



IFD SERIES POS PRINTER AND
CASH DRAWER OPOS CONTROL
APPLICATION PROGRAMMER'S GUIDE

U00122125401

Seiko Instruments Inc.

IFD SERIES POS PRINTER AND CASH DRAWER OPOS CONTROL APPLICATION PROGRAMMER'S GUIDE


First Edition	Ver.1.03	June 2010
Second Edition	Ver.1.20	December 2010

Copyright © 2010 by Seiko Instruments Inc.
All rights reserved.

Microsoft®, Windows®, Visual Studio®, Visual Basic® and Visual C++® are registered trademarks of Microsoft Corporation USA.

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

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

SII  is a trademark of Seiko Instruments Inc.

Chapter1	: OVERVIEW.....	9
1.1.	Scope of This Document.....	9
1.2.	Overview of OPOS Control	10
(1)	Structural Drawing of OPOS Control	10
(2)	Definitions	11
1.3.	Limitation (POS Printer, Cash Drawer).....	11
(1)	POS Printer	11
(2)	Cash Drawer	11
Chapter2	: INSTALLATION	12
2.1.	Conditions of Installation	12
2.2.	Installation Media	12
2.3.	Default Setting for Installation	13
2.4.	Installation Procedure	16
(1)	Manual mode	16
(2)	Silent mode.....	20
2.5.	Uninstallation Procedure	22
2.6.	Installation File List.....	24
Chapter3	: OPOS CONTROL	25
3.1.	General	25
3.2.	POS Printer.....	25
Chapter4	: OPOS INTERFACE SPECIFICATION (POS Printer).....	26
4.1.	Summary.....	26
4.2.	Data Characters and Escape Sequences	34
(1)	Escape Sequence operated when specified	34
(2)	Escape Sequence operated during printing	36
(3)	Escape Sequence operated when printing	36
4.3.	Common Properties	38
	BinaryConversion Property R/W	38
	CapCompareFirmwareVersion Property	39
	CapPowerReporting Property	40
	CapStatisticsReporting Property.....	40
	CapUpdateFirmware Property	40
	CapUpdateStatistics Property.....	41
	CheckHealthText Property	41
	Claimed Property	41
	DeviceEnabled Property R/W	42
	FreezeEvents Property R/W	43
	OpenResult Property	44
	OutputID Property	44

PowerNotify Property R/W	45
PowerState Property	46
ResultCode Property	47
ResultCodeExtended Property	48
State Property	49
ControlObjectDescription Property	49
ControlObjectVersion Property	49
ServiceObjectDescription Property	50
ServiceObjectVersion Property	50
DeviceDescription Property	50
DeviceName Property	50
4.4. Specific Properties	51
CapCharacterSet Property	51
CapCoverSensor Property	51
CapMapCharacterSet Property	51
CapTransaction Property	52
CapJrnPresent Property	52
CapJrn2Color Property	52
CapJrnBold Property	52
CapJrnDhigh Property	53
CapJrnDwide Property	53
CapJrnDwideDhigh Property	53
CapJrnEmptySensor Property	54
CapJrnItalic Property	54
CapJrnNearEndSensor Property	54
CapJrnUnderline Property	54
CapJrnCartridgeSensor Property	55
CapJrnColor Property	55
CapRecPresent Property	56
CapRec2Color Property	56
CapRecBarCode Property	56
CapRecBitmap Property	56
CapRecBold Property	57
CapRecDhigh Property	57
CapRecDwide Property	57
CapRecDwideDhigh Property	58
CapRecEmptySensor Property	58
CapRecItalic Property	58
CapRecLeft90 Property	58
CapRecNearEndSensor Property	59
CapRecPapercut Property	59
CapRecRight90 Property	59
CapRecRotate180 Property	60

CapRecStamp Property.....	60
CapRecUnderline Property.....	60
CapRecCartridgeSensor Property	60
CapRecColor Property	61
CapRecMarkFeed Property	61
CapRecPageMode Property.....	61
AsyncMode Property R/W	62
CartridgeNotify Property R/W	62
CharacterSet Property R/W	63
CharacterSetList Property	63
CoverOpen Property	63
ErrorLevel Property	64
ErrorStation Property.....	64
ErrorString Property	65
FontTypefaceList Property	65
FlagWhenIdle Property R/W	66
MapCharacterSet Property R/W	66
MapMode Property R/W	67
PageModeArea Property	68
PageModeDescriptor Property	69
PageModeHorizontalPosition Property R/W.....	70
PageModePrintArea Property R/W	71
PageModePrintDirection Property R/W.....	72
PageModeStation Property R/W	74
PageModeVerticalPosition Property R/W.....	75
RotateSpecial Property R/W.....	76
JrnLineChars Property R/W.....	77
JrnLineCharsList Property	78
JrnLineHeight Property R/W	79
JrnLineSpacing Property R/W	80
JrnLineWidthProperty	81
JrnLetterQuality Property R/W.....	81
JrnEmpty Property	81
JrnNearEnd Property.....	82
JrnCartridgeState Property	82
JrnCurrentCartridge Property R/W.....	82
RecLineChars Property R/W	83
RecLineCharsList Property.....	84
RecLineHeight Property R/W.....	85
RecLineSpacing Property R/W	86
RecLineWidth Property.....	87
RecLetterQuality Property R/W.....	87
RecEmpty Property	87

RecNearEnd Property	88
RecSidewaysMaxLines Property	88
RecSidewaysMaxChars Property	88
RecLinesToPaperCut Property	89
RecBarcodeRotationList Property	89
RecCartridgeState Property.....	90
RecCurrentCartridge Property R/W	90
RecBitmapRotationList Property.....	91
4.5. Common Methods.....	92
Open Method	92
Close Method	93
ClaimDevice Method	94
ReleaseDevice Method	95
CheckHealth Method.....	96
ClearOutput Method	97
CompareFirmwareVersion Method	98
DirectIO Method	99
ResetStatistics Method.....	102
RetrieveStatistics Method.....	103
UpdateFirmware Method	104
UpdateStatistics Method.....	104
4.6. Specific Methods.....	105
PrintNormal Method	105
PrintTwoNormal Method.....	106
PrintImmediate Method	107
BeginInsertion Method	108
EndInsertion Method	108
BeginRemoval Method	108
EndRemoval Method.....	109
CutPaper Method	109
RotatePrint Method	111
PrintBarcode Method.....	113
PrintBitmap Method.....	121
TransactionPrint Method	123
ValidateData Method.....	124
SetBitmap Method.....	126
SetLogo Method.....	128
ChangePrintSide Method	128
MarkFeed method	129
ClearPrintArea Method.....	129
PageModePrint Method.....	130
4.7. Events	133
DirectIOEvent Event.....	133

ErrorEvent Event	134
OutputCompleteEvent Event	134
StatusUpdateEvent Event	135
Chapter5 : OPOS INTERFACE SPECIFICATION (Cash Drawer)	136
5.1. Summary	136
5.2. Common Properties	138
BinaryConversion Properties R/W	138
CapCompareFirmwareVersion Property	139
CapPowerReporting Property	139
CapStatisticsReporting Property	139
CapUpdateFirmware Property	139
CapUpdateStatistics Property	140
CheckHealthText Property	140
Claimed Property	141
ControlObjectDescription Property	141
ControlObjectVersion Property	142
DeviceDescription Property	142
DeviceEnabled Property R/W	143
DeviceName Property	144
FreezeEvents Property R/W	144
OpenResult Property	145
PowerNotify Property R/W	146
PowerState Property	147
ResultCode Property	148
ResultCodeExtended Property	149
ServiceObjectDescription Property	150
ServiceObjectVersion Property	150
State Property	150
5.3. Common Methods	151
CheckHealth Method	151
ClaimDevice Method	152
Close Method	153
CompareFirmwareVersion Method	153
DirectIO method	153
Open Method	154
ReleaseDevice Method	155
ResetStatistics Method	155
RetrieveStatistics Method	155
UpdateFirmware Method	156
UpdateStatistics Method	156
5.4. Specific Properties	157
CapStatus Property	157

CapStatusMultiDrawerDetect Property	157
DrawerOpened Property.....	157
5.5. Specific Methods	158
OpenDrawer Method	158
WaitForDrawerClose Method	158
5.6. Events	159
DirectIOEvent Event.....	159
StatusUpdateEvent Event	159
Chapter6 : Registry for OCX.....	160
6.1. POS Printer	160
6.2. Cash Drawer	163
Chapter7 : Header File	165
7.1. POS Printer Header File	165

Chapter1: OVERVIEW

The IFD Series POS Printer and the Cash Drawer OPOS Control (hereinafter called "IFD OPOS Control") conform to OPOS 1.9 POS Printer Device and Cash Drawer Device and Control the IFD Series POS Printer manufactured by Seiko Instruments Inc. (hereinafter called "IFD").

Refer to "Appendix A OLE for Retail POS - OPOS Implementation Reference" in "UnifiedPOS Retail Peripheral Architecture Version 1.9"(hereinafter called UPOS Ver. 1.9) for details. And refer to "Appendix A - Section 4: OPOS Application Header File" in "UPOS Ver. 1.9."

The most up-to-date header files can be downloaded from the following web site:

<http://monroecs.com/opos.htm>

For additional information, the vendor-specific values of the IFD OPOS Control are defined additionally. For these values, refer to chapter 7 "Header File" in this guide. In this guide, "OPOS Control" and "OPOS OCX" have the same meanings.

1.1. Scope of This Document

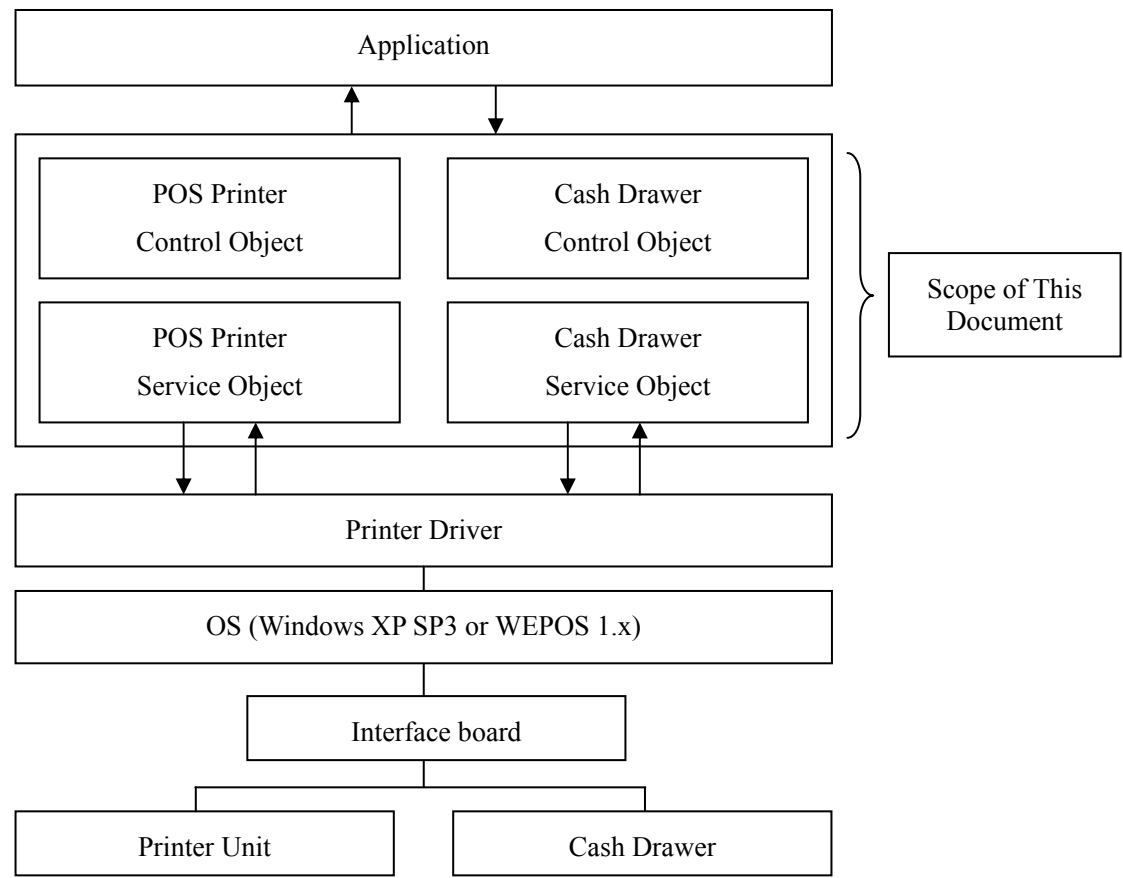
This specification (interface specification) is intended for reference mainly for application developers who use the IFD OPOS Control and describes the following items, which are necessary for them.

- How to install the IFD OPOS Control
- How to use the IFD OPOS Control
- Limitation of the IFD OPOS Control
- Explanation of the interface (property/method/event) of the IFD OPOS Control
- Explanation of the setting items of the IFD OPOS Control

1.2. Overview of OPOS Control

(1) Structural Drawing of OPOS Control

This OPOS Control adheres to the ActiveX Control specifications and exposes the properties, methods, and events to an Application. The Control cannot be seen on the UI when an application is executed and only the application that uses it requests the process through methods or properties. Responses are issued to the application through method return values and parameters, properties, and events. This OPOS Control is implemented as an in-process server. This OPOS Control can only control one POS Printer connected to an interface board connected to a POS (PC) by RS-232C, parallel cable or USB cable, and one Cash Drawer connected to the Interface board.



(2) Definitions

- **Control Object (CO)**
A Control Object exposes a set of properties, methods, and events to an application for its device class. This guide describes these APIs.
- **Service Object (SO)**
A Service Object is called by a Control Object to implement the OPOS prescribed functionality for a specific device.

1.3. Limitation (POS Printer, Cash Drawer)

(1) POS Printer

All interfaces of OPOS POS Printer device are provided, with the following limitations.

1. The method and property settings related to Slip print are not supported.
2. Italic, Custom Color, Shading and cartridge function are not supported.
3. All the following methods always return OPOS_E_ILLEGAL(106) after they are enabled.
 - BeginInsertion** method
 - EndInsertion** method
 - BeginRemoval** method
 - EndRemoval** method
 - ChangePrintSide** method
 - MarkFeed** method
 - PrintTwoNormal** method
4. If the registry "ProcessCompletionTiming" is set in "1", "Execution Response Request" command is used to control the print operation.
Do not include "Execution Response Request" command in the binary data following "Pass through embedded data (ESC|#E)" by Escape Sequence.
5. When an error occurs, hardware reset is transmitted by OCX to clear the remaining print data.
However, a part of the remaining print data might be printed depending on the timing of the error release and the hardware reset transmission on communication lines.

(2) Cash Drawer

OPOS Cash Drawer is available after setting the function of Cash Drawer to enable by the Function Settings in the interface board. See the technical reference of the interface board for the Function Settings. All interfaces of OPOS Cash Drawer device are provided, with the following limitations.

1. All the following methods always return OPOS_E_ILLEGAL(106) after they are enabled.
 - DirectIO** method
 - WaitForDrawerClose** method
2. **DirectIOEvent** event (Device-specific event)
Not supported.
3. **DrawerOpened** property, **StatusUpdateEvent** event
When **CapStatus** property is FALSE, the state change of the Cash Drawer cannot be reported.

Chapter2: INSTALLATION

Follow the procedure below for the installation of IFD OPOS Control.

