirectIO Method

Syntax

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

Remarks

This method is called to communicate with the Service Object directly.

This method is not supported.

Value	Meaning
OPOS_E_ILLEGAL(106)	This method is unavailable.
Other Values	Refer to the ResultCode property.

Open Method

Syntax

LONG Open (BSTR *DeviceName*);

DeviceName Parameter assigns the name of the device which opens.

Device name of this control is just the following.

- Serial Interface "PT330-331SERDR1", "PT330-331SERDR2"
- USB Interface "PT330-331USBDR1", "PT330-331USBDR2"

Remarks

It is called to open the device.

When the **Open** method is successful, the Common properties and other Specific-to-Class Properties are initialized.

Return Value

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

Value	Meaning
OPOS_SUCCESS(0)	It succeeds in Open.
OPOS_E_NOSERVICE(104)	It cannot be connected to corresponding Service Object.
OPOS_E_ILLEGAL(106)	The applicable Control has already opened.
OPOS_E_NOEXIST(109)	It cannot find assigned DeviceName. It also includes the case in which ***Device Name*** is set with null number.
OPOS_E_FAILER(111)	It failed to initialize OPOS.

ReleaseDevice Method

Syntax

LONG ReleaseDevice ();

Remarks

Call this device when to release exclusive access of Device.

If **DeviceEnabled** Property is **TRUE** and exclusive device, Device is made to disable. Don't execute at the time of Event Processing. (within Event Handler)

Return Value

One of the following values will be returned and contained to **ResultCode** Property, too.

Value	Meaning
OPOS_SUCCESS(0)	Exclusive Access is released. **Claimed** Property becomes **FALSE.**
OPOS_E_ILLEGAL(106)	Application doesn't have exclusive access right to applicable Device.
The others	Refer to the explanation of **ResultCode** Property.

ResetStatistics Method

Syntax

LONG ResetStatistics (BSTR *StatisticsBuffer***);**

Remarks

This method is not supported.

Return Value

Value	Meaning
OPOS_E_ILLEGAL(106)	This method is not supported.

RetrieveStatistics Method

Syntax

LONG RetrieveStatistics (BSTR* *pStatisticsBuffer***);**

Remarks

This method is not supported.

Return Value

Value	Meaning
OPOS_E_ILLEGAL(106)	This method is not supported.

UpdateFirmwareMethod

Syntax

LONG UpdateFirmware (BSTR *FirmwareFileName***);**

Remarks

This method is not supported.

Return Value

Value	Meaning
OPOS_E_ILLEGAL(106)	This method is not supported.

UpdateStatistics Method

Syntax

LONG UpdateStatistics (BSTR *StatisticsBuffer***);**

Remarks

This method is not supported.

Return Value

Value	Meaning
OPOS_E_ILLEGAL(106)	This method is not supported.

5.4. Specific Property

CapStatus Property

Syntax

BOOL CapStatus;

Remarks

TRUE: It can notify open/close state of Drawer.

FALSE: It **cannot** notify open/close state of Drawer.

This property is initialized by **Open** Method.

The property for the first drawer defined by "XXX[Device name]DR1" is set to **TRUE** and the property for the second drawer defined by "XXX[Device name]DR2" is set to **FALSE.** Each device name includes "SER" (Serial Interface) or "USB" (USB Interface).

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

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

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

CapStatusMultiDrawerDetect Property

Syntax

BOOL CapStatusMultiDrawerDetect;

Remarks

FALSE: All the drawers are closed or one or more than one drawers are open. It does not mean that it can notify the open/close state of any one drawer among multiple-drawer-configuration.

This property is initialized by **Open** Method.

DrawerOpend Property

Syntax

BOOL DrawerOpened;

Remarks

TRUE: Drawer is open.

FALSE: Drawer is closed.

When **CapStatus** Property is FALSE, Device cannot notify state change and this **DrawerOpened** Property is always FALSE.

When device is made to enable, this property is initialized to appropriate value.

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

*The second drawer defined by "XXX[Device Name]DR2" cannot know open/close state, so it is always **FALSE.** Each device name includes "SER" (Serial Interface) or "USB" (USB Interface).

5.5. Specific Method

OpenDrawer Method

Syntax

LONG OpenDrawer ()

Remarks

It opens drawer. When CapStatus property is set as TRUE, OpenDrawerMethod waits until the status of Drawer open is detected.This method fails at the time of exclusive access from other application.

Return Value

One of the following values are returned and contained in **ResultCode** Property, too.

Value	Meaning
OPOS_SUCCESS(0)	It ends normally.
OPOS_E_CLAIMED(102)	Other device is accessing exclusively.
OPOS_E_FAILURE(111)	It could not transmit to device.
OPOS_E_TIMEOUT(112)	Connection is succeeded with printer connected to Drawer, but Drawer Open cannot be detected after timeout period. When CapStatus proper is set TRUE, it will be stored.
The others	Refer to the items of **ResultCode**.

WaitForDrawerClose Method

Syntax

LONG WaitForDrawerClose (LONG *BeepTimeout*, LONG *BeepFrequency*, LONG *BeepDuration*, LONG *BeepDelay*);

Remarks

This method is not supported.

Return Value

Value	Meaning
OPOS_E_ILLEGAL(106)	This method is not supported.

5.6. Event

DirectIOEvent Event

Syntax

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

Remarks

This method is not notified.

StatusUpdateEvent Event

Syntax

void StatusUpdateEvent (LONG *Status*);

Remarks

The latest drawer state is set in *Status* Parameter.

Value	Meaning
CASH_SUE_DRAWERCLOSED(0)	Drawer is closed.
CASH_SUE_DRAWEROPEN(1)	Drawer is open.

Remarks

It is notified when drawer open/close state changes.

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

* Nothing is notified when the device is enabled and Drawer is closed.
“CASH_SUE_DRAWEROPEN(1)" is notified when the device is enabled and Drawer is opened.

6. Registry Used by OCX

The following sections provide the registry setting used by this OCX.

Although the value of the registry can be set manually, it should be set in the setting program. Since the contents of the registry are read when the **Open** method is executed by the OCX, if the value is changed while the OCX is operated, it is not reflected in the behavior. To make the new settings effective, issue the Close method in the OCX before issuing the Open method

In this OCX, the serial number of the printer being connected now is written in the registry to enable the replacement of the printer without using a setup tool.
Because it is not possible to write it in HKEY_LOCAL_MACHINE of the registry when logging on in the User authority, the item for which writing is necessary corresponds by making it in HKEY_CURRENT_USER.