2.1. Conditions of Installation

- Operating Environment
 - OS Microsoft Windows XP SP3 / WEPOS
 - CPU Recommend Pentium 3 550 MHz or faster
 - Memory Recommend 128 MB or more
 - HDD Free memory of 2 MB or more

2.2. Installation Media

Installation media is provided in CD-ROM etc.

CD-ROM is constructed as below.

```
\ (root)
\Install --- Installer
\Disk1  --- Installation contents such as Setup.exe.
```

2.3. Default Setting for Installation

The default settings at Installation are saved in the SetConfig.ini file at \Disk 1. The default settings for IFD OPOS Control can be changed by running Setup.exe after making changes in the SetConfig.ini File. These values are reflected in the Registry where IFD OPOS Control refers.

Key, Value Name	Default	Detail
[Installation]		
TargetFolder	\OPOS\SII\	This is a default install path for the OPOS Control and should be full path under the System drive. For silent installation, the OPOS Control is installed in this folder.
[POSPrinterSetupInformation]		
DefaultPOSPrinter	IFD00x	Default logic device name
[POSPrinterSetupInformationIFD00x]		
Version	1.9.9	Version of OPOS
Description	SII POS Printer Service Object, Copyright (C) 2010 Seiko Instruments Inc.	Details of POS Printer Service Object
LogFileName	""	Log file name
LogLevel	-1	Output log level -1: No output 0: Error 1: Warning 2: Information 3: Debug 4: Trace
LogFileSize	0	Maximum size of log (KB)
AutoCutter	1	Autocutter function 0: Enable 1: Disable
CurrentStation	2	POS Printer to be used. 1: Journal 2: Receipt
DriverName	SII IFD00x (2inch)	Printer driver name
MCAutoSave	1	Storing of maintenance counter 0: Disable 1: Enable
NearEnd	1	Paper-near-end sensor function 0: Enable 1: Disable

PeripheralDevice	2	Peripheral device selection 0: Reserved 1: Drawer is enable (when IFD501 is used) 2: Drawer is enable (when IFD001 is used) 3: Reserved
SendTimeout	10000	Specify the send timeout (millisecond).
ReceiveTimeout	10000	Specify the receive timeout (millisecond).
DataRegistrationTimeout	30000	Specify the user area reduction timeout (millisecond).
ProcessCompletionTiming	1	Timing for process completion 0: Timing at completion of data transmission 1: Timing at completion of the data printing
[POSPrinterSetupInformationIFD50x]		
DriverName	SII IFD50x (2inch)	Printer driver name
PeripheralDevice	1	Peripheral device selection
The values other than that for DriverName and PeripheralDevice are same as those for the [POSPrinterSetupInformationIFD00x] section.		
[POSPrinterSetupInformationPTD00]		
DriverName	SII PTD00 (2inch)	Printer driver name
The values other than that for DriverName are same as those for the [POSPrinterSetupInformationIFD00x] section.		
[POSPrinterSetupInformationPTD50]		
DriverName	SII PTD50 (2inch)	Printer driver name
PeripheralDevice	1	Peripheral device selection
The values other than that for DriverName and PeripheralDevice are same as those for the [POSPrinterSetupInformationIFD00x] section.		
[CashDrawerSetupInformation]		
DefaultCashDrawer	IFD00x	Default logic device name
[CashDrawerSetupInformationIFD00x]		
Version	1.9.6	Version of OPOS

Description	SII Cash Drawer Service Object, Copyright (C) 2010 Seiko Instruments Inc.	Details of Cash Drawer Service Object
LogFileName	""	Log file name
LogLevel	-1	Output log level -1: No output 0: Error 1: Warning 2: Information 3: Debug 4: Trace
LogFileSize	0	Maximum size of log (KB)
DriverName	SII IFD00x (2inch)	Printer driver name
OnTimer	50	Pulse on time of drawer kick
OffTimer	500	Pulse off time of drawer kick
InvertDrawerStatus	F	Synchronization of drawer sensor status and cash drawer status F: When the drawer sensor status is "High", the cash drawer is open. T: When the drawer sensor status is "Low", the cash drawer is open.
[CashDrawerSetupInformationIFD50x]		
DriverName	SII IFD50x (2inch)	Printer driver name
The values other than that for DriverName are same as those for the [CashDrawerSetupInformationIFD00x] section.		
[CashDrawerSetupInformationPTD00]		
DriverName	SII PTD00 (2inch)	Printer driver name
The values other than that for DriverName are same as those for the [CashDrawerSetupInformationIFD00x] section.		
[CashDrawerSetupInformationPTD50]		
DriverName	SII PTD50 (2inch)	Printer driver name
The values other than that for DriverName are same as those for the [CashDrawerSetupInformationIFD00x] section.		

2.4. Installation Procedure

For the installation procedure, there are the following two modes.

- Manual mode
- Silent mode

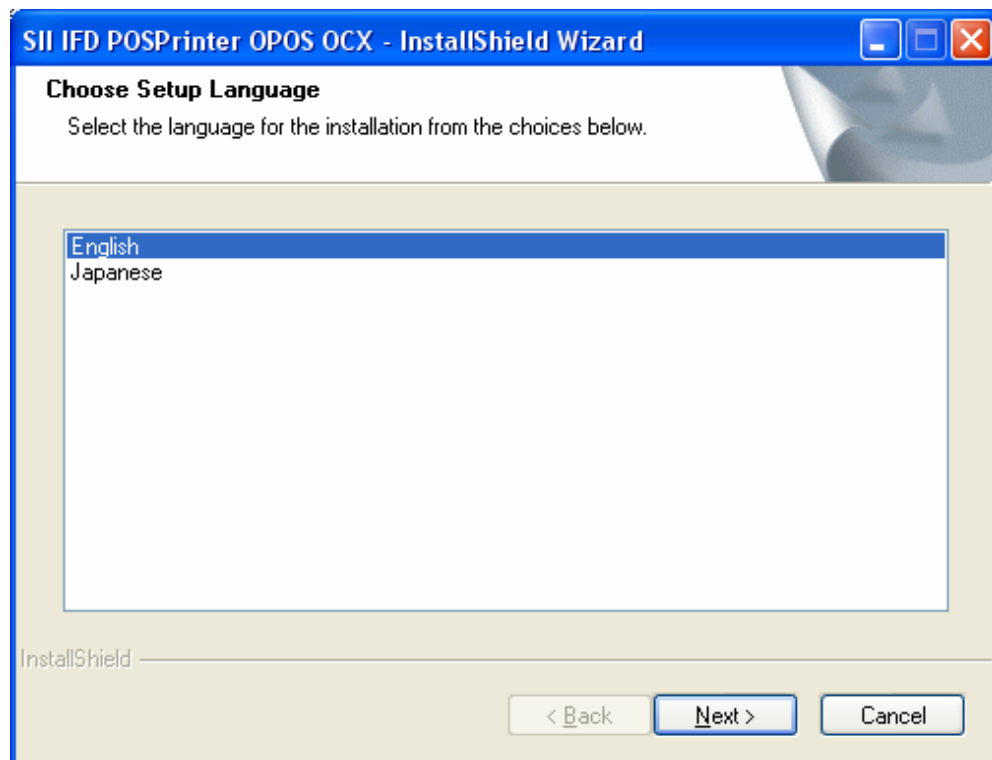
Each mode allows "Modify," "Repair" and "Remove."

"Modify" and "Repair" are same function. If the data is already registered, the data is overwritten. If not, the data is installed newly.

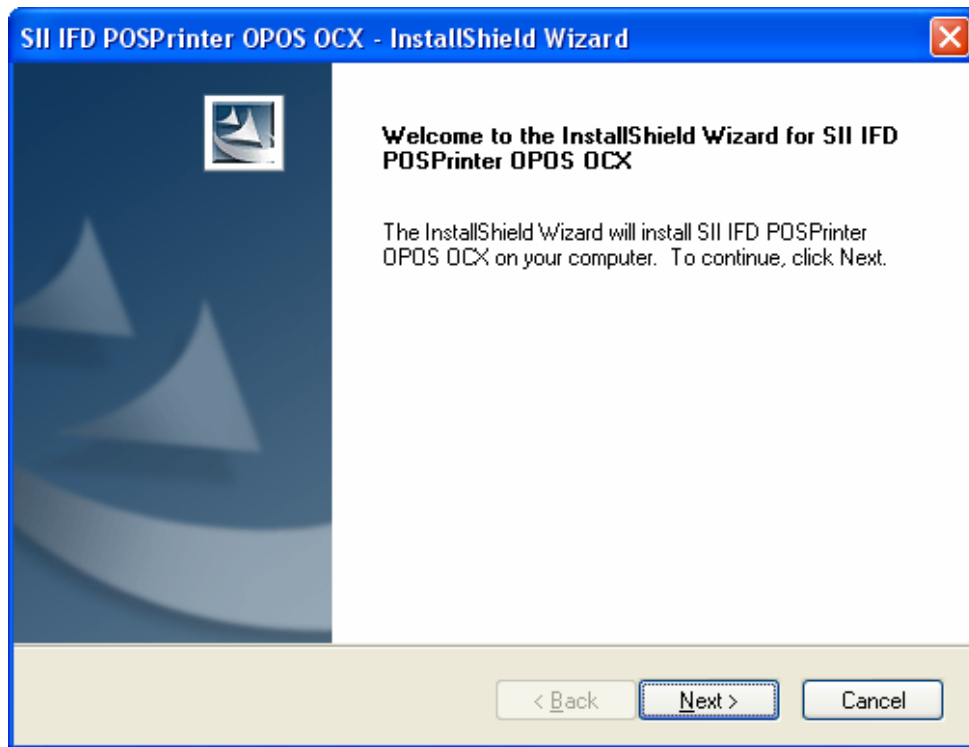
Manual mode and Silent mode are explained separately.

(1) Manual mode

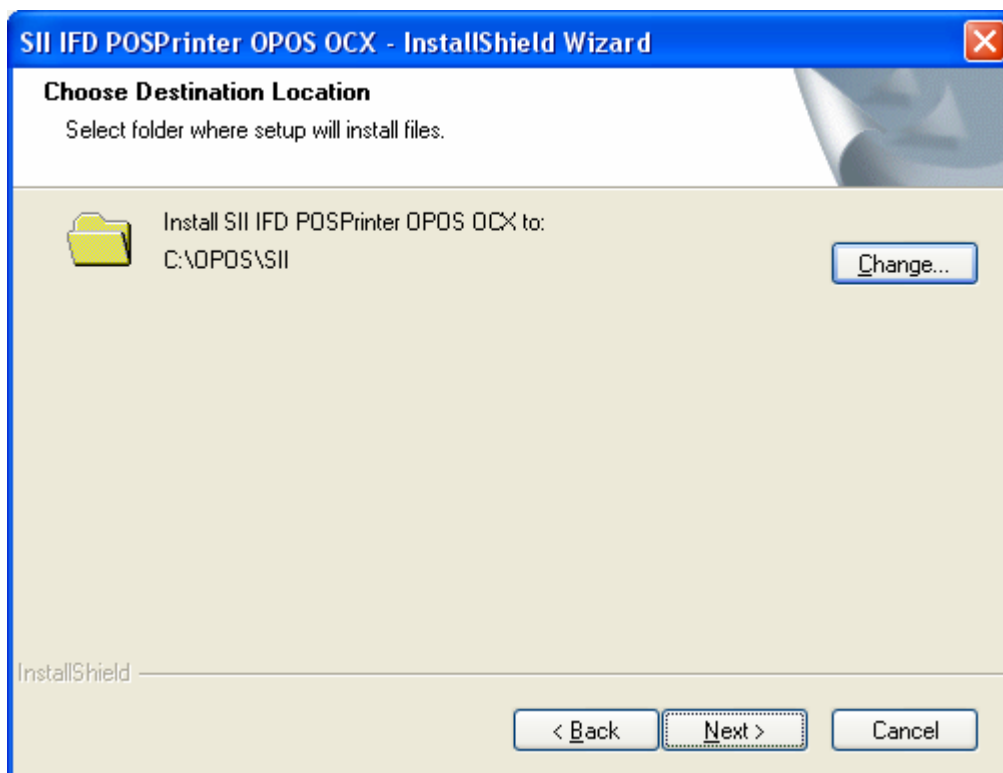
1. Insert the Installation media, CD-ROM into the drive. Then start "My computer" or "Explorer" and show the drive.
2. The dialog box below is shown and the setup program starts. Click "Next(N) >." Hereinafter the installation procedure is explained with the installer screen.



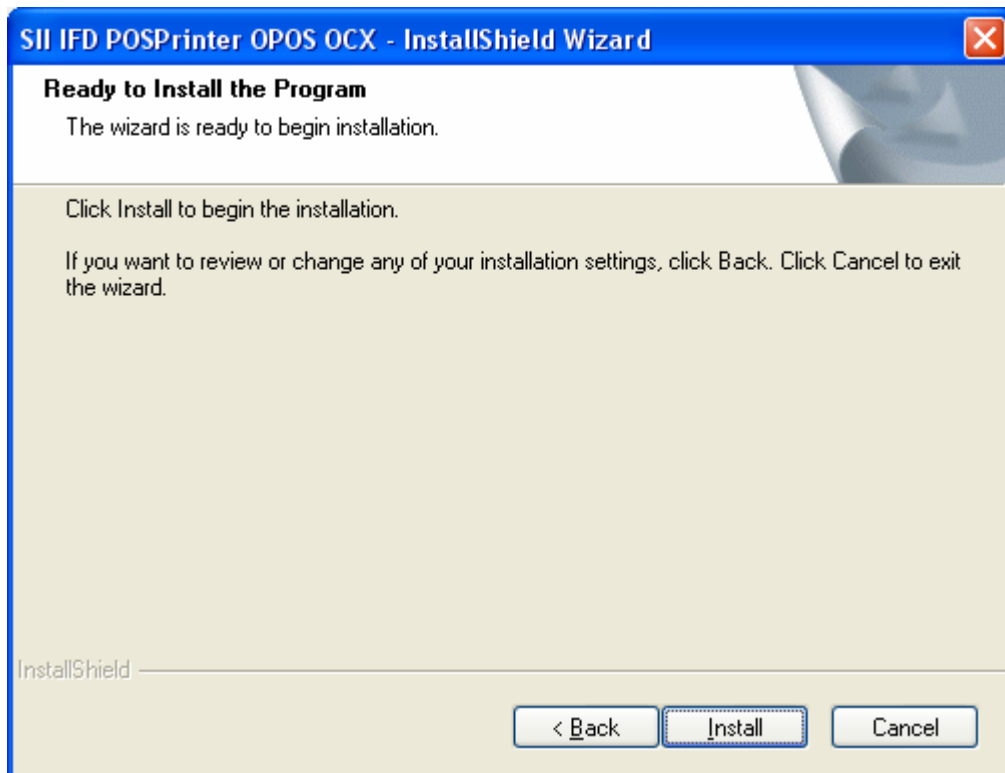
3. The dialog box below is shown. Click "Next(N) >."



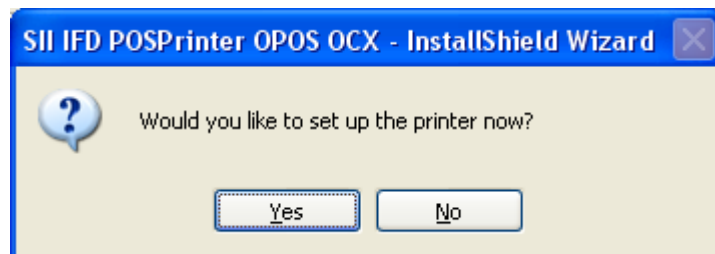
4. Specify the installation folder. Install folder is located in [System drive]:\OPOS\SII by default. Click "Next>."



5. After selecting the place for installation, the dialog box below is shown. If you click "Install," the installation is started.



6. After data copy is completed, select whether the configuration program is to be started for setting POS Printer.



7. When the configuration program starts, the dialog box below is shown. If the "OK" button is pressed, the setting value is written into the registry. The "Cancel" button cancels the configuration program. The configuration program can also be started from the OS Start menu, [Start] - [All Programs] - [SII] - [SII IFD POSPrinter OPOS OCX] - [SetupTool].

SetupTool

Common

DeviceName: IFD00x

Add Remove

IFD00x IFD50x PTD00 PTD50

SettingDriver | Sensor | Station | Drawer

SendTimeout(ms): 10000

ReceiveTimeout(ms): 10000

DataRegistrationTimeout(ms): 30000

ProcessCompletionTiming: Data transmitted Printed

PrinterDriver: SII IFD00x (2inch) SW DIP

LogicalDeviceName: Change LogicalDeviceName IFD00x

OK Cancel

8. The installation is completed. Click "Finish" to exit the installation wizard. Restart the PC after the installation.

SII IFD POSPrinter OPOS OCX - InstallShield Wizard

InstallShield Wizard Complete

The InstallShield Wizard has successfully installed SII IFD POSPrinter OPOS OCX. Click Finish to exit the wizard.

< Back Finish Cancel

(2) Silent mode

1. Copy the \Disk1 folder of installation media (such as CD-ROM) to any folder of a PC to be installed.
2. Start the following command from the application, which starts the installer.

```
..\ \ Disk1\Setup.exe -s -f1respond_file_path -f2result_file_path
```

Depending on the specified Respond file, the installer operates as below.

Respond File (iss)	Operation of Installer
setup.iss	New Installation
setup_change.iss	Modification
setup_repair.iss	Repair
setup_delete.iss	Removal

Specify the folder where Setup.exe is stored for the Respond file path.

* Do not execute "Remove (setup_delete.iss)" at initialization state or after deletion is completed.

3. Installation starts. The installation folder is set to the folder below by default.

```
[System drive]:\OPOS\SII
```

4. When command completes the process, the log file, which records the installation result, is output in the specified result file path. The example is shown as below. The result is recorded in ResultCode. Restart the PC after the installation.

```
[InstallShield Silent]
Version=v7.00
File=Log File
[ResponseResult]
ResultCode=0
[Application]
Name=SII IFD POSPrinter OPOS OCX
Version=1.00.00
Company=SII
Lang=0011
```

Value of ResultCode	Meaning
0	Success
-1	General error
-2	Invalid mode
-3	No necessary data in Respond file (iss file).
-4	Shortage of memory
-5	File cannot be found.
-6	Respond file cannot be written.
-7	Log file cannot be written.
-10	Data type is invalid.
-11	Unknown error during setup.
-12	Dialog is not available.
-51	Specified folder cannot be created.
-52	Specified file or folder cannot be accessed.
-53	Invalid option is selected.

2.5. Uninstallation Procedure

Open the control panel and execute "Add or Remove Programs."

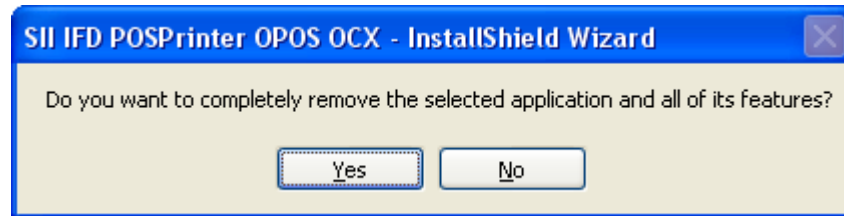
Then select "SII IFD POSPrinter OPOS OCX" and click "Modify" or "Remove."

For the uninstallation procedure in silent mode, refer to the installation procedure. (In this case, specify the respond file for uninstallation "setup_delete.iss.")

- When selecting "Modify"
 1. The dialog to select "Modify," "Repair" and "Remove" is indicated. Select "Modify" or "Repair" for modifying OCX and installing and "Remove" for uninstallation, and click "Next>."
 - * "Modify" and "Repair" execute same process.
 2. Selected process is executed.
 3. For uninstallation, the files which cannot be deleted by the uninstaller may remain in the installed folder ([System drive]:\OPOS\SII\). Please delete them manually.

- When selecting "Remove"

1. The dialog box is displayed to confirm uninstallation. Click "Yes"



2. Uninstallation is executed.

3. The files that cannot be deleted by the uninstall process may remain in the installed folder ([System drive]:\OPOS\SII\). Please delete them manually.

That completes the uninstallation process.

2.6. Installation File List

IFD OPOS Control files are located as below.

[System drive]:\OPOS\SII\	
POSPrinterCO109.ocx	Control object for POS Printer
POSPrinterSO_IFD.dll	Service object for POS Printer
CashDrawerCO109.ocx	Control object for Cash Drawer
CashDrawerSO_IFD.dll	Service object for Cash Drawer
SetupTool_IFD.exe	Configuration program

Chapter3: OPOS CONTROL

3.1. General

The application uses the OPOS Control according to the procedure below.

1. **Open** method: Call to link the Control Object to the Service Object.
2. **ClaimDevice** method: Call to gain exclusive access to the device.
3. **DeviceEnabled** method: Set to TRUE to make the device operational.
4. Use the device (Each property, method, event).
5. **DeviceEnabled** property: Set to FALSE to disable the device.
6. **ReleaseDevice** method: Call to release exclusive access to the device.
7. **Close** method: Call to release the Service Object from the Control Object.

For more information on the usage, refer to the document "UPOS Ver. 1.9."

3.2. POS Printer

POS Printer supports only "Receipt" and "Journal." For the method and property related to Slip, the interface is provided but the performance is not supported. POS Printer supports synchronous/asynchronous output based on the general output model. Also, POS Printer is the device used exclusively.

Chapter4: OPOS INTERFACE SPECIFICATION (POS Printer)

4.1. Summary

Common Properties

Property Name	Type	Access	Availability Condition	Initial Value after Open
BinaryConversion	Long	R/W	Open	OPOS_BC_NONE (0)
CapCompareFirmwareVersion	Boolean	R	Open	FALSE
CapPowerReporting	Long	R	Open	OPOS_PR_STANDARD (1)
CapStatisticsReporting	Boolean	R	Open	FALSE
CapUpdateFirmware	Boolean	R	Open	FALSE
CapUpdateStatistics	Boolean	R	Open	FALSE
CheckHealthText	String	R	Open	""
Claimed	Boolean	R	Open	FALSE
DeviceEnabled	Boolean	R/W	Open & Claim	FALSE
FreezeEvents	Boolean	R/W	Open	FALSE
OpenResult	Long	R	--	OPOS_SUCCESS(0)
OutputID	Long	R	Open	0
PowerNotify	Long	R/W	Open	OPOS_PN_DISABLED (0)
PowerState	Long	R	Open	OPOS_PS_UNKNOWN (2000)
ResultCode	Long	R	--	OPOS_SUCCESS(0)
ResultCodeExtended	Long	R	Open	0
State	Long	R	--	OPOS_S_IDLE (2)
ControlObjectDescription	String	R	--	"SII POS Printer Control Object, Copyright (C) 2009 Seiko Instruments Inc."
ControlObjectVersion	Long	R	--	1009004
ServiceObjectDescription	String	R	Open	"SII IFD00x (2inch) POS Printer Service Object, Copyright (C) 2010 Seiko Instruments Inc."*2
ServiceObjectVersion	Long	R	Open	1009009
DeviceDescription	String	R	Open	"SII IFD00x (2inch) POS Printer" *2
DeviceName	String	R	Open	"IFD00x (2inch) POS Printer" *2

Specific Properties

Property Name	Type	Access	Availability Condition	Initial Value after Open
CapCharacterSet	Long	R	Open	PTR_CCS_KANJI (11)
CapCoverSensor	Boolean	R	Open	TRUE
CapMapCharacterSet	Boolean	R	Open	FALSE
CapTransaction	Boolean	R	Open	TRUE
CapJrnPresent	Boolean	R	Open	TRUE
CapJrn2Color	Boolean	R	Open	FALSE
CapJrnBold	Boolean	R	Open	TRUE
CapJrnDhigh	Boolean	R	Open	TRUE
CapJrnDwide	Boolean	R	Open	TRUE
CapJrnDwideDhigh	Boolean	R	Open	TRUE
CapJrnEmptySensor	Boolean	R	Open	TRUE
CapJrnItalic	Boolean	R	Open	FALSE
CapJrnNearEndSensor	Boolean	R	Open	TRUE ^{*1}
CapJrnUnderline	Boolean	R	Open	TRUE
CapJrnCartridgeSensor	Long	R	Open	0
CapJrnColor	Long	R	Open	PTR_COLOR_PRIMARY (0x00000001)
CapRecPresent	Boolean	R	Open	TRUE
CapRec2Color	Boolean	R	Open	FALSE
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	TRUE ^{*1}
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	PTR_COLOR_PRIMARY (0x00000001)
CapRecMarkFeed	Long	R	Open	0
CapRecPageMode	Boolean	R	Open	TRUE
AsyncMode	Boolean	R/W	Open	FALSE
CartridgeNotify	Long	R/W	Open	PTR_CN_DISABLED (0)
CharacterSet	Long	R/W	Open, Claim, & Enable	999 ^{*1}
CharacterSetList	String	R	Open	"437,932,999,1252"
CoverOpen	Boolean	R	Open, Claim, & Enable	FALSE
ErrorLevel	Long	R	Open	PTR_EL_NONE (1)
ErrorStation	Long	R	Open	0
ErrorString	String	R	Open	""
FontTypefaceList	String	R	Open	""
FlagWhenIdle	Boolean	R/W	Open	FALSE
MapCharacterSet	Boolean	R/W	Open * Cannot be written	FALSE
MapMode	Long	R/W	Open	PTR_MM_DOTS (1)
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)
JrnLineChars	Long	R/W	Open, Claim, & Enable	36 ^{*2}
JrnLineCharsList	String	R	Open	"27,30,33,36,43,48,54" ^{*2}
JrnLineHeight	Long	R/W	Open, Claim, & Enable	24
JrnLineSpacing	Long	R/W	Open, Claim, & Enable	30

JrnLineWidth	Long	R	Open, Claim, & Enable	432 ^{*2}
JrnLetterQuality	Boolean	R/W	Open, Claim, & Enable	FALSE
JrnEmpty	Boolean	R	Open, Claim, & Enable	FALSE
JrnNearEnd	Boolean	R	Open, Claim, & Enable	FALSE
JrnCartridgeState	Long	R	Open, Claim, & Enable	PTR_CART_UNKNOWN (0x10000000)
JrnCurrentCartridge	Long	R/W	Open, Claim, & Enable	PTR_COLOR_PRIMARY (0x00000001)
RecLineChars	Long	R/W	Open, Claim, & Enable	36 ^{*2}
RecLineCharsList	String	R	Open	"27,30,33,36,43,48,54" ^{*2}
RecLineHeight	Long	R/W	Open, Claim, & Enable	24
RecLineSpacing	Long	R/W	Open, Claim, & Enable	30
RecLineWidth	Long	R	Open, Claim, & Enable	432 ^{*2}
RecLetterQuality	Boolean	R/W	Open, Claim, & Enable	FALSE
RecEmpty	Boolean	R	Open, Claim, & Enable	FALSE
RecNearEnd	Boolean	R	Open, Claim, & Enable	FALSE
RecSidewaysMaxLines	Long	R	Open, Claim, & Enable	14 ^{*2}
RecSidewaysMaxChars	Long	R	Open, Claim, & Enable	144 ^{*2}

RecLinesToPaperCut	Long	R	Open, Claim, & Enable	3
RecBarCodeRotationList	String	R	Open	"0,R90,L90,180"
RecCartridgeState	Long	R	Open, Claim, & Enable	PTR_CART_UNKNOWN (0x10000000)
RecCurrentCartridge	Long	R/W	Open, Claim, & Enable	PTR_COLOR_PRIMARY (0x00000001)
RecBitmapRotationList	String	R	Open	"0,R90,L90,180"

*1: Variable item depends on the registry value.

*2: Variable item depends on the printer driver to be used.

The initial value of the CapRec.....property or Rec.....property is the initial value of when the receipt(2) is selected by "CurrentStation" of the registry value.

The initial value of the CapJrn..... property or Jrn..... property is the initial value of when the journal(1) is selected by "CurrentStation" of the registry value.

The following properties for POS Printer are not supported.

CapConcurrentJrnRec	CapSlpItalic	SlpLineWidth
CapConcurrentJrnSlp	CapSlpLeft90	SlpLetterQuality
CapConcurrentRecSlp	CapSlpNearEndSensor	SlpEmpty
CapConcurrentPageMode	CapSlpRight90	SlpNearEnd
CapSlpPresent	CapSlpRotate180	SlpSidewaysMaxLines
CapSlpFullslip	CapSlpUnderline	SlpSidewaysMaxChars
CapSlp2Color	CapSlpBothSidesPrint	SlpMaxLines
CapSlpBarCode	CapSlpCartridgeSensor	SlpLinesNearEndToEnd
CapSlpBitmap	CapSlpColor	SlpBarCodeRotationList
CapSlpBold	CapSlpPageMode	SlpPrintSide
CapSlpDhigh	SlpLineChars	SlpCartridgeState
CapSlpDwide	SlpLineCharsList	SlpCurrentCartridge
CapSlpDwideDhigh	SlpLineHeight	SlpBitmapRotationList
CapSlpEmptySensor	SlpLineSpacing	

Common Methods

Method Name	Availability Condition
Open	--
Close	Open
ClaimDevice	Open
ReleaseDevice	Open & Claim
CheckHealth	Open, Claim, & Enable
ClearOutput	Open, Claim, & Enable ^{*1}
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 Methods

Method Name	Availability Condition
PrintNormal	Open, Claim, & Enable
PrintTwoNormal	Open, Claim, & Enable
PrintImmediate	Open, Claim, & Enable
BeginInsertion	Open, Claim, & Enable
EndInsertion	Open, Claim, & Enable
BeginRemoval	Open, Claim, & Enable
EndRemoval	Open, Claim, & Enable
CutPaper	Open, Claim, & Enable
RotatePrint	Open, Claim, & Enable
PrintBarCode	Open, Claim, & Enable
PrintBitmap	Open, Claim, & Enable
TransactionPrint	Open, Claim, & Enable
ValidateData	Open, Claim, & Enable
SetBitmap	Open, Claim, & Enable
SetLogo	Open, Claim, & Enable
ChangePrintSide	Open, Claim, & Enable
MarkFeed	Open, Claim, & Enable
ClearPrintArea	Open, Claim, & Enable
PageModePrint	Open, Claim, & Enable

Events

Event Name	Occurrence Condition
DirectIOEvent	Open, Claim, & Enable ^{*1}
ErrorEvent	Open, Claim, & Enable
OutputCompleteEvent	Open, Claim, & Enable
StatusUpdateEvent	Open, Claim, & Enable

*1: Item for which the necessary condition differs from that in "UnifiedPOS Retail Peripheral Architecture, Ver. 1.9."