When the DeviceEnabled property is made True, the registry is written.

When USB interface is used, the serial number of the printer set by a setup tool is written in HKEY_LOCAL_MACHINE of the registry. When the DeviceEnabled property is made True, a necessary item is copied onto HKEY_CURRENT_USER at any time.

The item in HKEY_CURRENT_USER is as follows.

Serial interface:" PrinterSerialNo"
USB interface:"Port" and "PrinterSerialNo"
LAN interface:" PrinterSerialNo"

6.1. POS Printer (Serial Interface)

Setting of the first of them (Example.PT330-331SERPRT of Standard spec).

```
HKEY_LOCAL_MACHINE\SOFTWARE\OLEforRetail\ServiceOPOS
  \POSPrinter\
      DefaultPOSPrinter="PT330-331USBPRT"
      \PT330-331SERPRT= OPOS.POSPRINTER.SO.OKI.OP.1
          "Service"="OPPOSPrinterSO.dll"
          "Version"="1.13.1"
          "Description"="OP 1 Station Thermal POSPrinter"
          "Port"="USB"
          "SendTimeout"="12000"
          "InvertDrawerStatus"="0"
          "Smoothing"="1"
          "RecLineChars"="48"
          "RecLineCharsList"="48,57,72"
          "RecLineHeight"="24"
          "RecLineSpacing"="30"
          "RecLineWidth"="576"
          "CapRec2Color"="F"
          "BlackMarkEnabled"="F"
          "CodePage"="101,102,103,437,850,851,852,857,858,860,863,864,865,866,869,932,998,1252,28592,28597"
          "DefaultCodePage"="998"
          "LogFolder"=""
          "LogFileName"=""
          "LogLevel"="-1"
          "DataCompress"="F"
          "ErrorRecoveryMode"="1"
          "PowerOnNotify"="1"
          "PrintLevel"="100"
          “PrintSpeed”=“220”
          "PNESense"="E"
          "SerialNo"="N"
          "ID"="3"
          "BatchPrint"="D"
          "ResetTimeout"="15000"
          "PrinterSerialNo"=""
          "DefaultFont"="A"
          "FuncExtensionFont"="1"
          "ExtensionFont"="0"
          "AnkCharsSpacing"="0"
          "FuncFontC"="1"
          "CutAtCoverClose"="D"
          "ErrorAlert"="N"
          "BuzzerInterval"="2"
          "BuzzerRepetition"="3"
          " FuncPrintSpeedMAX "="220"
```

```
HKEY_CURRENT_USER\Software\OLEforRetail\ServiceOPOS
  \POSPrinter\
      \PT330-331SERPRT
          PrinterSerialNo=""
```

Setting of the second of them (Example.PT330-331SER2PRT of Standard spec).

```
HKEY_LOCAL_MACHINE\SOFTWARE\OLEforRetail\ServiceOPOS
   \POSPrinter\
        DefaultPOSPrinter="PT330-331USBPRT"
        \PT330-331SER2PRT= OPOS.POSPRINTER.SO.OKI.OP.1
            "Service"="OPPOSPrinterSO.dll"
            "Version"="1.13.1"
            "Description"="OP 1 Station Thermal POSPrinter"
            "Port"="USB"
            "SendTimeout"="12000"
            "InvertDrawerStatus"="0"
            "Smoothing"="1"
            "RecLineChars"="48"
            "RecLineCharsList"="48,57,72"
            "RecLineHeight"="24"
            "RecLineSpacing"="30"
            "RecLineWidth"="576"
            "CapRec2Color"="F"
            "BlackMarkEnabled"="F"
            "CodePage"="101,102,103,437,850,851,852,857,858,860,863,864,865,866,869,932,998,1252,28592,28597"
            "DefaultCodePage"="998"
            "LogFolder"=""
            "LogFileName"=""
            "LogLevel"="-1"
            "DataCompress"="F"
            "ErrorRecoveryMode"="1"
            "PowerOnNotify"="1"
            "PrintLevel"="100"
            “PrintSpeed”=“220”
            "PNESense"="E"
            "SerialNo"="N"
            "ID"="6"
            "BatchPrint"="D"
            "ResetTimeout"="15000"
            "PrinterSerialNo"=""
            "DefaultFont"="A"
            "FuncExtensionFont"="1"
            "ExtensionFont"="0"
            "AnkCharsSpacing"="0"
            "FuncFontC"="1"
            "CutAtCoverClose"="D"
            "ErrorAlert"="N"
            "BuzzerInterval"="2"
             "BuzzerRepetition"="3"
            " FuncPrintSpeedMAX "="220"
```

```
HKEY_CURRENT_USER\Software\OLEforRetail\ServiceOPOS
   \POSPrinter\
        \PT330-331SER2PRT
            PrinterSerialNo=""
```

6.2. POS Printer (USB Interface)

Setting of the first of them (Example.PT330-331USBPRT of Standard spec).

```
HKEY_LOCAL_MACHINE\SOFTWARE\OLEforRetail\ServiceOPOS
  \POSPrinter\
      DefaultPOSPrinter="PT330-331USBPRT"
      \PT330-331USBPRT= OPOS.POSPRINTER.SO.OKI.OP.1
          "Service"="OPPOSPrinterSO.dll"
          "Version"="1.13.1"
          "Description"="OP 1 Station Thermal POSPrinter"
          "Port"="COM1"
          "Protocol"="115200,N,8,1,x"
          "SendTimeout"="12000"
          "InvertDrawerStatus"="0"
          "Smoothing"="1"
          "RecLineChars"="48"
          "RecLineCharsList"="48,57,72"
          "RecLineHeight"="24"
          "RecLineSpacing"="30"
          "RecLineWidth"="576"
          "CapRec2Color"="F"
          "BlackMarkEnabled"="F"
          "CodePage"="101,102,103,437,850,851,852,857,858,860,863,864,865,866,869,932,998,1252,28592,28597"
          "DefaultCodePage"="998"
          "LogFolder"=""
          "LogFileName"=""
          "LogLevel"="-1"
          "DataCompress"="F"
          "ErrorRecoveryMode"="1"
          "PowerOnNotify"="1"
          "PrintLevel"="100"
          “PrintSpeed”=“220”
          "PNESense"="E"
          "SerialNo"="N"
          "ID"="1"
          "BatchPrint"="D"
          "PrinterSerialNo"=""
          "DefaultFont"="A"
          "FuncExtensionFont"="1"
          "ExtensionFont"="0"
          "AnkCharsSpacing"="0"
          "FuncFontC"="1"
          "CutAtCoverClose"="D"
          "ErrorAlert"="N"
          "BuzzerInterval"="2"
          "BuzzerRepetition"="3"
          " FuncPrintSpeedMAX "="220"
```

```
HKEY_CURRENT_USER\Software\OLEforRetail\ServiceOPOS
  \POSPrinter\
      \PT330-331USBPRT
          Port="USB"
          PrinterSerialNo=""
```