4.2. Data Characters and Escape Sequences

(1) Escape Sequence operated when specified

Name	Data	Remarks
Paper cut	ESC #P	Cuts receipt paper. The character '#' is replaced by an ASCII decimal string indicating the percentage cut desired. If 100 is specified, then a full cut is performed. If 1 to 99 is specified, a partial cut is performed. If '#' is omitted, a full cut is performed. If '#' is out of 1 - 100, this is ignored. This is ignored during rotated 90° right/left mode by RotatePrint method or during page mode by PageModePrint method. This is also ignored when PTR_S_JOURNAL(1) is specified for the <i>Station</i> parameter.
Feed and Paper cut	ESC #fP	Cuts receipt paper, after feeding the paper by the RecLinesToPaperCut lines. The character '#' is defined by the "Paper cut" escape sequence. If '#' is out of 1 - 100, this is ignored. This is ignored during rotated 90° right/left mode by RotatePrint method or during page mode by PageModePrint method. This is also ignored when PTR_S_JOURNAL(1) is specified for the <i>Station</i> parameter.
Feed, Paper cut, and Stamp print	ESC #sP	Not supported.
Bitmap print	ESC #B	Prints the pre-stored bitmap. The character '#' is replaced by the bitmap number. If '#' is omitted, the data is understood as print data instead of escape sequence and OPOS_SUCCESS(0) is returned when the ValidateData method is used. A value of 1 - 20 can be specified for '#'. When journal is specified, this escape sequence is invalid because bitmap cannot be registered.
Print top logo	ESC tL	Prints the pre-stored top logo.
Print bottom logo	ESC bL	Prints the pre-stored bottom logo.
Print stamp	ESC sL	Not supported.
Feed lines	ESC #fF	Feed the paper forward by lines. The character '#' is replaced by an ASCII decimal string indicating the number of lines to be fed. If '#' is omitted, then one line is fed. A value of 0 - 255 can be specified for '#'. If '#' exceeds this range, feeding lines is not performed. This is ignored during rotated 90°right/left mode by RotatePrint method.

Feed units	ESC #uF	Feed the paper forward by units in MapMode . The character '#' is replaced by an ASCII decimal string indicating the number of units to be fed. If '#' is omitted, then one unit is fed. If the MapMode is selected in PTR_MM_DOTS(1), '#' is available from 1 to 255. If the '#' exceeds this range, feeding units is not performed. This is ignored during rotated 90°right/left mode by RotatePrint method.
Feed reverse	ESC #rF	Not supported.
Pass through embedded data	ESC #E	Send the following "#E" characters of data through to the hardware without modifying it. The character '#' is replaced by an ASCII decimal string indicating the number of bytes following the escape sequence that should be passed through as-is to the hardware. If '#' is omitted, the data is understood as print data instead of escape sequence and OPOS_SUCCESS(0) is returned when the ValidateData method is used. '#' is available from 1 to 65535. If the '#' exceeds this range, transmission of embedded data is not performed. If the print data specified by '#' is not set after the escape sequence is specified, the available print data to send is only sent (Example: If ESC 2E'a' is specified, only 'a' is sent since only 1 byte is set for the character string). Also, during rotated 90° right/left mode by RotatePrint method, the width cannot be calculated exactly because data string specified by transmission of embedded data is not counted as character string. Therefore, make an appropriate adjustment by inserting blanks.

(2) Escape Sequence operated during printing

It is characteristic that the state is kept until it is changed explicitly.

Name	Data	Remarks
Font typeface	ESC #fT	Not supported.

(3) Escape Sequence operated when printing

It is characteristic that it is reset at the end of each print method or by a "Normal" sequence.

Name	Data	Remarks
Bold	ESC bC	Prints in bold.
Underline	ESC #uC	Prints with underline. The character '#' is replaced by an ASCII decimal string indicating the thickness of the underline in printer dot units. The available thickness is from 0 to 2. If '#' is omitted, then a thickness of 1 is used. If '#' is 3 or larger, then a thickness of 2 is used. The thickness of the underline can be selected only one type per line. Also, the thickness of the underline specified at the end of the line has priority. If the ESC N is placed after the underline escape sequence, the underline is not indicated at the line since ESC N escape sequence specifies a thickness of 0.
Italic	ESC iC	Not supported.
Custom color	ESC #rC	Not supported.
Red color	ESC rC	Not supported.
Reverse video	ESC rvC	Prints in a reverse video format.
Shading	ESC #sC	Not supported.
Single high and wide	ESC 1C	Prints normal size.
Double wide	ESC 2C	Prints double-wide characters.
Double high	ESC 3C	Prints double-high characters.
Double high and wide	ESC 4C	Prints double-high/double-wide characters.
Scale horizontally	ESC #hC	Supports '#' from 1 to 8. If '#' is omitted, the data is understood as print data instead of escape sequence and OPOS_SUCCESS(0) is returned when the ValidateData method is used.
Scale vertically	ESC #vC	Supports '#' from 1 to 8. If '#' is omitted, the data is understood as print data instead of escape sequence and OPOS_SUCCESS(0) is returned when the ValidateData method is used.
RGB Color	ESC #fC	Not supported.

Center	ESC cA	Aligns following text in the center. This must be specified at the head of the line. If not, this is invalid. Also, if there is a linefeed on the print data, the center is valid after linefeed. This specification is ignored during rotated 90° right/left mode by RotatePrint method or during page mode by PageModePrint method.
Right justify	ESC rA	Aligns the subsequent texts to the right. This must be specified at the head of the line. If not, this is invalid. Also, if there is a linefeed on the print data, the right justification valid after linefeed. This specification is ignored during rotated 90° right/left mode by RotatePrint method or during page mode by PageModePrint method.
Normal	ESC N	Restores printer characteristics to normal condition.
SubScript	ESC tbC	Not supported.
SuperScript	ESC tpC	Not supported.

4.3. Common Properties

BinaryConversion Property R/W

Syntax **LONG BinaryConversion;**

Remarks OPOS passes multi-character input and output using BStrings. BStrings may be safely used for text data. As the BStrings are passed between the application and the OPOS Control, OLE may perform language-specific translations to or from Unicode. When BStrings are used to pass binary data, then these translations may alter the data such that the data byte in a BString character at the application does not match the corresponding byte at the Control. This mismatch is more likely when BString pointers are used, since the Unicode characters are presented to the application and/or Control, and a language difference between them may cause misinterpretation.

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

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

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

The **BinaryConversion** values are:

Value	Meaning
OPOS_BC_NONE(0)	Data is placed one byte per BString character, with no conversion.
OPOS_BC_NIBBLE(1)	Each byte is converted into two characters. (This option provides for the fastest conversion between binary and ASCII characters.) First character = 0x30 + bits 7-4 of the data byte. Second character = 0x30 + bits 3-0 of the data byte. Example: Byte value 154 = 0x9A is converted into the characters 0x39 0x3A (= the string "9:"). Note that this conversion is not the more common hexadecimal ASCII, which would have converted 154 to 0x39 0x41 (= the string "9A").
OPOS_BC_DECIMAL(2)	Each byte is converted into three characters. VAL (string) may be used on each 3 characters to convert from ASCII to binary. RIGHT("^^"+STR(byte), 3) may be used to produce 3 ASCII characters from each byte, where '^' represents the space character. Example 1: Byte value 154 = 0x9A becomes the characters 0x31 0x35 0x34 (= the string "154"). Example 2: Byte value 8 = 0x08 becomes the characters 0x30 0x30 0x38 (= the string "008").

When **BinaryConversion** is on (that is, not OPOS_BC_NONE(0)) and the property or method parameter description specifies that **BinaryConversion** applies, before setting the property or passing the method parameter, convert the string data into the format specified by the **BinaryConversion** value.

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

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

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

CapCompareFirmwareVersion Property

Syntax **BOOL CapCompareFirmwareVersion;**

Remarks FALSE: The function that compares firmware versions is not supported.
This property is initialized to FALSE by the **Open** method.

CapPowerReporting Property

Syntax **LONG CapPowerReporting;**

Remarks Identifies the reporting capabilities of the device.

The **CapPowerReporting** values are:

Value	Meaning
OPOS_PR_STANDARD(1)	The Service Object can determine and report two of the power states – OFF_OFFLINE (that is, off or offline)and ONLINE.

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

CapStatisticsReporting Property

Syntax **BOOL CapStatisticsReporting;**

Remarks FALSE: No statistical data regarding the device is available.

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

CapUpdateFirmware Property

Syntax **BOOL CapUpdateFirmware;**

Remarks FALSE: Firmware update is not supported.

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

CapUpdateStatistics Property

Syntax	BOOL CapUpdateStatistics;
Remarks	FALSE: None of the statistical data can be reset/updated by the application. This property is initialized to FALSE by the Open method.

CheckHealthText Property

Syntax	BSTR CheckHealthText;
Remarks	<p>Holds the results of the most recent call to the CheckHealth method. The following examples illustrate the results of diagnosis.</p> <ul style="list-style-type: none">- For Internal Successful: "Internal HCheck: Not Support"- For External Successful: "External HCheck: Successful" Failed: "External HCheck: Error"- For Interactive Successful: "Interactive HCheck: Successful" * Failed: "Interactive HCheck: Error" <p>This value is initialized to an "(empty string)" before the first call to CheckHealth method.</p> <p>* In the case of Interactive, "Interactive HCheck: Successful." is set if the dialog box is closed without testing after the command is made.</p>

Claimed Property

Syntax	BOOL Claimed;
Remarks	TRUE: The device is claimed for exclusive access. FALSE: The device is released for sharing with other applications. The value of Claimed property is initialized to FALSE by the Open method.

DeviceEnabled Property R/W

Syntax	BOOL DeviceEnabled;	
Remarks	TRUE:	The device has been placed in an operational state. If changed to TRUE, then the device is brought to an operational state.
	FALSE:	The device has been disabled. If changed to FALSE, then the device is disabled.
	The application must set this property to TRUE before using output devices.	
	This property is initialized to FALSE by the Open method.	
Return	When this property is set, one of the following values is placed in the ResultCode property.	
	Value	Meaning
	OPOS_SUCCESS(0)	The property was set successfully.
	OPOS_E_NOTCLAIMED(103)	An exclusive use device must be claimed before the device may be enabled.
	OPOS_E_NOHARDWARE(107)	The device is not connected to the system or is not powered on.
	OPOS_E_FAILURE(111)	The device cannot be enabled. Setting information and the information from device may be different.
	OPOS_E_TIMEOUT(112)	The Service Object timed out waiting for a response from the device, or the data was unable to be transmitted to the device within the timeout.
	OPOS_E_BUSY(113)	Setting of property was failed due to processing. Set the property after process is completed.
	OPOS_E_EXTENDED(114)	ResultCodeExtended = OPOS_EPTR_COVER_OPEN(201): Cover is open. ResultCodeExtended = OPOS_EPTR_JRN_EMPTY (202): The journal is out of paper. ResultCodeExtended = OPOS_EPTR_REC_EMPTY (203): The receipt is out of paper.

FreezeEvents Property R/W

Syntax	BOOL FreezeEvents;				
Remarks	<p>TRUE: The application has requested that the Control not deliver events. Events will be held by the Control until events are unfrozen.</p> <p>FALSE: The application allows events to be delivered. If some events have been held while events were frozen and all other conditions are correct for delivering the events, then changing FreezeEvents to FALSE will cause these events to be delivered.</p> <p>An application may choose to freeze events for a specific sequence of code where interruption by an event is not desirable.</p> <p>If an error occurs while the print method such as the PrintNormal method is operated under the AsyncMode property is TRUE, ErrorEvent is frozen and the State property turns to OPOS_S_BUSY(3). In this case, clear the frozen event by the ClearOutput method or conduct the Close method after ErrorEvent is occurred by setting FALSE since the control cannot be closed under this circumstance.</p> <p>This property is initialized to FALSE by the Open method.</p>				
Return	<p>When this property is set, the following value is placed in the ResultCode property:</p> <table><tr><th>Value</th><th>Meaning</th></tr><tr><td>OPOS_SUCCESS(0)</td><td>The property was set successfully.</td></tr></table>	Value	Meaning	OPOS_SUCCESS(0)	The property was set successfully.
Value	Meaning				
OPOS_SUCCESS(0)	The property was set successfully.				

OpenResult Property

Syntax	LONG OpenResult;														
Remarks	Holds additional details about the most recent Open method. The OpenResult values are: <table><tr><th>Value</th><th>Meaning</th></tr><tr><td>OPOS_SUCCESS(0)</td><td>Successful open.</td></tr><tr><td>OPOS_OR_ALREADYOPEN(301)</td><td>Control already open.</td></tr><tr><td>OPOS_OR_REGBADNAME(302)</td><td>The registry does not contain a key for the specified device name. Or, the device name is not specified.</td></tr><tr><td>OPOS_OR_REGPROGID(303)</td><td>Could not read the device name key's default value, or could not convert the Programmatic ID it holds into a valid Class ID.</td></tr><tr><td>OPOS_OR_CREATE(304)</td><td>Could not create a service object instance, or could not get its IDispatch interface.</td></tr><tr><td>OPOS_OR_BADIF(305)</td><td>The service object does not support one or more of the methods required by its release. The setting of device name may be different from the Service Object.</td></tr></table>	Value	Meaning	OPOS_SUCCESS(0)	Successful open.	OPOS_OR_ALREADYOPEN(301)	Control already open.	OPOS_OR_REGBADNAME(302)	The registry does not contain a key for the specified device name. Or, the device name is not specified.	OPOS_OR_REGPROGID(303)	Could not read the device name key's default value, or could not convert the Programmatic ID it holds into a valid Class ID.	OPOS_OR_CREATE(304)	Could not create a service object instance, or could not get its IDispatch interface.	OPOS_OR_BADIF(305)	The service object does not support one or more of the methods required by its release. The setting of device name may be different from the Service Object.
Value	Meaning														
OPOS_SUCCESS(0)	Successful open.														
OPOS_OR_ALREADYOPEN(301)	Control already open.														
OPOS_OR_REGBADNAME(302)	The registry does not contain a key for the specified device name. Or, the device name is not specified.														
OPOS_OR_REGPROGID(303)	Could not read the device name key's default value, or could not convert the Programmatic ID it holds into a valid Class ID.														
OPOS_OR_CREATE(304)	Could not create a service object instance, or could not get its IDispatch interface.														
OPOS_OR_BADIF(305)	The service object does not support one or more of the methods required by its release. The setting of device name may be different from the Service Object.														
	This property is initialized to OPOS_SUCCESS(0) by the Open method.														

OutputID Property

Syntax	LONG OutputID;
Remarks	Holds the identifier of the most recently started asynchronous output. (call to an asynchronous method when the AsyncMode property is set to TRUE). When a method successfully initiates an asynchronous output, the Control assigns an identifier to the request. When the output completes, the Control will fire an OutputCompleteEvent passing this output ID as a parameter. Output ID is numbered from 1 to 99999 cyclically.

PowerNotify Property R/W

Syntax	LONG PowerNotify;						
Remarks	<p>Contains the type of power notification selection made by the Application. The PowerNotify values are:</p> <table><tr><th>Value</th><th>Meaning</th></tr><tr><td>OPOS_PN_DISABLED(0)</td><td>The Control will not provide any power notifications to the application. No power notification StatusUpdateEvents will be fired, and PowerState may not be set.</td></tr><tr><td>OPOS_PN_ENABLED(1)</td><td>The Control will fire power notification StatusUpdateEvents and update PowerState, beginning when DeviceEnabled is set to TRUE. The level of functionality depends upon CapPowerReporting.</td></tr></table> <p>PowerNotify may only be set while the device is disabled; that is, while DeviceEnabled is FALSE.</p> <p>This property is initialized to OPOS_PN_DISABLED(0) by the Open method. This value provides compatibility with earlier releases.</p>	Value	Meaning	OPOS_PN_DISABLED(0)	The Control will not provide any power notifications to the application. No power notification StatusUpdateEvents will be fired, and PowerState may not be set.	OPOS_PN_ENABLED(1)	The Control will fire power notification StatusUpdateEvents and update PowerState , beginning when DeviceEnabled is set to TRUE. The level of functionality depends upon CapPowerReporting .
Value	Meaning						
OPOS_PN_DISABLED(0)	The Control will not provide any power notifications to the application. No power notification StatusUpdateEvents will be fired, and PowerState may not be set.						
OPOS_PN_ENABLED(1)	The Control will fire power notification StatusUpdateEvents and update PowerState , beginning when DeviceEnabled is set to TRUE. The level of functionality depends upon CapPowerReporting .						
Return	<p>When this property is set, one of the following values is placed in the ResultCode.</p> <table><tr><th>Value</th><th>Meaning</th></tr><tr><td>OPOS_SUCCESS(0)</td><td>The property was set successfully.</td></tr><tr><td>OPOS_E_ILLEGAL(106)</td><td>The device is already enabled.</td></tr></table>	Value	Meaning	OPOS_SUCCESS(0)	The property was set successfully.	OPOS_E_ILLEGAL(106)	The device is already enabled.
Value	Meaning						
OPOS_SUCCESS(0)	The property was set successfully.						
OPOS_E_ILLEGAL(106)	The device is already enabled.						

PowerState Property

Syntax **LONG** PowerState;

Remarks The **PowerState** values are:

Value	Meaning
OPOS_PS_UNKNOWN(2000)	Cannot determine the device's power state, for one of the following reasons: <ul style="list-style-type: none">• PowerNotify = OPOS_PN_DISABLED(0)• DeviceEnabled = FALSE
OPOS_PS_ONLINE(2001)	The device is powered on and ready for use.
OPOS_PS_OFF_OFFLINE(2004)	The device is either off or offline.

This property is initialized to OPOS_PS_UNKNOWN(2000) by the **Open** method.

ResultCode Property

Syntax **LONG ResultCode;**

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

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

The **ResultCode** values are:

Value	Meaning
OPOS_SUCCESS(0)	Successful operation.
OPOS_E_CLOSED(101)	Attempt was made to access a closed device.
OPOS_E_NOTCLAIMED(103)	Attempt was made to access an exclusive-use device that must be claimed before the method or property set action can be used.
OPOS_E_NOSERVICE(104)	The Control cannot communicate with the Service Object. Most likely, a setup or configuration error must be corrected.
OPOS_E_DISABLED(105)	Cannot perform operation while device is disabled.
OPOS_E_ILLEGAL(106)	Attempt was made to perform an illegal or unsupported operation with the device, or an invalid parameter value was used.
OPOS_E_NOHARDWARE(107)	The device is not connected to the system or is not powered on.
OPOS_E_NOEXIST(109)	The file name (or other specified value) does not exist.
OPOS_E_FAILURE(111)	The device cannot perform the requested procedure, even though the device is connected to the system, powered on, and on-line.
OPOS_E_TIMEOUT(112)	The Service Object timed out waiting for a response from the device, or the data was unable to be transmitted to the device within the timeout.
OPOS_E_BUSY(113)	The current Service Object state does not allow this request. For example, if asynchronous output is in progress, certain methods may not be allowed.
OPOS_E_EXTENDED(114)	A class-specific error condition occurred. The error condition code is available in the ResultCodeExtended property.

ResultCodeExtended Property

Syntax LONG ResultCodeExtended;

Remarks When **ResultCode** is set to OPOS_E_EXTENDED(114), the following POS Printer class-specific error information value and this OPOS Control-specific error information value are set.

When the **ResultCode** is set to other value, a value of 0 is set.

This OPOS Control takes the following values.

ResultCodeExtended	Constant Name (Content)	Description
201 ^{*1}	OPOS_EPTR_COVER_OPEN	Cover is open.
202	OPOS_EPTR_JRN_EMPTY	The journal is out of paper.
203	OPOS_EPTR_REC_EMPTY	The receipt is out of paper.
206	OPOS_EPTR_TOOBIG	The bitmap file is too wide to print without conversion or too big to convert. The maximum print width must be up to the value of RecLineWidth during normal and inversion mode and within the maximum value of the connected device during the bitmap is rotated 90° to the right/left by the RotatePrint method. For the height, the range not exceeding the maximum value of the connected device is supported during normal and inversion mode and RecLineWidth dots is supported during rotated 90° right/left mode.
207	OPOS_EPTR_BADFORMAT	The specified file is not a bitmap file or not a supported format.
280	OPOS_ESTATS_ERROR	At least one of the specified statistics could not be reset.
Specific error 1001 ^{*2}	OPOS_EPTR_VPPPOWER	Vp voltage error
Specific error 1002	OPOS_EPTR_AUTOCUTTER	Autocutter error has occurred.
Specific error 1003 ^{*3}	OPOS_EPTR_PRS_JAM	Presenter jam error has occurred.
Specific error 1004 ^{*3}	OPOS_EPTR_PRS_RETRACT	Presenter feed error has occurred.
Specific error 1005	OPOS_EPTR_HEAD_TEMP	Head-temperature error has occurred.
Specific error 1007	OPOS_EPTR_PRS_PAPER	Presenter out of paper error has occurred.
Specific error 1008 ^{*2}	OPOS_EPTR_NOCGROM	No CGROM error has occurred.
Specific error 1009	OPOS_EPTR_IMAGEAREA_FULL	No memory is available to register the image to the user area of the printer.

^{*1}: The IFD OPOS Control handles a platen position sensor as a cover open sensor.

^{*2}: This is a non-recoverable error.

^{*3}: This code might not be noticed depending on the registry settings.

State Property

Syntax	LONG State;	
Remarks	Contains the current state of the Control.	
	Value	Meaning
	OPOS_S_CLOSED(1)	The Control is closed.
	OPOS_S_IDLE(2)	The Control is in a good state and is not busy.
	OPOS_S_BUSY(3)	The Control is in a good state and is busy performing output.
	OPOS_S_ERROR(4)	An error has been reported, and the application must recover the Control to a good state before normal I/O can resume. This state is only possible inside the ErrorEvent event handler.
	This property is always readable.	

ControlObjectDescription Property

Syntax	BSTR ControlObjectDescription;	
Remarks	"SII POS Printer Control Object, Copyright (C) 2009 Seiko Instruments Inc." is set. The property identifies the Control Object. This property is always readable.	

ControlObjectVersion Property

Syntax	LONG ControlObjectVersion;	
Remarks	This property holds the Control Object version number. This property is always readable.	

ServiceObjectDescription Property

Syntax	BSTR ServiceObjectDescription;
Remarks	A character string that identifies the Service Object is set to this property. The string to be set differs depending on the printer driver used. Example: "SII IFD00x (2inch) POS Printer Service Object, Copyright (C) 2010 Seiko Instruments Inc." etc. This property is initialized by the Open method.

ServiceObjectVersion Property

Syntax	LONG ServiceObjectVersion;
Remarks	This property holds the Service Object version number. This property is initialized by the Open method.

DeviceDescription Property

Syntax	BSTR DeviceDescription;
Remarks	This property provides devices and related information. The value to be set differs depending on the printer driver used. Example: "SII IFD00x (2inch) POS Printer" etc. This property is initialized by the Open method.

DeviceName Property

Syntax	BSTR DeviceName;
Remarks	Short string identifying the device. This property identifies the device information about it. The value to be set differs depending on the printer driver used. Example: "IFD00x (2inch) POS Printer" etc. This property is initialized by the Open method.

4.4. Specific Properties

CapCharacterSet Property

Syntax	LONG CapCharacterSet;				
Remarks	Holds the printable character setting of the POS Printer. This property has the following value. <table><tr><th>Value</th><th>Meaning</th></tr><tr><td>PTR_CCS_KANJI(11)</td><td>The default character set supports code page 932, including ASCII characters 0x20 through 0x7F and the Japanese kana characters 0xA1 through 0xDF, and also including the Shift-JIS kanji characters, Levels 1 and 2.</td></tr></table> This property is initialized to PTR_CCS_KANJI(11) by the Open method.	Value	Meaning	PTR_CCS_KANJI(11)	The default character set supports code page 932, including ASCII characters 0x20 through 0x7F and the Japanese kana characters 0xA1 through 0xDF, and also including the Shift-JIS kanji characters, Levels 1 and 2.
Value	Meaning				
PTR_CCS_KANJI(11)	The default character set supports code page 932, including ASCII characters 0x20 through 0x7F and the Japanese kana characters 0xA1 through 0xDF, and also including the Shift-JIS kanji characters, Levels 1 and 2.				

CapCoverSensor Property

Syntax	BOOL CapCoverSensor;
Remarks	TRUE: The POS Printer has a "Cover open sensor." This property is initialized to TRUE by the Open method. * The IFD OPOS Control handles a platen position sensor as a cover open sensor.

CapMapCharacterSet Property

Syntax	BOOL CapMapCharacterSet;
Remarks	FALSE: The Service Object cannot exactly map the characters to the character sets defined in the CharacterSetList property. This property is initialized to FALSE by the Open method.

CapTransaction Property

Syntax	BOOL CapTransaction;
Remarks	TRUE: The POS Printer transactions are supported by each station. This property is initialized to TRUE by the Open method.

CapJrnPresent Property

Syntax	BOOL CapJrnPresent;
Remarks	TRUE: Journal can be specified for the station. FALSE: Journal cannot be specified for the station. When Journal is selected for the POS Printer used by the registry, this property is initialized to TRUE by the Open method. Otherwise, it is initialized to FALSE.

CapJrn2Color Property

Syntax	BOOL CapJrn2Color;
Remarks	FALSE: 2 colored print is not supported. This property is initialized to FALSE by the Open method.

CapJrnBold Property

Syntax	BOOL CapJrnBold;
Remarks	TRUE: The journal can print bold characters. FALSE: The journal cannot support bold characters. When Journal is selected for the POS Printer used by the registry, this property is initialized to TRUE by the Open method. Otherwise, it is initialized to FALSE.

CapJrnDhigh Property

Syntax **BOOL CapJrnDhigh;**

Remarks TRUE: The journal can print double high characters.
FALSE: The journal cannot support double high characters.

When Journal is selected for the POS Printer used by the registry, this property is initialized to TRUE by the **Open** method. Otherwise, it is initialized to FALSE.

CapJrnDwide Property

Syntax **BOOL CapJrnDwide;**

Remarks TRUE: The journal can print double wide characters.
FALSE: The journal cannot support double wide characters.

When Journal is selected for the POS Printer used by the registry, this property is initialized to TRUE by the **Open** method. Otherwise, it is initialized to FALSE.

CapJrnDwideDhigh Property

Syntax **BOOL CapJrnDwideDhigh;**

Remarks TRUE: The journal can print double high/double wide characters.
FALSE: The journal cannot support double high/double wide characters.

When Journal is selected for the POS Printer used by the registry, this property is initialized to TRUE by the **Open** method. Otherwise, it is initialized to FALSE.

CapJrnEmptySensor Property

Syntax	BOOL CapJrnEmptySensor;
Remarks	TRUE: The journal has an "Out-of-paper sensor." FALSE: The journal cannot support an "Out-of-paper sensor." When Journal is selected for the POS Printer used by the registry, this property is initialized to TRUE by the Open method. Otherwise, it is initialized to FALSE.

CapJrnItalic Property

Syntax	BOOL CapJrnItalic;
Remarks	FALSE: The journal can not print italic characters. This property is initialized to FALSE by the Open method.

CapJrnNearEndSensor Property

Syntax	BOOL CapJrnNearEndSensor;
Remarks	TRUE: The journal has a "paper-near-end sensor." FALSE: The journal cannot support a "paper-near-end sensor." When Journal is selected for the POS Printer used by the registry and the printer has a "paper-near-end sensor," this property is initialized to TRUE by the Open method. Otherwise, it is initialized to FALSE.

CapJrnUnderline Property

Syntax	BOOL CapJrnUnderline;
Remarks	TRUE: The journal can print underlined characters. FALSE: The journal cannot support underlined characters. When Journal is selected for the POS Printer used by the registry, this property is initialized to TRUE by the Open method. Otherwise, it is initialized to FALSE.

CapJrnCartridgeSensor Property

Syntax	LONG CapJrnCartridgeSensor;
Remarks	0: Journal Cartridge monitoring sensors are not supported. This property is initialized to 0 by the Open method.

CapJrnColor Property

Syntax	LONG CapJrnColor;
Remarks	0: The journal station is invalid. PTR_COLOR_PRIMARY(0x00000001): Journal color cartridges are not supported. When Journal is selected for the POS Printer used by the registry, this property is initialized to PTR_COLOR_PRIMARY(0x00000001) by the Open method. Otherwise, it is initialized to 0.

CapRecPresent Property

Syntax **BOOL CapRecPresent;**

Remarks TRUE: Receipt can be specified for the station.
 FALSE: Receipt cannot be specified for the station.

When Receipt is selected for the POS Printer used by the registry, this property is initialized to TRUE by the **Open** method. Otherwise, it is initialized to FALSE.

CapRec2Color Property

Syntax **BOOL CapRec2Color;**

Remarks FALSE: 2 colored print is not supported.

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

CapRecBarCode Property

Syntax **BOOL CapRecBarCode;**

Remarks TRUE: The receipt has bar code printing capability.

When Receipt is selected for the POS Printer used by the registry, this property is initialized to TRUE by the **Open** method. Otherwise, it is initialized to FALSE.

CapRecBitmap Property

Syntax **BOOL CapRecBitmap;**

Remarks TRUE: The receipt can print bitmaps.

When Receipt is selected for the POS Printer used by the registry, this property is initialized to TRUE by the **Open** method. Otherwise, it is initialized to FALSE.

CapRecBold Property

Syntax **BOOL CapRecBold;**

Remarks TRUE: The receipt can print bold characters.

When Receipt is selected for the POS Printer used by the registry, this property is initialized to TRUE by the **Open** method. Otherwise, it is initialized to FALSE.

CapRecDhigh Property

Syntax **BOOL CapRecDhigh;**

Remarks TRUE: The receipt can print double high characters.

When Receipt is selected for the POS Printer used by the registry, this property is initialized to TRUE by the **Open** method. Otherwise, it is initialized to FALSE.

CapRecDwide Property

Syntax **BOOL CapRecDwide;**

Remarks TRUE: The receipt can print double wide characters.

When Receipt is selected for the POS Printer used by the registry, this property is initialized to TRUE by the **Open** method. Otherwise, it is initialized to FALSE.

CapRecDwideDhigh Property

Syntax **BOOL CapRecDwideDhigh;**

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

When Receipt is selected for the POS Printer used by the registry, this property is initialized to TRUE by the **Open** method. Otherwise, it is initialized to FALSE.

CapRecEmptySensor Property

Syntax **BOOL CapRecEmptySensor;**

Remarks TRUE: The receipt has an "Out-of-paper sensor."