Setting of the second of them (Example.PT330-331USB2PRT of Standard spec).

```
HKEY_LOCAL_MACHINE\SOFTWARE\OLEforRetail\ServiceOPOS
   \POSPrinter\
        DefaultPOSPrinter="PT330-331USBPRT"
        \PT330-331USB2PRT= OPOS.POSPRINTER.SO.OKI.OP.1
             "Service"="OPPOSPrinterSO.dll"
             "Version"="1.13.1"
             "Description"="OP 1 Station Thermal POSPrinter"
             "Port"="COM2"
             "Protocol"="115200,N,8,1,x"
             "SendTimeout"="12000"
             "InvertDrawerStatus"="0"
             "Smoothing"="1"
             "RecLineChars"="48"
             "RecLineCharsList"="48,57,72"
             "RecLineHeight"="24"
             "RecLineSpacing"="30"
             "RecLineWidth"="576"
             "CapRec2Color"="F"
             "BlackMarkEnabled"="F"
             "CodePage"="101,102,103,437,850,851,852,857,858,860,863,864,865,866,869,932,998,1252,28592,28597"
             "DefaultCodePage"="998"
             "LogFolder"=""
             "LogFileName"=""
             "LogLevel"="-1"
             "DataCompress"="F"
             "ErrorRecoveryMode"="1"
             "PowerOnNotify"="1"
             "PrintLevel"="100"
             “PrintSpeed”=“220”
             "PNESense"="E"
             "SerialNo"="N"
             "ID"="4"
             "BatchPrint"="D"
             "PrinterSerialNo"=""
             "DefaultFont"="A"
             "FuncExtensionFont"="1"
             "ExtensionFont"="0"
             "AnkCharsSpacing"="0"
             "FuncFontC"="1"
             "CutAtCoverClose"="D"
             "ErrorAlert"="N"
             "BuzzerInterval"="2"
             "BuzzerRepetition"="3"
             " FuncPrintSpeedMAX "="220"
```

```
HKEY_CURRENT_USER\Software\OLEforRetail\ServiceOPOS
   \POSPrinter\
        \PT330-331USB2PRT
             Port="USB"
             PrinterSerialNo=""
```

6.3. POS Printer (LAN Interface)

Setting of the first of them (Example.PT330-331LANPRT_xxx.xxx.xxx.xxx of Standard spec).

```
HKEY_LOCAL_MACHINE\SOFTWARE\OLEforRetail\ServiceOPOS
  \POSPrinter\
      DefaultPOSPrinter="PT330-331USBPRT"
      \PT330-331LANPRT_xxx.xxx.xxx.xxx= OPOS.POSPRINTER.SO.OKI.OP.1
          "Service"="OPPOSPrinterSO.dll"
          "Version"="1.13.1"
          "Description"="OP 1 Station Thermal POSPrinter"
          "IPAddress"=""
          "SendTimeout"="30000"
          "Smoothing"="1"
          "RecLineChars"="48"
          "RecLineCharsList"="48,57,72"
          "RecLineHeight"="24"
          "RecLineSpacing"="30"
          "RecLineWidth"="576"
          "CapRec2Color"="F"
          "BlackMarkEnabled"="F"
          "CodePage"="101,102,103,437,850,851,852,857,858,860,863,864,865,866,869,932,998,1252,28592,28597"
          "DefaultCodePage"="998"
          "LogFolder"=""
          "LogFileName"=""
          "LogLevel"="-1"
          "DataCompress"="F"
          "ErrorRecoveryMode"="1"
          "PowerOnNotify"="1"
          "PrintLevel"="100"
          “PrintSpeed”=“220”
          "PNESense"="E"
          "SerialNo"="N"
          "BatchPrint"="D"
          "ResetTimeout"="15000"
          "PrinterSerialNo"=""
          "DefaultFont"="A"
          "FuncExtensionFont"="1"
          "ExtensionFont"="0"
          "AnkCharsSpacing"="0"
          "FuncFontC"="1"
          "Apartment"="0"
          "CutAtCoverClose"="D"
          "ErrorAlert"="N"
          "BuzzerInterval"="2"
          "BuzzerRepetition"="3"
          " FuncPrintSpeedMAX "="220"
```

```
HKEY_CURRENT_USER\Software\OLEforRetail\ServiceOPOS
  \POSPrinter\
      \PT330-331LANPRT_xxx.xxx.xxx.xxx
          PrinterSerialNo=""
```

6.4. The explanation of the registry item (POS Printer)

Value	Remarks
DefaultPOSPrinter	Default Device Name (PT330-331USBPRT)
Service	SO file name.
Version	SO version.
Description	SO detailed information.
Port	Communication port name.
Protocol	Communication protocol. (Only for serial interface)
SendTimeout	Send timeout. (ms) Duration of timeout from when data cannot be sent until OPOS_E_TIMEOUT(112) occurs while data is sent to the POS printer.
InvertDrawerStatus	Connection drawer invert flag. Due to the machine characteristics of the drawer, the POS printer may notify the open/close status reversed. Set '0' for normal operation and set '1' to reverse the status.
Smoothing	Specifies whether to perform smoothing process in the POS printer. For '1', when the RecLetterQuality property is set to TRUE, smoothing process is applied to the font that is equal to or larger than double-width. For '0', regardless the value of the RecLetterQuality property, smoothing process is not applied. (Default is '1')
RecLineChars	Characters to be printed in a line. Any value of RecLineCharsList is set. (Default is "48".)
RecLineCharsList	The list of the number of the characters that can be printed in a line by the POS printer. According to the conditions for printing, it is set as follows: Receipt is 80 mm and printing characters are 48 per line: "48,57,72" Receipt is 80 mm and printing characters are 42 per line: "42,51,64" Receipt is 58 mm and printing characters are 35 per line: "35,42,52" Receipt is 58 mm and printing characters are 32 per line: "32,38,48"
RecLineHeight	The height of the character in the Printer in dot. (Default value is "24")