When Receipt is selected for the POS Printer used by the registry, this property is initialized to TRUE by the **Open** method. Otherwise, it is initialized to FALSE.

CapRecItalic Property

Syntax **BOOL CapRecItalic;**

Remarks FALSE: The receipt cannot support Italic characters.

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

CapRecLeft90 Property

Syntax **BOOL CapRecLeft90;**

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

When Receipt is selected for the POS Printer used by the registry, this property is initialized to TRUE by the **Open** method. Otherwise, it is initialized to FALSE.

CapRecNearEndSensor Property

Syntax	BOOL CapRecNearEndSensor;
Remarks	<p>TRUE: The receipt has a "paper-near-end sensor." FALSE: The receipt cannot support a "paper-near-end sensor."</p> <p>When Receipt is selected for the POS Printer used by the registry and the printer has a "paper-near-end sensor." this property is initialized to TRUE by the Open method. Otherwise, it is initialized to FALSE.</p>

CapRecPapercut Property

Syntax	BOOL CapRecPapercut;
Remarks	<p>TRUE: The receipt can perform paper cuts.</p> <p>When Receipt is selected for the POS Printer used by the registry, this property is initialized to TRUE by the Open method. Otherwise, it is initialized to FALSE.</p>

CapRecRight90 Property

Syntax	BOOL CapRecRight90;
Remarks	<p>TRUE: The receipt can print in a rotated 90° right mode.</p> <p>When Receipt is selected for the POS Printer used by the registry, this property is initialized to TRUE by the Open method. Otherwise, it is initialized to FALSE.</p>

CapRecRotate180 Property

Syntax	BOOL CapRecRotate180;
Remarks	TRUE: The receipt can print in a rotated upside down mode. When Receipt is selected for the POS Printer used by the registry, this property is initialized to TRUE by the Open method. Otherwise, it is initialized to FALSE.

CapRecStamp Property

Syntax	BOOL CapRecStamp;
Remarks	FALSE: The receipt cannot support stamp printing. This property is initialized to FALSE by the Open method.

CapRecUnderline Property

Syntax	BOOL CapRecUnderline;
Remarks	TRUE: The receipt can print underlined characters. When Receipt is selected for the POS Printer used by the registry, this property is initialized to TRUE by the Open method. Otherwise, it is initialized to FALSE.

CapRecCartridgeSensor Property

Syntax	LONG CapRecCartridgeSensor;
Remarks	0: Receipt Cartridge monitoring sensors are not supported. This property is initialized to 0 by the Open method.

CapRecColor Property

Syntax	LONG CapRecColor;
Remarks	<p>0: The receipt station is invalid.</p> <p>PTR_COLOR_PRIMARY(0x00000001): Receipt color cartridges are not supported.</p> <p>When Receipt is selected for the POS Printer used by the registry, this property is initialized to PTR_COLOR_PRIMARY(0x00000001) by the Open method. Otherwise, it is initialized to 0.</p>

CapRecMarkFeed Property

Syntax	LONG CapRecMarkFeed;
Remarks	<p>0: The Receipt station cannot support the Control function for the marked thermal paper.</p> <p>The Control function for the marked thermal paper is indicated. The Control function for the marked thermal paper is not supported.</p> <p>This property is initialized to 0 by the Open method.</p>

CapRecPageMode Property

Syntax	BOOL CapRecPageMode;
Remarks	<p>TRUE: The printer is capable of supporting Page Mode for the receipt station.</p> <p>When Receipt is selected for the POS Printer used by the registry, this property is initialized to TRUE by the Open method. Otherwise, it is initialized to FALSE.</p>

AsyncMode Property R/W

Syntax	BOOL AsyncMode;
Remarks	TRUE: The print methods for DirectIO , PrintNormal , CutPaper , PrintBarCode , PrintBitmap , RotatePrint , and TransactionPrint are performed asynchronously. FALSE: The methods are performed synchronously. This property is initialized to FALSE by the Open method.

CartridgeNotify Property R/W

Syntax	LONG CartridgeNotify;				
Remarks	PTR_CN_DISABLED(0): Cartridge state notification is not provided. This property is initialized to PTR_CN_DISABLE (0) by the Open method.				
Return	When this property is set, the following value is placed in the ResultCode property:				
<table><tr><th>Value</th><th>Meaning</th></tr><tr><td>OPOS_E_ILLEGAL(106)</td><td>Property setting is not supported.</td></tr></table>		Value	Meaning	OPOS_E_ILLEGAL(106)	Property setting is not supported.
Value	Meaning				
OPOS_E_ILLEGAL(106)	Property setting is not supported.				

CharacterSet Property R/W

Syntax	LONG CharacterSet;	
Remarks	Holds the character set for printing characters. This property is initialized to 999 after the Open method. It has one of the following values:	
	Value	Meaning
	437	Selects the PC437 (USA: Standard Europe) character set.
	932	Selects the Katakana character set as the Japanese (Shift-JIS) character set.
	PTR_CS_WINDOWS (999)	Sets the Windows ANSI characters.
	1252	Selects Codepage 1252 character set.
Return	When this property is set, one of the following values is placed in the ResultCode property:	
	Value	Meaning
	OPOS_SUCCESS(0)	This property was set successfully.
	OPOS_E_NOTCLAIMED(103)	Exclusive access is not available. Call the ClaimDevice method to gain exclusive access.
	OPOS_E_DISABLED(105)	Not enabled. Call after setting DeviceEnabled property to TRUE.
	OPOS_E_ILLEGAL(106)	Incorrect value was used.

CharacterSetList Property

Syntax	BSTR CharacterSetList;
Remarks	This is a character string of character set number. "437,932,999,1252" is set. This property is initialized to "437,932,999,1252" by the Open method.

CoverOpen Property

Syntax	BOOL CoverOpen;
Remarks	TRUE: Platen position error has occurred. FALSE: None of the errors above has occurred. This property is initialized while the device is enabled and keeps the current state.

ErrorLevel Property

Syntax	LONG ErrorLevel;								
Remarks	Holds the severity of the error condition. It has one of the following values: <table><tr><th>Value</th><th>Meaning</th></tr><tr><td>PTR_EL_NONE(1)</td><td>No error condition is present.</td></tr><tr><td>PTR_EL_RECOVERABLE(2)</td><td>A recoverable error has occurred.</td></tr><tr><td>PTR_EL_FATAL(3)</td><td>A non-recoverable error has occurred. This error is set when the ResultCode property is OPOS_E_EXTENDED(114) and the ResultCodeExtended property is one of the following values. OPOS_EPTR_NOCGROM(108) OPOS_EPTR_VPPower(1001) OPOS_EPTR_PRS_JAM (1003) OPOS_EPTR_PRS_RETRACT(1004)</td></tr></table>	Value	Meaning	PTR_EL_NONE(1)	No error condition is present.	PTR_EL_RECOVERABLE(2)	A recoverable error has occurred.	PTR_EL_FATAL(3)	A non-recoverable error has occurred. This error is set when the ResultCode property is OPOS_E_EXTENDED(114) and the ResultCodeExtended property is one of the following values. OPOS_EPTR_NOCGROM(108) OPOS_EPTR_VPPower(1001) OPOS_EPTR_PRS_JAM (1003) OPOS_EPTR_PRS_RETRACT(1004)
Value	Meaning								
PTR_EL_NONE(1)	No error condition is present.								
PTR_EL_RECOVERABLE(2)	A recoverable error has occurred.								
PTR_EL_FATAL(3)	A non-recoverable error has occurred. This error is set when the ResultCode property is OPOS_E_EXTENDED(114) and the ResultCodeExtended property is one of the following values. OPOS_EPTR_NOCGROM(108) OPOS_EPTR_VPPower(1001) OPOS_EPTR_PRS_JAM (1003) OPOS_EPTR_PRS_RETRACT(1004)								
	This property is set by the Control just before the notification of ErrorEvent . When the error is cleared, then the property is changed to PTR_EL_NONE (1).								

ErrorStation Property

Syntax	LONG ErrorStation;						
Remarks	Holds the station that was printing when an error was detected. One of the following values is set to this property. <table><tr><th>Value</th><th>Meaning</th></tr><tr><td>PTR_S_JOURNAL(1)</td><td>Journal printer</td></tr><tr><td>PTR_S_RECEIPT(2)</td><td>Receipt printer</td></tr></table>	Value	Meaning	PTR_S_JOURNAL(1)	Journal printer	PTR_S_RECEIPT(2)	Receipt printer
Value	Meaning						
PTR_S_JOURNAL(1)	Journal printer						
PTR_S_RECEIPT(2)	Receipt printer						
	This property is set before ErrorEvent is notified.						

ErrorString Property

Syntax **BSTR ErrorString;**

Remarks Holds a vendor-supplied description of the current error.

ResultCode	Value	Priority if setting
OPOS_E_NOHARDWARE(107)	Off/Offline error	12
OPOS_E_FAILURE(111)	Failure	10
OPOS_E_TIMEOUT(112)	Time out	11

ResultCodeExtended	Value	Priority if setting
OPOS_EPTR_COVER_OPEN(201)	Platen open error	8
OPOS_EPTR_JRN_EMPTY (202)	Out-of-paper error	5
OPOS_EPTR_REC_EMPTY(203)	Out-of-paper error	6
OPOS_EPTR_VPPPOWER(1001)	Unrecoverable error	1
OPOS_EPTR_AUTOCUTTER(1002)	Autocutter error	2
OPOS_EPTR_PRS_JAM(1003)	Presenter jam error	3
OPOS_EPTR_PRS_RETRACT(1004)	Presenter retract error	4
OPOS_EPTR_HEAD_TEMP(1005)	Head temperature error	9
OPOS_EPTR_PRS_PAPER (1007)	Presenter Out-of-paper	7

If multiple errors occur at the same time, the error with the highest priority is indicated.

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

FontTypefaceList Property

Syntax **BSTR FontTypefaceList;**

Remarks Holds the fonts and/or typefaces that are supported by the printer.
An "(empty string)" indicates that only the default typeface is supported.
This property is initialized to "(empty string)" by the **Open** method.

FlagWhenIdle Property R/W

Syntax	BOOL FlagWhenIdle;								
Remarks	<p>TRUE: A StatusUpdateEvent will be enqueued when the device is in the idle state.</p> <p>FALSE: This event is not notified.</p> <p>This property is automatically reset to FALSE when this status event is delivered. The main use of idle status event that is controlled by this property is to give the application control when all outstanding asynchronous outputs have been processed. The event will be enqueued if the outputs were completed successfully or if they were cleared by the event handler that receives ErrorEvent.</p> <p>If the State property is already set to OPOS_S_IDLE(2) when the FlagWhenIdle property is set to TRUE, then a StatusUpdateEvent is enqueued immediately. The application can therefore depend upon the event, with no race condition between the starting of its last asynchronous output and the setting of this flag.</p> <p>This property is initialized to FALSE by the Open method.</p> <p>When this property is set, the following value is placed in the ResultCode property:</p> <table><thead><tr><th>Value</th><th>Meaning</th></tr></thead><tbody><tr><td>OPOS_SUCCESS(0)</td><td>The property was set successfully.</td></tr><tr><td>OPOS_E_NOTCLAIMED(103)</td><td>Exclusive access is not available. Call the ClaimDevice method to gain exclusive access.</td></tr><tr><td>OPOS_E_DISABLED(105)</td><td>Not enabled. Call after setting DeviceEnabled property to TRUE.</td></tr></tbody></table>	Value	Meaning	OPOS_SUCCESS(0)	The property was set successfully.	OPOS_E_NOTCLAIMED(103)	Exclusive access is not available. Call the ClaimDevice method to gain exclusive access.	OPOS_E_DISABLED(105)	Not enabled. Call after setting DeviceEnabled property to TRUE.
Value	Meaning								
OPOS_SUCCESS(0)	The property was set successfully.								
OPOS_E_NOTCLAIMED(103)	Exclusive access is not available. Call the ClaimDevice method to gain exclusive access.								
OPOS_E_DISABLED(105)	Not enabled. Call after setting DeviceEnabled property to TRUE.								

MapCharacterSet Property R/W

Syntax	BOOL MapCharacterSet;				
Remarks	This property is initialized to FALSE by the Open method.				
Return	When this property is set, the following value is placed in the ResultCode property: <table><thead><tr><th>Value</th><th>Meaning</th></tr></thead><tbody><tr><td>OPOS_E_ILLEGAL(106)</td><td>Property setting is not supported.</td></tr></tbody></table>	Value	Meaning	OPOS_E_ILLEGAL(106)	Property setting is not supported.
Value	Meaning				
OPOS_E_ILLEGAL(106)	Property setting is not supported.				

MapMode Property R/W

Syntax LONG MapMode;

Remarks Holds the mapping mode of the POS Printer. The mapping mode defines the unit of measure used for other properties, such as line heights and line spacing. The following mapping modes are supported. The values inside () indicate the value converted into dot.

Value	Meaning
PTR_MM_DOTS(1)	POS Printer's dot width, 0.125 mm (1 dot)
PTR_MM_TWIPS(2)	1/1,440 of an inch (0.1411 dot)
PTR_MM_ENGLISH(3)	0.001 inch (0.203 dot)
PTR_MM_METRIC(4)	0.01 mm (0.08 dot)

For each mapping mode, the unit is converted using one of the following calculation formulae.

MapMode Property	Conversion
PTR_MM_DOTS(1) Printer dot width (dot value)	No conversion
PTR_MM_TWIPS(2) 1/1,440 inch	$\text{twips} = (180 / \text{inch}) \times (\text{dot value})$ $(\text{dot value}) = (\text{inch} \times \text{twips}) / 180$
PTR_MM_ENGLISH(3) 0.001 inch	$\text{english} = (125 / \text{inch}) \times (\text{dot value})$ $(\text{dot value}) = (\text{english} \times \text{twips}) / 125$
PTR_MM_METRIC(4) 0.01 mm	$\text{metric} = 12.5 \times (\text{dot value})$ $(\text{dot value}) = \text{metric} / 12.5$

(1 inch = 25.4 mm)

The **MapMode** property only changes the unit of each property for display, and all internal processings are performed in dot regardless of the **MapMode** property. Therefore, the rounding errors of values do not accumulate. When converting a dot value to a map mode value, the value is rounded up to an integer. When converting from a map mode value to a dot value, the decimal part is truncated.

Setting this property may also change **JrnLineSpacing**, **JrnLineWidth**, **JrnLineHeight**, **RecLineSpacing**, **RecLineWidth**, **RecLineHeight**, **PageModeArea**, **PageModePrintArea**, **PageModeHorizontalPosition**, **PageModeVerticalPosition**.

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

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

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

PageModeArea Property

Syntax **BSTR PageModeArea;**

Remarks Holds the page area for the selected **PageModeStation** expressed in the unit of the measure given by **MapMode**. This page area can be different than the print area and is determined by the hardware capability of the printer. The string consists of two ASCII numbers separated by a comma, in the following order: horizontal size, vertical size.

When a valid station is specified in **PageModeStation** property, one of the following values is set to this property. (When **MapMode** = PTR_MM_DOTS (1))

Value	Meaning
"432,4110"	When selecting IFD00x (2inch)
"384,4610"	When selecting IFD50x (2inch)
"576,3100"	When selecting IFD00x (3inch)/IFD50x (3inch)

A valid station must be specified to the **PageModeStation** property before accessing to this property. When a valid station is not specified, an "(empty string)" is returned.

PageModeDescriptor Property

Syntax LONG PageModeDescriptor;

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

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

The **PageModeStation** property must be set to a valid station before accessing this property, otherwise the value zero (0) is returned.