Value	Remarks
RecLineSpacing	The spacing between print lines in the POS printer in dot. RecLineHeight to 127 dot can be set. In case of Extension font model, default value is "34"(Use the Extension font), and in case of another model, default value is "30"(Do not use Extension font).
RecLineWidth	Paper width of the POS printer in dot. According to the conditions for printing, it is set as follows: Receipt is 80 mm and printing characters are 48 per line: “576” Receipt is 80 mm and printing characters are 42 per line: “512” Receipt is 58 mm and printing characters are 35 per line: “420” Receipt is 58 mm and printing characters are 32 per line: “384”
CapRec2Color	Specifies whether 2-color printing is available. When 2-color printing is invalid in the Printer, set to “F”. (Default) When 2-color printing is valid, set to “T”. In case of PT330-331, make this value “F”.
LogLevel **LogFolder** **LogFileName**	Refer to the Chapter 7 “Log File” for details of the log file.
PowerOnNotify	This is created when the setting is performed with the setting program at least once, and hold the power on notification that is set with the setting program immediately before setting. (Regardless the value of the PowerNotify property of the OCX property, it specifies whether the printer send the message when the power is turned on.) When “0” is set, it is invalid, and when “1”, it is valid. It is not referred for this value by OCX, but please set “1” so that there is the case that cannot survey land in power supply OFF of the printer.
BlackMarkEnabled	Specifies whether the blackmarl sensor is mounted. For “T”, it is mounted and for “F”, it is not mounted. In case of PT330-331, make this value “F”.
CodePage	This is the value set in the CharacterSetList property of the POSPrinter.
DefaultCodePage	This is the value set in the CharacterSet property of the POSPrinter

Value	Remarks
DataCompress	This value is unused.
ErrorRecoveryMode	Specifies how the error is recovered. “0”: when error occurs, data that is not sent is sent again, “1”: data that is not sent is discarded and error state is recovered by the command. (Default value is “1”)
PrintLevel	Specifies the printing thickness. The default’s printing level of black is taken as 100 %. The percentage can be set from 70 to 130 by 10 %.
PrintSpeed	Specifies the printing speed. It can be set to the numbers form 100 to 220mm/s. The default is set to 220mm/s that is highest printing speed.
PNESense	Specifies whether to notify low paper. When it is set to “E” (enabled), notification is executed. When it is set to “D” (disabled), notification is not executed. “E” (enabled) is set as the default setting.
SerialNo	This value is unused.
ID	The identifier to share the port. This setting should not be changed.
BatchPrint	Specifies the batch printing feature. When "D" (Disable) is set, the batch feature is disabled, and when "E" (Enabled) is set, the batch feature is enabled. "D" (Disabled) is set as the default setting.
ResetTimeout	The timeout-time from the transmission of the reset command to the printer to unloading the USB driver and the timeout-time until the reload are set by the unit of ms.
PrinterSerialNo	The serial number of the printer connected.
DefaultFont	Definition of default font used. ("A":Font A, "B":Font B)
FuncExtensionFont	Extension font function. (1:Enable, 0:Disable) In case of Extension font model, this value is always "1", and in case of another model, this value is always "0".
ExtensionFont	Setting of whether to use the Extension font. In case of Extension font model, default value is "1"(Use the Extension font), and in case of another model, default value is "0"(Do not use Extension font).

Value	Remarks
AnkCharsSpacing	Setting of the space between ANK characters. In case of Extension font model, default value is "2"(2/203 inch, 0.25mm), and in case of another model, default value is "0"(0mm).
FuncFontC	Font C function. (1:Use Font C, 0:Do not use Font C) This value is always "1".
IPAddress	POS printer IP address
Apartment	Sets initializing and finalizing method for SxJcp32.dll in simultaneous control situation of 2 or more Printers, and this value is only used for Printers working with LAN interface connection. With "0" setting, 2 or more Printers can be controlled in same process space, but all OCXs must be booted from same process. With "1" setting, single Printer can be controlled by OCXs booted in plural processes, but LAN Printer must exist only single unit for each single process. For example, plural POS printer can be controlled by plural programs with standard EXE created by Visual Basic and pasted single OCX.Default setting is "0".
CutAtCoverClose	Specifies the cut operation when cover is closed. When “E” (Enable) is set, cut operation is executed when cover is closed. When “D”(Disable) is set, cut operation is not executed when cover is closed.
ErrorAlert	Specifies the warning method with the buzzer when the error occers. When “N” is set, the buzzer doesn't beep when the printer error occurs. When "O" is set, the buzzer beeps only once when the printer error occurs. When "C" is set, the buzzer keeps beeping from the occurrence of the error to release.
BuzzerInterval	Specifies the beeping interval of the buzzer. The value shows the pattern number. The value can be set from 1(Pattern 1) to 5(Pattern 5)
BuzzerRepetition	Specifies the repetition frequency of the buzzer. The repetition frequency of the buzzer can be set from 0 (None) to 5 (Five).

6.5. Drawer

(Example.PT330-331SERDR1/DR2,PT330-331SER2DR1/DR2 of Standard spec)

```
HKEY_LOCAL_MACHINE\SOFTWARE\OLEforRetail\ServiceOPOS
    \CashDrawer\
        DefaultCashDrawer="PT330-331USBDR1"
        AlternateCashDrawer="PT330-331USBDR2"
        \PT330-331SERDR1 = OPOS.CASHDRAWER.SO.OKI.OPDSO.1
            "Service"="OPCashDrawerSO.dll"
            "Version"="1.13.1"
            "Description"="OP Drawer1"
            "DeviceName"="PT330-331SERDR1"
            "ServiceObjectDescription"=" CashDrawer OPOS Service Object"
            "DeviceDescription"="OP CashDrawer"
            "Port"="COM1"
            "Protocol"="115200,N,8,1,x"
            "DrawerNo"="1"
            "OffTimer"="80"
            "OnTimer"="20"
            "ID"="1"
            "CapStatus"="T"
            "DrawerOpenTimeout"="2500"
        \PT330-331SERDR2 = OPOS.CASHDRAWER.SO.OKI.OPDSO.1
            "DeviceName"="PT330-331SERDR2"
            "DrawerNo"="2"
            "CapStatus"="F"
               Other settings are same asPT330-331PT330-331SERDR1
        \PT330-331SER2DR1 = OPOS.CASHDRAWER.SO.OKI.OPDSO.1
            "DeviceName"="PT330-331SER2DR1"
            "ID"="4"
               Other settings are same as PT330-331SERDR1
        \PT330-331SER2DR2 = OPOS.CASHDRAWER.SO.OKI.OPDSO.1
            "DeviceName"="PT330-331SER2DR2"
            "DrawerNo"="2"
            "CapStatus"="F"
            "ID"="4"
               Other settings are same asPT330-331PT330-331SERDR1
```

(Example. PT330-331USBDR1/DR2,PT330-331USB2DR1/DR2 of Standard spec)

```
HKEY_LOCAL_MACHINE\SOFTWARE\OLEforRetail\ServiceOPOS
    \CashDrawer\
        DefaultCashDrawer="PT330-331USBDR1"
        AlternateCashDrawer="PT330-331USBDR2"
        \PT330-331USBDR1 = OPOS.CASHDRAWER.SO.OKI.OPDSO.1
            "Service"="OPCashDrawerSO.dll"
            "Version"="1.13.1"
            "Description"="OP Drawer1"
            "DeviceName"="PT330-331USBDR1"
            "ServiceObjectDescription"=" CashDrawer OPOS Service Object"
            "DeviceDescription"="OP CashDrawer"
            "Port"="USB"
            "DrawerNo"="1"
            "OffTimer"="80"
            "OnTimer"="20"
            "ID"="3"
            "CapStatus"="T"
            "DrawerOpenTimeout"="2500"
        \PT330-331USBDR2 = OPOS.CASHDRAWER.SO.OKI.OPDSO.1
            "DeviceName"="PT330-331USBDR2"
            "DrawerNo"="2"
            "CapStatus"="F"
               Other settings are same asPT330-331PT330-331USBDR1
        \PT330-331USB2DR1 = OPOS.CASHDRAWER.SO.OKI.OPDSO.1
            "DeviceName"="PT330-331USB2DR1"
            "ID"="6"
               Other settings are same as PT330-331USBDR1
        \PT330-331USB2DR2 = OPOS.CASHDRAWER.SO.OKI.OPDSO.1
            "DeviceName"="PT330-331USB2DR2"
            "DrawerNo"="2"
            "CapStatus"="F"
            "ID"="6"
               Other settings are same asPT330-331PT330-331USBDR1
```

(Example. PT330-331USBDR1/DR2,PT330-331USB2DR1/DR2 of Standard spec)

```
HKEY_CURRENT_USER\Software\OLEforRetail\ServiceOPOS
    \CashDrawer\
        \PT330-331USBDR1
            Port=”USB”
        \PT330-331USBDR2
            Port=”USB”
        \PT330-331USB2DR1
            Port=”USB”
        \PT330-331USB2DR2
            Port=”USB”
```

6.6. The explanation of the registry item (Drawer)

Value	Remarks
DefaultCashDrawer	Default Device Name (PT330-331USBDR1)
AlternateCashDrawer	Alternate Device Name (PT330-331USBDR2)
CapStatus	Supports the open/close status of the Drawer. ("T" is supported, and "F" is not supported.)
Service	SO file name.
Version	SO version.
Description	SO detailed information.
DeviceDescription	The value of the DeviceDescription property.
DeviceName	The value of the DeviceName property.
DrawerNo	The number of the Drawer. (1 or 2)
OffTimer	The drawer kick on time (ms)
OnTimer	The drawer kick off time (ms)
Port	Communication port name.
Protocol	Setting value of serial port communication protocol.
ServiceObjectDescription	The value of the ServiceObjectDescription property.
DrawerOpenTimeout	Timeout period (ms) of Drawer open detection used by OpenDrawer Method and CheckHealth Method (External Check).

7. Log Files

The POSPrinter OPOS OCX Control has the function to output the log files by setting the **LogFolder**, **LogFileName**, **LogLevel** properties in the registry. The behavior of log file output is as follows:

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

   LogFolder = "C:\OPOS\Okidata\PT\Log\"
   For **LogFileName = "POSPrinter"**, when the log file is set as **"C:OPOS\Okidata\PT\log\[*Device Name*]POSPrinter.log[DD]"** and the date when the POS is executed is June 5, 2007, the file name of the log file is created as **"C:OPOS\Okidata\PT\log\[*Device Name*]POSPrinter.log05"**.

2. If the file created by the naming rule in the step 1 already exists and current month and the month when the file is updated last are different, then the existing log file is deleted. In other case, the file is created new or additionally written.

3. As the results, the log files of the last one-month are pooled on the POS (PC). Upper bound of the file size for each log file is not limited. If output includes the detailed logs, take special care to prevent shortage of the disk space.