PageModeHorizontalPosition Property R/W

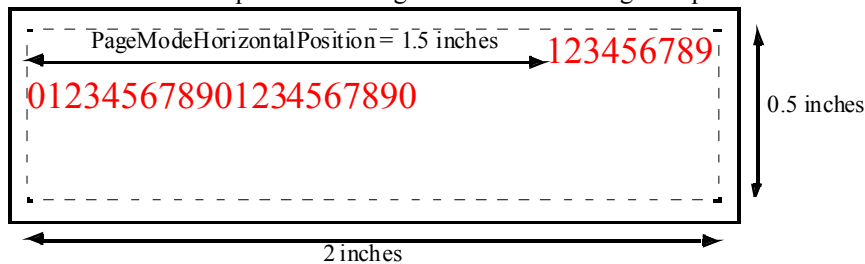
Syntax **LONG PageModeHorizontalPosition;**

Remarks Holds the horizontal start position offset within the print area for the selected **PageModeStation**, expressed in the unit of measure given by **MapMode**. The horizontal direction is the same as the actual **PageModePrintDirection** property. A read/get on this property will return the horizontal position offset set by the last write/set and not the current position. The **PageModeStation** property must be set to a valid station before accessing this property, otherwise the value zero (0) is returned.

The following code sample shows the usage of **PageModeHorizontalPosition**.

```
myptr.setMapMode(PTR_MM_ENGLISH);  
myptr.setPageModeStation(PTR_S_RECEIPT);  
myptr.pageModePrint(PTR_PM_PAGE_MODE);  
// Set print area to 2 inches by 0.5 inches  
myptr.setPageModePrintArea("0,0,2000,500");  
myptr.setPageModePrintDirection(PTR_PD_LEFT_TO_RIGHT);  
myptr.setPageModeHorizontalPosition(1500);  
myptr.printNormal(PTR_S_RECEIPT, "123456789012345678901234567890\n");
```

The code sample above will generate the following receipt.



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

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

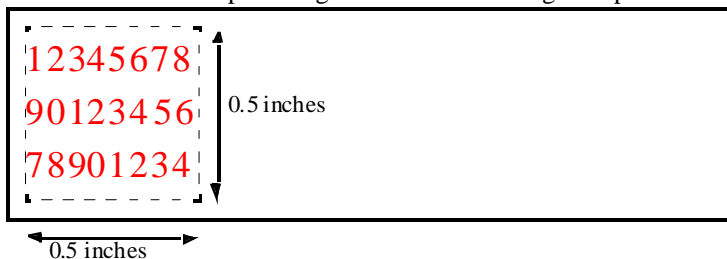
PageModePrintArea Property R/W

Syntax **BSTR PageModePrintArea;**

Remarks Holds the page mode print area for the selected **PageModeStation** property expressed in the unit specified by **MapMode**. The maximum print area is the page area. The property consists of four ASCII numbers separated by commas, in the following order: horizontal start, vertical start, horizontal size, vertical size. For example, if the string is "50,100,200,400," then the station print area is a rectangle beginning at the top left position (50,100), and the bottom right position (249,499). This property is initialized to "0,0,0,0." The text beyond the right edge of the page mode print area will be printed to the next line. Any text or image beyond the bottom of the print area will not be printed. For example:

```
myptr.setMapMode(PTR_MM_ENGLISH);  
myptr.setPageModeStation(PTR_S_RECEIPT);  
myptr.pageModePrint(PTR_PM_PAGE_MODE);  
// Set print area to half inch square block  
myptr.setPageModePrintArea("0,0,500,500");  
myptr.setPageModePrintDirection(PTR_PD_LEFT_TO_RIGHT);  
myptr.printNormal(PTR_S_RECEIPT, "123456789012345678901234567890\n");
```

The above code sample will generate the following receipt.



A valid station must be specified to the **PageModeStation** property before accessing to this property. When a valid station is not specified, an "(empty string)" is returned.

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

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

PageModePrintDirection Property R/W

Syntax LONG PageModePrintDirection;

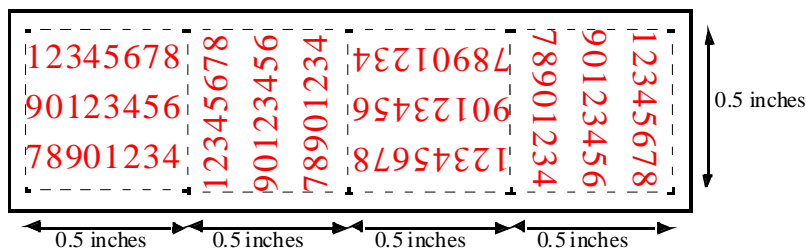
Remarks Holds the print direction. The print direction shall be as follows:

Value	Meaning
PTR_PD_LEFT_TO_RIGHT(1)	Print left to right, starting at top left position of the print area, i.e., normal printing.
PTR_PD_BOTTOM_TO_TOP(2)	Print bottom to top, starting at the bottom left position of the print area, i.e., rotated left 90° printing.
PTR_PD_RIGHT_TO_LEFT(3)	Print right to left, starting at the bottom right position of the print area, i.e., upside down printing.
PTR_PD_TOP_TO_BOTTOM(4)	Print top to bottom, starting at the top right position of the print area, i.e., rotated right 90° printing.

This property is initialized to PTR_PD_LEFT_TO_RIGHT(1) when the device is first enabled following the **Open** method. Changing this property may also changes the correction direction of the print start point indicated by the **PageModeHorizontalPosition** and **PageModeVerticalPosition** properties. Changing this property is only effective for the current print area. By changing the print areas, it is possible to generate a receipt or slip with text printed in multiple rotations. For example:

```
myptr.setMapMode(PTR_MM_ENGLISH);  
myptr.setPageModeStation(PTR_S_RECEIPT);  
myptr.pageModePrint(PTR_PM_PAGE_MODE);  
// Set print area to half inch square block  
myptr.setPageModePrintArea("0,0,500,500");  
myptr.setPageModePrintDirection(PTR_PD_LEFT_TO_RIGHT);  
myptr.printNormal(PTR_S_RECEIPT, "123456789012345678901234567890\n");  
myptr.setPageModePrintArea("500,0,500,500");  
myptr.setPageModePrintDirection(PTR_PD_BOTTOM_TO_TOP);  
myptr.printNormal(PTR_S_RECEIPT, "123456789012345678901234567890\n");  
myptr.setPageModePrintArea("1000,0,500,500");  
myptr.setPageModePrintDirection(PTR_PD_RIGHT_TO_LEFT);  
myptr.printNormal(PTR_S_RECEIPT, "123456789012345678901234567890\n");  
myptr.setPageModePrintArea("1500,0,500,500");  
myptr.setPageModePrintDirection(PTR_PD_TOP_TO_BOTTOM);  
myptr.printNormal(PTR_S_RECEIPT, "123456789012345678901234567890\n");
```

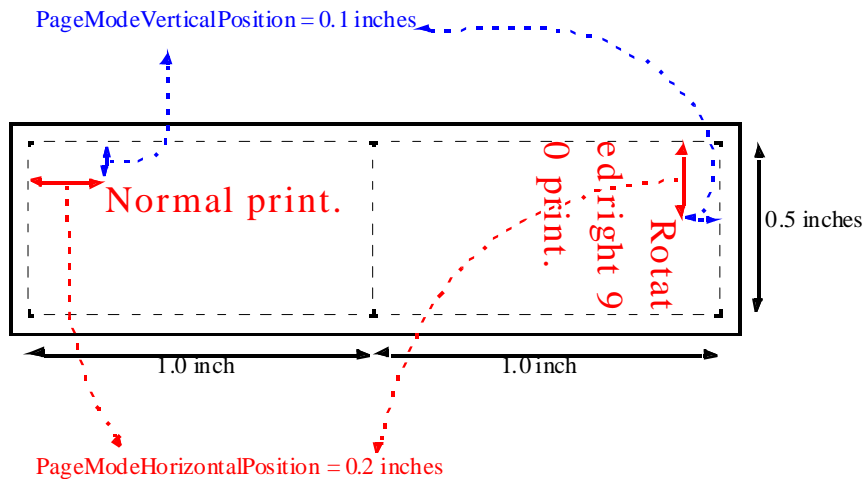
The above code sample will generate the following receipt.



It is also possible to generate rotated text.

```
myptr.setMapMode(PTR_MM_ENGLISH);
myptr.setPageModeStation(PTR_S_RECEIPT);
myptr.pageModePrint(PTR_PM_PAGE_MODE);
myptr.pageModeVerticalPosition(100);
myptr.pageModeHorizontalPosition(200);
myptr.setPageModePrintArea("0,0,1000,500");
myptr.setPageModePrintDirection(PTR_PD_LEFT_TO_RIGHT);
myptr.printNormal(PTR_S_RECEIPT, "Normal print.\n");
myptr.setPageModePrintArea("1000,0,1000,500");
myptr.setPageModePrintDirection(PTR_PD_TOP_TO_BOTTOM);
myptr.printNormal(PTR_S_RECEIPT, "Rotated right 90 print.\n");
myptr.setPageModePrint(PTR_PM_NORMAL);
```

The above code sample will generate the following receipt.



The **PageModeStation** property must be set to a valid station before accessing this property, otherwise the value zero (0) is returned.

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

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

PageModeStation Property R/W

Syntax	LONG PageModeStation;	
Remarks	Set the print station for subsequent Page Mode properties. Note that PageModePrint will allow for the selection of the print station that the output will be generated on. The available station is PTR_S_RECEIPT(2) only. This property is initialized to 0 by the Open method. A valid station must be specified for this property before accessing the property or method of the page mode function.	
Return	When this property is set, the following value is placed in the ResultCode property:	
	Value	Meaning
	OPOS_SUCCESS(0)	The property was set successfully.
	OPOS_E_ILLEGAL(106)	Improper value was specified. This error is returned when Receipt is selected for the POS Printer used by the registry.

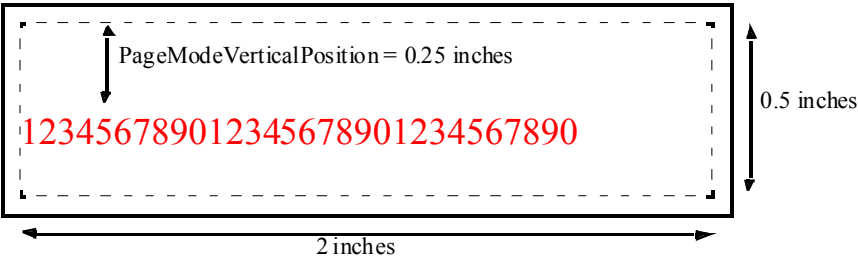
PageModeVerticalPosition Property R/W

Syntax	LONG PageModeVerticalPosition;
Remarks	Holds the vertical start position offset within the print area for the selected PageModeStation , expressed in the unit of measure given by MapMode . The vertical direction is perpendicular to the direction specified in the actual PageModePrintDirection property. If the exact position cannot be supported then the position is set to the closest supported value. A read/get on this property will return the vertical position offset set by the last write/set and not the current position.

The following code sample shows usage of **PageModeVerticalPosition**.

```
myptr.setMapMode(PTR_MM_ENGLISH);
myptr.setPageModeStation(PTR_S_RECEIPT);
myptr.pageModePrint(PTR_PM_PAGE_MODE);
// Set print area to 2 inches by 0.5 inches
myptr.setPageModePrintArea("0,0,2000,500");
myptr.setPageModePrintDirection(PTR_PD_LEFT_TO_RIGHT);
myptr.setPageModeVerticalPosition(250);
myptr.printNormal(PTR_S_RECEIPT,"123456789012345678901234567890\n");
```

The above code sample will generate the following receipt.



The **PageModeStation** property must be set to a valid station before accessing this property, otherwise the value zero (0) is returned.

Return	When this property is set, the following value is placed in the ResultCode property:
Value	Meaning
OPOS_SUCCESS(0)	The property was set successfully.
OPOS_E_ILLEGAL(106)	Improper value was specified.

RotateSpecial Property R/W

Syntax **LONG RotateSpecial;**

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

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

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

Value	Meaning
OPOS_SUCCESS(0)	The property was set successfully.
OPOS_E_ILLEGAL(106)	Improper value of the property was specified.

JrnLineChars Property R/W

Syntax LONG JrnLineChars;

Remarks Holds the number of characters that may be printed on a journal line.
Depending on the specified number of characters per line, the printer prints in the following font.

○ For paper width: 2 inches (when IFD001 is used)

JrnLineChars	Print font (H×W)	Character space
27	Font A(24×12dots)	4 dots
30	Font A(24×12dots)	2 dots
33	Font A(24×12dots)	1 dot
36 (Default)	Font A(24×12dots)	0 dot
43	Font B(16×8dots)	2 dots
48	Font B(16×8dots)	1 dot
54	Font B(16×8dots)	0 dot

○ For paper width: 2 inches (when IFD501 is used)

JrnLineChars	Print font (H×W)	Character space
24	Font A(24×12dots)	4 dots
27	Font A(24×12dots)	2 dots
29	Font A(24×12dots)	1 dot
32 (Default)	Font A(24×12dots)	0 dot
38	Font B(16×8dots)	2 dots
42	Font B(16×8dots)	1 dot
48	Font B(16×8dots)	0 dot

○ For paper width: 3 inches

JrnLineChars	Print font (H×W)	Character space
36	Font A(24×12dots)	4 dots
41	Font A(24×12dots)	2 dots
44	Font A(24×12dots)	1 dot
48 (Default)	Font A(24×12dots)	0 dot
57	Font B(16×8dots)	2 dots
64	Font B(16×8dots)	1 dot
72	Font B(16×8dots)	0 dot

If changed to a line character width that is less than or equal to the maximum value allowed for the printer, then the width is set to the specified value. If the exact width cannot be supported, then subsequent lines will be printed with a character size that most closely supports the specified characters per line. (For example, if 40 is set when the paper width is 2 inches (when IFD001 is used), then the Service Object selects a **JrnLineChars** =43.)

If the character width is not supported, then an error is returned. (For example, if 60 is set when the paper width is 2 inches (when IFD001 is used), then an error occurs.)

Setting **JrnLineChars** also updates **JrnLineHeight** and **JrnLineSpacing**.

This value of **JrnLineChars** is initialized to the default value by the **Open** method.

However, when the POS Printer specified by the registry is not journal, it is initialized to 0.

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

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

JrnLineCharsList Property

Syntax **BSTR JrnLineCharsList;**

Remarks Holds the line character widths supported by the journal station.
This property is initialized to the following values by the **Open** method. However, when the POS Printer specified by the registry is not journal, it is initialized to "(empty string)."

Value	Meaning
"27,30,33,36,43,48,54"	For paper width: 2 inches (when IFD001 is used)
"24,27,29,32,38,42,48"	For paper width: 2 inches (when IFD501 is used)
"36,41,44,48,57,64,72"	For paper width: 3 inches

JrnLineHeight Property R/W

Syntax LONG JrnLineHeight;

Remarks Holds the journal print line height, expressed in the unit defined by **MapMode**.
If updating the content of this property, an error is occurred since there are plural number of character/line with same height.
When **JrnLineChars** is changed, **JrnLineHeight** is updated to the default line height for the selected **JrnLineChars**. The relationship between **JrnLineChars** value and **JrnLineHeight** are as follows.