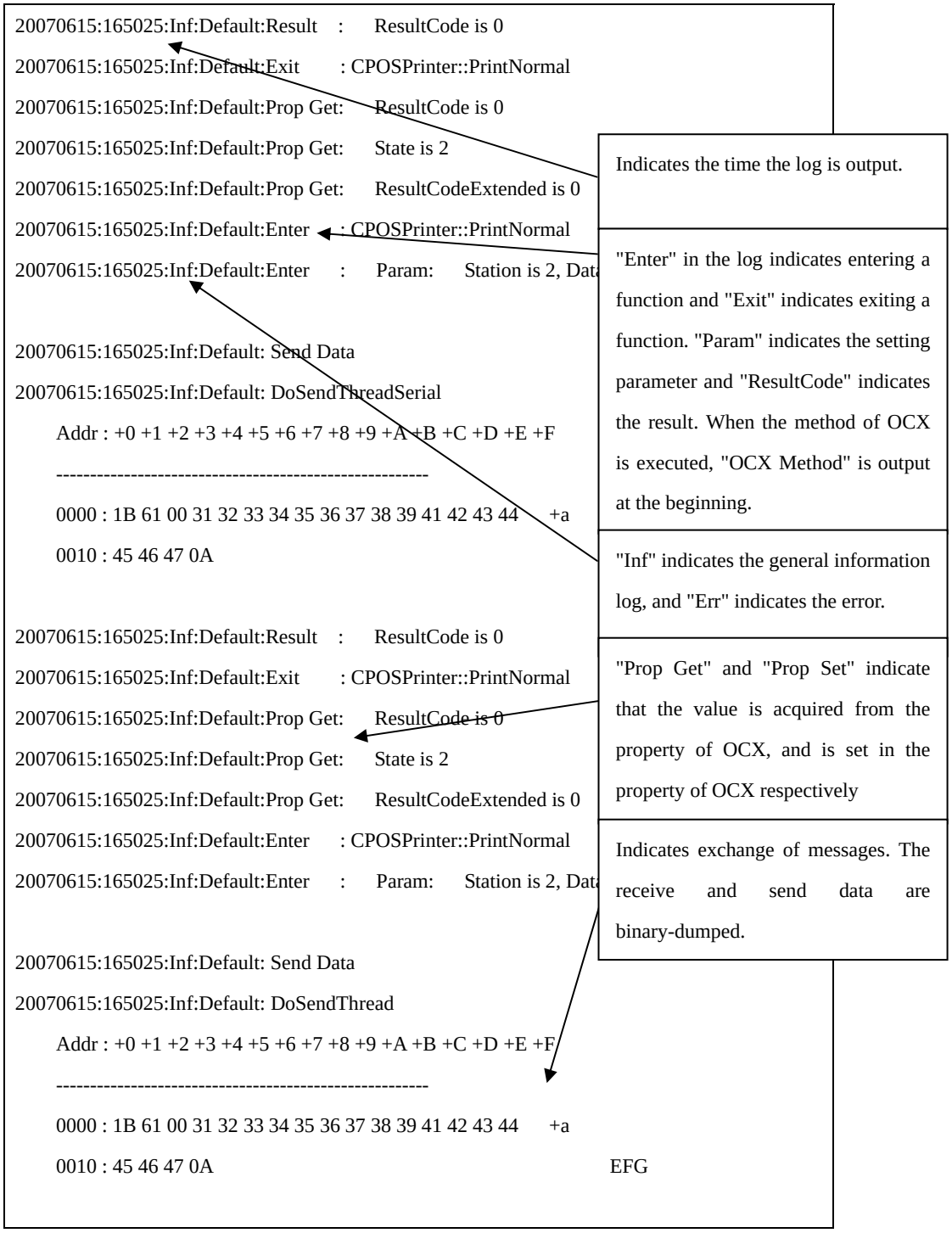
4. Based on the **LogLevel** setting, output the following log file.
   LogLevel = -1: Outputs no log.
   LogLevel = 0: Outputs normal trace log, warning and error log.
   LogLevel = 1: Outputs only error log.

* The Drawer OCX Control does not output the log file.

How to read the log file

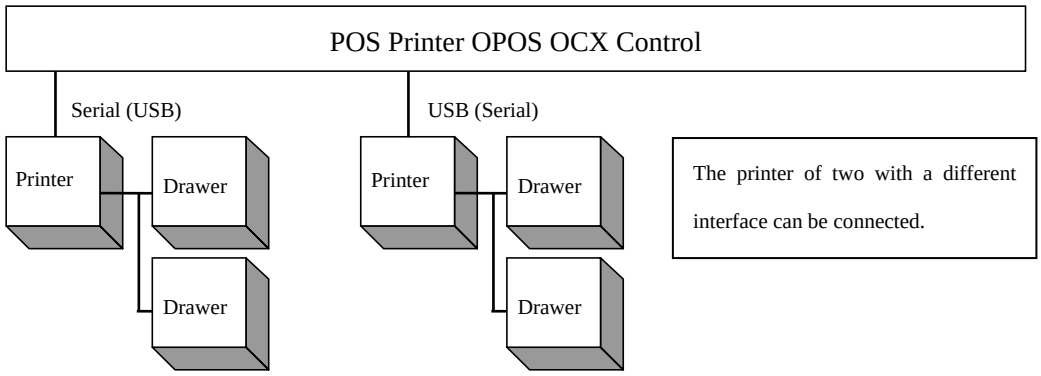
LogLevel = 0

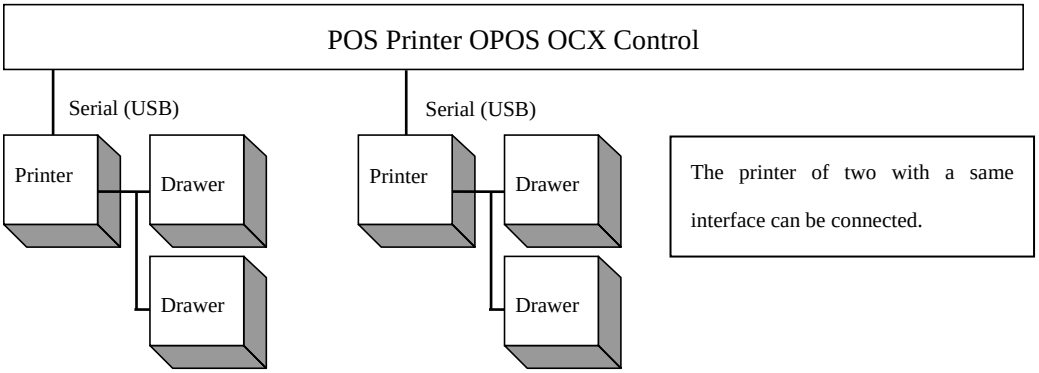
Example:



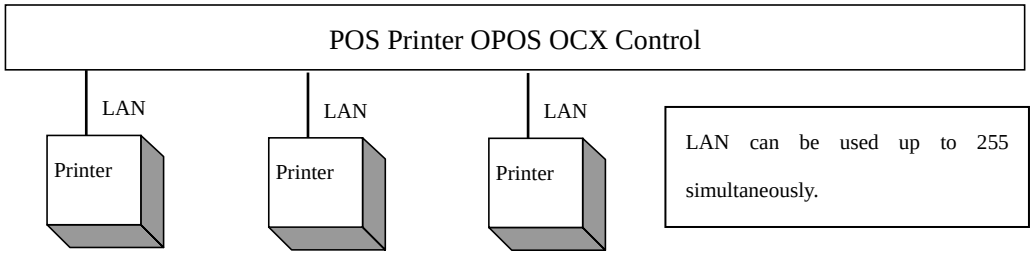
8. Using Multiple Printers

The POS Printer OPOS OCX control can use two of a serial and USB simultaneously.





The POS Printer OPOS OCX control can use up to 255 of LAN simultaneously.



9. Replacement of printer

In case of Serial and USB and LAN interface, the replacement of the printer is completed only by connecting a new printer after the application is ended, and executing the application again. (After the Close method is issued, the Open method and the ClaimDevice method are issued, and the DeviceEnabled property is changed into TRUE.)
The setup tool need not be used.

Note: The printer connected with XXXUSBPRT cannot be replaced with XXXUSB2PRT. This opposite cannot be done. The purpose is to prevent the replacement of the printer from happening by mistake at two USB connection.
When you want to replace it, connect printer one by one, and set it with the setup tool.

Note: When replacing USB printer to another USB one, depending on the environment of PC, OPOS_E_TIMEOUT may occur to take time to recognize the USB device. And when you connect new printer, USB device driver install wizard may be displayed. Change the DeviceEnabled property into TRUE again, after finishing this wizard (*1).

*1: When the printer is connected to PC in USB interface and "Found New Hardware Wizard" was displayed, set it in the following procedures.

For Windows 7 / Server 2008 R2

1. The message "Installing device driver software" is displayed in a balloon(lower right corner of the screen).
2. After a while, the message "Device driver software was not successfully installed" is displayed. This is not a problem.

For Windows Vista / Server 2008

1. The "Found New Hardware" is displayed, select [Don't show this message again for this device].
   * When "User Account Control" screen is displayed, select [Continue].

For Windows XP / Server 2003 / WEPOS / WEPOS2009

1. The "Found New Hardware Wizard" screen is displayed. Select [No, not this time] and then click [Next].
2. When "What do you want the wizard to do?" is displayed, select [Install from a list or specific location] and then click [Next].
3. Select [Search for the best driver in these locations], clear all checkboxes and then click [Next].
4. When "Cannot Install this Hardware" is displayed, click [Don't prompt me again to install this software] and then click [Finish].

For Windows 2000

1. The "Found New Hardware Wizard" screen is displayed, click [Next].
2. When "What do you want the wizard to do?" is displayed, select [Search for a suitable driver for my device] and then click [Next].
3. When [Locate Driver Files] is displayed, clear all checkboxes and then click [Next].
4. When "Driver Files Search Results" is displayed, select [Disable the device] and then click [Finish].

Revision History

Revision #	Date	Revised Content
1.0	2011/06/29	Newly created. Conforms OPOS 1.13.
1.1	2011/07/08	2.3 Installation Procedure - Image of 3 is changed.
1.2	2012/03/01	2.6. Setting Program Usage - Description of print speed is changed. 4.2. Print Data and Escape Sequences - Description of PDF417 is Added. 4.6. Exclusive-Use Method - Description of PDF417 is Added in PrintBarcode Method. 6.Registry Used by OCX - Description of print speed is changed.

Oki contact details

OkicontactName. Oki Systems (UK) Limited
OkicontactAddress.550 Dundee Road
Slough Trading Estate
Slough
Berkshire
SL1 4LE

Tel:+44 (0) 1753 819819
Fax:+44 (0) 1753 819899
http://www.oki.co.uk

Oki Systems Ireland Limited
The Square Industrial Complex
Tallaght
Dublin 24

Tel:+353 (0) 1 4049590
Fax:+353 (0)1 4049591
http://www.oki.ie

Oki Systems Ireland Limited - Northern Ireland
19 Ferndale Avenue
Glengormley
BT36 5AL
Northern Ireland

Tel:+44 (0) 7767 271447
Fax:+44 (0) 1 404 9520
http://www.oki.ie

Technical Support for all Ireland:
Tel:+353 1 4049570
Fax:+353 1 4049555
E-mail: tech.support@oki.ie

OKI Systems (Czech and Slovak), s.r.o.
IBC – Pobřežní 3
186 00 Praha 8
Czech Republic

Tel: +420 224 890158
Fax:+420 22 232 6621
Website: www.oki.cz, www.oki.sk

Oki Systems (Deutschland) GmbH
Hansaallee 187
40549 Düsseldorf

Tel: 01805/6544357**
01805/OKIHELP**
Fax: +49 (0) 211 59 33 45
Website:
www.okiprintingsolutions.de
info@oki.de

**0,14€/Minute aus dem dt. Festnetz der T-Com (Stand 11.2008)

Διανομέας των συστημάτων OKI
CPI S.A1 Rafailidou str.
177 78 Tavros
Athens
Greece

Tel: +30 210 48 05 800
Fax:+30 210 48 05 801
EMail:sales@cpi.gr

Oki Systems (Iberica), S.A.U
C/Teide, 3
San Sebastian de los Reyes
28703, Madrid

Tel:+34 91 3431620
Fax: +34 91-3431624
Atención al cliente: 902 36 00 36
Website: www.oki.es

Oki Systèmes (France) S.A.
44-50 Av. du Général de Gaulle
94246 L'Hay les Roses
Paris

Tel:+33 01 46 15 80 00
Télécopie:+33 01 46 15 80 60
Website: www.oki.fr

OKI Systems (Magyarország) Kft.
Capital Square
Tower 2
7th Floor
H-1133 Budapest,
Váci út 76
Hungary

Telefon: +36 1 814 8000
Telefax: +36 1 814 8009
Website: www.okihu.hu

OKI Systems (Italia) S.p.A.
via Milano, 11,
20084 Lacchiarella (MI)

Tel:+39 (0) 2 900261
Fax:+39 (0) 2 90026344
Website: www.oki.it

OKI Printing Solutions
Platinium Business Park II, 3rd Floor
ul. Domaniewska 42
02-672 Warsaw
Poland

Tel:+48 22 448 65 00
Fax:+48 22 448 65 01
Website: www.oki.com.pl
E-mail: oki@oki.com.pl
Hotline: 0800 120066
E-mail: tech@oki.com.pl

Oki Systems (Ibérica) S.A.
Sucursal Portugal
Edifício Prime -
Av. Quinta Grande 53
7º C Alfragide
2614-521 Amadora
Portugal

Tel:+351 21 470 4200
Fax:+351 21 470 4201
Website:www.oki.pt
E-mail : oki@oki.pt

Oki Service
Serviço de apoio técnico ao Cliente
Tel: 808 200 197
E-mail : okiserv@oki.pt

OKI Europe Ltd. (Russia)
Office 702, Bldg 1
Zagorodnoye shosse
117152, Moscow

Tel: +74 095 258 6065
Fax: +74 095 258 6070
e-mail: info@oki.ru
Website: www.oki.ru

Technical support:
Tel: +7 495 564 8421
e-mail: tech@oki.ru

Oki Systems (Österreich)
Campus 21
Businesszentrum Wien Sued
Liebermannstrasse A02 603
22345 Brun am Gebirge

Tel: +43 223 6677 110
Drucker Support:
+43 (0) 2236 677110-501
Fax Support:
+43 (0) 2236 677110-502
Website: www.oki.at

OKI Europe Ltd. (Ukraine)
Raisy Opkinoy Street,8
Building B, 2nd Floor,
Kiev 02002
Ukraine

Tel: +380 44 537 5288
e-mail: info@oki.ua
Website: www.oki.ua

OKI Sistem ve Yazıcı Çözümleri Tic. Ltd. Şti.
Harman sok Duran Is Merkezi,
No:4, Kat:6,
34394, Levent
İstanbul

Tel: +90 212 279 2393
Faks: +90 212 279 2366
Web: www.oki.com.tr
www.okiprintingsolutions.com.tr

Oki Systems (Belgium)
Medialaan 24
1800 Vilvoorde

Helpdesk: 02-2574620
Fax: 02 2531848
Website: www.oki.be

AlphaLink Bulgaria Ltd.
2 Kukush Str.
Building "Antim Tower", fl. 6
1463 Sofia, Bulgaria

tel: +359 2 821 1160
fax: +359 2 821 1193
Website: http://bulgaria.oki.com

OKI Printing Solutions
Herstedøstervej 27
2620 Albertslund
Danmark

Adm.: +45 43 66 65 00
Hotline: +45 43 66 65 40
Salg: +45 43 66 65 30
Fax: +45 43 66 65 90
Website: www.oki.dk

Oki Systems (Finland) Oy
Polaris Capella
Vänrikinkuja 3
02600 Espoo

Tel: +358 (0) 207 900 800
Fax: +358 (0) 207 900 809
Website: www.oki.fi

Oki Systems (Holland) b.v.
Neptunustraat 27-29
2132 JA Hoofddorp

Helpdesk: 0800 5667654
Tel:+31 (0) 23 55 63 740
Fax:+31 (0) 23 55 63 750
Website: www.oki.nl

Oki Systems (Norway) AS
Tevlingveien 23
N-1081 Oslo

Tel:+47 (0) 63 89 36 00
Telefax:+47 (0) 63 89 36 01
Ordrefax:+47 (0) 63 89 36 02
Website: www.oki.no

General Systems S.R.L. (Romania)
Sos. Bucuresti-Ploiesti Nr. 135.
Bucharest 1
Romania

Tel: +40 21 303 3138
Fax: +40 21303 3150
Website: http://romania.oki.com

Var vänlig kontakta din Återförsäljare i första hand, för konsultation. I andra hand kontakta

Oki Systems (Sweden) AB
Borgafjordsgatan 7
Box 1191
164 26 Kista

Tel. +46 (0) 8 634 37 00
e-mail:
info@oki.se för allmänna frågor om Oki produkter

support@oki.se för teknisk support gällandes Oki produkter

Vardagar: 08.30 - 12.00,
13.00 - 16.00
Website: www.oki.se

Oki Systems (Schweiz)
Baslerstrasse 15
CH-4310 Rheinfelden

Support deutsch +41 61 827 94 81
Support français +41 61 827 94 82
Support italiano +41 061 827 9473
Tel: +41 61 827 9494
Website: www.oki.ch

Oki Data Americas Inc.(United States)
2000 Bishops Gate Blvd.
Mt. Laurel, NJ 08054
USA

Tel: 1-800-654-3282
Fax: 1-856-222-5247
http://WWW.OKIPRINTINGSOLUTIONS.COM
http://my.okidata.com

Oki Data Americas Inc.(Canada • Canadá)
4140 B Sladeview Crescent Units 7&8
Mississauga, Ontario
Canada L5L 6A1

Tél: 1-905-608-5000
Téléc: 1-905-608-5040
http://WWW.OKIPRINTINGSOLUTIONS.COM

Oki Data Americas Inc.(América Latina (OTRO))
2000 Bishops Gate Blvd.
Mt. Laurel, NJ 08054
USA

Tel (Español): 1-856-222-7496
1-856-222-5276
Fax: 1-856-222-5260
Email: LASatisfaction@okidata.com

Oki Data de Mexico, S.A. de C.V.
Mariano Escobedo #748, Piso 8
Col. Nueva Anzures
C.P. 11590, México, D.F.

Tel: 52-555-263-8780
Fax: 52-555-250-3501
http://WWW.OKIPRINTINGSOLUTIONS.COM

Oki Data do Brasil, Ltda.
Rua Avenida Alfrefo Egidio de souza Aranha
100-4º andar-Bloco C Chacara Santo Antonio
Sao Paulo, Brazil 04726-170

Tel: 55-11-3444-6747 (Grande São Paulo)
0800-11-5577 (Demais localidades)
Fax: 5511-3444-3501
e-mail: okiserv@okidata.com.br
http://WWW.OKIPRINTINGSOLUTIONS.COM

Oki Data (Singapore) Pte. Ltd.
438A Alexandra Road #02-11/12,
Lobby 3, Alexandra Technopark
Singapore(119967)

Tel:(65) 6221 3722
Fax:(65) 6594 0609
http://www.okidata.com.sg

Oki Systems (Thailand) Ltd.
1168/81-82 Lumpini Tower,
27th Floor Rama IV Road
Tungmahamek, Sathorn
Bangkok 10120

Tel:(662) 679 9235
Fax:(662) 679 9243/245
http://www.okisysthai.com

Oki Systems (Hong Kong) Ltd.
Suite 1908, 19/F, Tower 3,
China Hong Kong City
33 Canton Road, TsimShaTsui,
Kowloon, Hong Kong

Tel: (852) 3543 9288
Fax: (852) 3549 6040
http://www.okiprintingsolutions.com.hk

Oki Data(Australia) Pty Ltd.
Levwl1 67 Epping Road, Macquarie Park
NSW 2113, Australia

Tel: +61 2 8071 0000
(Support Tel: 1800 807 472)
Fax: +61 2 8071 0010
http://www.oki.com.au

Comworth Systems Ltd.
8 Antares Place Mairangi Bay,
Auckland, New Zealand

Tel:(64) 9 477 0500
Fax:(64) 9 477 0549
http://www.comworth.co.nz

Oki Data(S) P Ltd. Malaysia Rep Office
Suite 21.03, 21st Floor Menara IGB,
Mid Valley City,
Lingkaran Syed Pura 59200,
Kuala Lumpur, Malaysia

Tel: (60) 3 2287 1177
Fax: (60) 3 2287 1166

Oki Data Corporation
4-11-22 Shibaura, Minato-ku,Tokyo
108-8551, Japan

www.okiprintingsolutions.com