○ For paper width: 2 inches (when IFD001 is used)

JrnLineChars	JrnLineHeight (MapMode = PTR_MM_DOTS (1))
--------------	-------------------------------------------

27,30,33,36	24
-------------	----

43,48,54	16
----------	----

○ For paper width: 2 inches (when IFD501 is used)

JrnLineChars	JrnLineHeight (MapMode = PTR_MM_DOTS (1))
--------------	-------------------------------------------

24,27,29,32	24
-------------	----

38,42,48	16
----------	----

○ For paper width: 3 inches

JrnLineChars	JrnLineHeight (MapMode = PTR_MM_DOTS (1))
--------------	-------------------------------------------

36,41,44,48	24
-------------	----

57,64,72	16
----------	----

This property is initialized to 24 by the **Open** method. However, when the POS Printer specified by the registry is not journal, it is initialized to 0.

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

Value	Meaning
-------	---------

OPOS_E_ILLEGAL(106)	The content of property cannot be changed.
---------------------	--------------------------------------------

JrnLineSpacing Property R/W

Syntax LONG JrnLineSpacing;

Remarks Holds the spacing of each single-high print line, including both the printed line height plus the space between each pair of lines. Line spacing is expressed in the unit of measure given by **MapMode**. The value that is smaller than **JrnLineHeight** cannot be specified. If the smaller value is specified, OPOS_E_ILLEGAL(106) is informed and the property is ignored.
Also, the configurable range differs depending on the setting of **JrnLineChars**. The configurable ranges are as follows.

○ For paper width: 2 inches (when IFD001 is used)

JrnLineChars	Configurable range (MapMode = PTR_MM_DOTS (1))
--------------	---------------------------------------------------

27,30,33,36	24 - 255
43,48,54	16 - 255

○ For paper width: 2 inches (when IFD501 is used)

JrnLineChars	Configurable range (MapMode = PTR_MM_DOTS (1))
--------------	---------------------------------------------------

24,27,29,32	24 - 255
38,42,48	16 - 255

○ For paper width: 3 inches

JrnLineChars	Configurable range (MapMode = PTR_MM_DOTS (1))
--------------	---------------------------------------------------

36,41,44,48	24 - 255
57,64,72	16 - 255

If the value other than the configurable range is specified, OPOS_E_ILLEGAL(106) is informed and the property is ignored. When **JrnLineChars** is changed and if new **JrnLineHeight** is bigger than the value of **JrnLineSpacing**, the value of **JrnLineHeight** is set. In the same way, if the value of **JrnLineSpacing** exceeds the configurable range, the maximum value of the configurable range is automatically set.

This property is initialized to 30 (when the **MapMode** property is PTR_MM_DOTS(1)) by the **Open** method. However, when the POS Printer specified by the registry is not journal, it is initialized to 0.

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

Value	Meaning
OPOS_SUCCESS(0)	The property was set successfully.
OPOS_E_ILLEGAL(106)	The setting range of property was improper.

JrnLineWidthProperty

Syntax	LONG JrnLineWidth;								
Remarks	<p>Holds the width of a line of JrnLineChars, expressed in the unit of measure given by MapMode.</p> <p>This property is initialized to the following values depending on the model after the Open method. However, when the POS Printer specified by the registry is not journal, it is initialized to 0.</p> <table><tr><th>Model (Paper width)</th><th>JrnLineWidth (MapMode = PTR_MM_DOTS (1))</th></tr><tr><td>2 inch (when IFD001 is used)</td><td>432</td></tr><tr><td>2 inch (when IFD501 is used)</td><td>384</td></tr><tr><td>3 inch</td><td>576</td></tr></table>	Model (Paper width)	JrnLineWidth (MapMode = PTR_MM_DOTS (1))	2 inch (when IFD001 is used)	432	2 inch (when IFD501 is used)	384	3 inch	576
Model (Paper width)	JrnLineWidth (MapMode = PTR_MM_DOTS (1))								
2 inch (when IFD001 is used)	432								
2 inch (when IFD501 is used)	384								
3 inch	576								

JrnLetterQuality Property R/W

Syntax	BOOL JrnLetterQuality;
Remarks	<p>FALSE: Prints in the high speed mode.</p> <p>This property is initialized to FALSE by the Open method.</p>

JrnEmpty Property

Syntax	BOOL JrnEmpty;
Remarks	<p>TRUE: The journal is out of paper.</p> <p>FALSE: The journal paper is present.</p> <p>This property is initialized and kept current while the device is enabled. However, when the POS Printer specified by the registry is not journal, it is always set to FALSE.</p>

JrnNearEnd Property

Syntax	BOOL JrnNearEnd;
Remarks	TRUE: The journal paper is low. FALSE: The journal paper is not low. This property is initialized and kept current while the device is enabled. However, when the POS Printer specified by the registry is not journal, it is always set to FALSE.

JrnCartridgeState Property

Syntax	LONG JrnCartridgeState;				
Remarks	This property contains the status of the currently selected Journal cartridge (ink, ribbon or toner). It contains the following value. <table><tr><th>Value</th><th>Meaning</th></tr><tr><td>PTR_CART_UNKNOWN (0x10000000)</td><td>Device does not support cartridge state reporting.</td></tr></table> This property is initialized to PTR_CART_UNKNOWN(0x10000000) by the Open method.	Value	Meaning	PTR_CART_UNKNOWN (0x10000000)	Device does not support cartridge state reporting.
Value	Meaning				
PTR_CART_UNKNOWN (0x10000000)	Device does not support cartridge state reporting.				

JrnCurrentCartridge Property R/W

Syntax	LONG JrnCurrentCartridge;				
Remarks	Journal Cartridge selection is not supported. This property is initialized to PTR_COLOR_PRIMARY(0x00000001) when the journal is valid in the registry or to 0 when the journal is invalid by the Open method.				
Return	When this property is set, the following value is placed in the ResultCode property: <table><tr><th>Value</th><th>Meaning</th></tr><tr><td>OPOS_E_ILLEGAL(106)</td><td>An invalid property value was specified.</td></tr></table>	Value	Meaning	OPOS_E_ILLEGAL(106)	An invalid property value was specified.
Value	Meaning				
OPOS_E_ILLEGAL(106)	An invalid property value was specified.				

RecLineChars Property R/W

Syntax LONG RecLineChars;

Remarks Holds the number of characters that may be printed on a receipt line. Depending on the specified number of characters per line, the printer prints in the following font.

○ For paper width: 2 inches (when IFD001 is used)

RecLineChars	Print font (H×W)	Character space
27	Font A(24×12dots)	4 dots
30	Font A(24×12dots)	2 dots
33	Font A(24×12dots)	1 dot
36 (Default)	Font A(24×12dots)	0 dot
43	Font B(16×8dots)	2 dots
48	Font B(16×8dots)	1 dot
54	Font B(16×8dots)	0 dot

○ For paper width: 2 inches (when IFD501 is used)

RecLineChars	Print font (H×W)	Character space
24	Font A(24×12dots)	4 dots
27	Font A(24×12dots)	2 dots
29	Font A(24×12dots)	1 dot
32 (Default)	Font A(24×12dots)	0 dot
38	Font B(16×8dots)	2 dots
42	Font B(16×8dots)	1 dot
48	Font B(16×8dots)	0 dot

○ For paper width: 3 inches

RecLineChars	Print font (H×W)	Character space
36	Font A(24×12dots)	4 dots
41	Font A(24×12dots)	2 dots
44	Font A(24×12dots)	1 dot
48 (Default)	Font A(24×12dots)	0 dot
57	Font B(16×8dots)	2 dots
64	Font B(16×8dots)	1 dot
72	Font B(16×8dots)	0 dot

If changed to a line character width that is less than or equal to the maximum value allowed for the printer, then the width is set to the specified value. If the exact width cannot be supported, then subsequent lines will be printed with a character size that most closely supports the specified characters per line. (For example, if 40 is set when the paper width is 2 inches (when IFD001 is used), then the Service Object selects a **RecLineChars** =43.)

If the character width is not supported, then an error is returned. (For example, if 60 is set when the paper width is 2 inches (when IFD001 is used), then an error occurs.)

Setting **RecLineChars** may also update **RecLineHeight** and **RecLineSpacing**.

This property is initialized to the default value by the **Open** method. However, when the POS Printer specified by the registry is not receipt, it is initialized to 0.

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

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

RecLineCharsList Property

Syntax	BSTR RecLineCharsList;	
Remarks	Holds the line character widths supported by the receipt station. This property is initialized to the following values by the Open method. However, when the POS Printer specified by the registry is not receipt, it is initialized to "(empty string)."	
	Value	Meaning
	"27,30,33,36,43,48,54"	For paper width: 2 inches (when IFD001 is used)
	"24,27,29,32,38,42,48"	For paper width: 2 inches (when IFD501 is used)
	"36,41,44,48,57,64,72"	For paper width: 3 inches

RecLineHeight Property R/W

Syntax	LONG RecLineHeight;																		
Remarks	<p>Holds the receipt print line height, expressed in the unit of measure given by MapMode.</p> <p>If updating the content of this property, an error is occurred since there are plural number of character/line with same height. When RecLineChars is changed, RecLineHeight is updated to the default line height for the selected RecLineChars. The relationship between RecLineChars value and RecLineHeight is as follows.</p> <p>○ For paper width: 2 inches (when IFD001 is used)</p> <table><tr><th>RecLineChars</th><th>RecLineHeight (MapMode = PTR_MM_DOTS (1))</th></tr><tr><td>27,30,33,36</td><td>24</td></tr><tr><td>43,48,54</td><td>16</td></tr></table> <p>○ For paper width: 2 inches (when IFD501 is used)</p> <table><tr><th>RecLineChars</th><th>RecLineHeight (MapMode = PTR_MM_DOTS (1))</th></tr><tr><td>24,27,29,32</td><td>24</td></tr><tr><td>38,42,48</td><td>16</td></tr></table> <p>○ For paper width: 3 inches</p> <table><tr><th>RecLineChars</th><th>RecLineHeight (MapMode = PTR_MM_DOTS (1))</th></tr><tr><td>36,41,44,48</td><td>24</td></tr><tr><td>57,64,72</td><td>16</td></tr></table> <p>This property is initialized to 24 by the Open method. However, when the POS Printer specified by the registry is not receipt, it is initialized to 0.</p>	RecLineChars	RecLineHeight (MapMode = PTR_MM_DOTS (1))	27,30,33,36	24	43,48,54	16	RecLineChars	RecLineHeight (MapMode = PTR_MM_DOTS (1))	24,27,29,32	24	38,42,48	16	RecLineChars	RecLineHeight (MapMode = PTR_MM_DOTS (1))	36,41,44,48	24	57,64,72	16
RecLineChars	RecLineHeight (MapMode = PTR_MM_DOTS (1))																		
27,30,33,36	24																		
43,48,54	16																		
RecLineChars	RecLineHeight (MapMode = PTR_MM_DOTS (1))																		
24,27,29,32	24																		
38,42,48	16																		
RecLineChars	RecLineHeight (MapMode = PTR_MM_DOTS (1))																		
36,41,44,48	24																		
57,64,72	16																		
Return	<p>When this property is set, the following value is placed in the ResultCode property:</p> <table><tr><th>Value</th><th>Meaning</th></tr><tr><td>OPOS_E_ILLEGAL(106)</td><td>The content of property cannot be changed.</td></tr></table>	Value	Meaning	OPOS_E_ILLEGAL(106)	The content of property cannot be changed.														
Value	Meaning																		
OPOS_E_ILLEGAL(106)	The content of property cannot be changed.																		

RecLineSpacing Property R/W

Syntax LONG RecLineSpacing;

Remarks Holds the spacing of each single-high print line, including both the printed line height plus the whitespace between each pair of lines. Line spacing is expressed in the unit of measure given by **MapMode**. The value that is smaller than **RecLineHeight** cannot be specified. If the smaller value is specified, OPOS_E_ILLEGAL(106) is informed and the property is ignored. Also, the configurable range differs depending on the setting of **RecLineChars**. The configurable ranges are as follows.

○ For paper width: 2 inches (when IFD001 is used)

RecLineChars	Configurable range (MapMode = PTR_MM_DOTS (1))
---------------------	------------------------------------------------------------------

27,30,33,36	24 - 255
-------------	----------

43,48,54	16 - 255
----------	----------

○ For paper width: 2 inches (when IFD501 is used)

RecLineChars	Configurable range (MapMode = PTR_MM_DOTS (1))
---------------------	------------------------------------------------------------------

24,27,29,32	24 - 255
-------------	----------

38,42,48	16 - 255
----------	----------

○ For paper width: 3 inches

RecLineChars	Configurable range (MapMode = PTR_MM_DOTS (1))
---------------------	------------------------------------------------------------------

36,41,44,48	24 - 255
-------------	----------

57,64,72	16 - 255
----------	----------

If the value other than the configurable range is specified, OPOS_E_ILLEGAL(106) is informed and the property is ignored. When **RecLineChars** is changed and if new **RecLineHeight** is bigger than the value specified for **RecLineSpacing**, the value of **RecLineHeight** is set. In the same way, if the value of **RecLineSpacing** exceeds the configurable range, the maximum value of configurable range is automatically set.

This property is initialized to 30 (when the **MapMode** property is PTR_MM_DOTS(1)) by the **Open** method. However, when the POS Printer specified by the registry is not receipt, it is initialized to 0.

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

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

RecLineWidth Property

Syntax	LONG RecLineWidth;								
Remarks	Holds the width of a line of RecLineChars characters, expressed in the unit of measure given by MapMode . This property is initialized to one of the following values depending on the model after the Open method. However, when the POS Printer specified by the registry is not receipt, it is initialized to 0. <table><tr><th>Model (Paper width)</th><th>RecLineWidth (MapMode = PTR_MM_DOTS (1))</th></tr><tr><td>2 inch (when IFD001 is used)</td><td>432</td></tr><tr><td>2 inch (when IFD501 is used)</td><td>384</td></tr><tr><td>3 inch</td><td>576</td></tr></table>	Model (Paper width)	RecLineWidth (MapMode = PTR_MM_DOTS (1))	2 inch (when IFD001 is used)	432	2 inch (when IFD501 is used)	384	3 inch	576
Model (Paper width)	RecLineWidth (MapMode = PTR_MM_DOTS (1))								
2 inch (when IFD001 is used)	432								
2 inch (when IFD501 is used)	384								
3 inch	576								

RecLetterQuality Property R/W

Syntax	BOOL RecLetterQuality;
Remarks	FALSE: Prints in the high speed mode. This property is initialized to FALSE by the Open method.

RecEmpty Property

Syntax	BOOL RecEmpty;
Remarks	TRUE: The receipt is out of paper. FALSE: The receipt paper is present. This property is initialized and kept current while the device is enabled. However, when the POS Printer specified by the registry is not receipt, it is always set to FALSE.

RecNearEnd Property

Syntax	BOOL RecNearEnd;
Remarks	TRUE: The receipt paper is low. FALSE: The receipt paper is not low. This property is initialized when the device is enabled and keeps the current value while the device is enabled. However, when the POS Printer specified by the registry is not receipt, it is always set to FALSE.

RecSidewaysMaxLines Property

Syntax	LONG RecSidewaysMaxLines;
Remarks	Holds the maximum number of lines that may be printed in the sideways mode (rotated 90° to the left or right). The RecLineWidth property is the value calculated from the RecLineSpacing property. Therefore, this property can be changed by changing the RecLineSpacing property. This property is initialized when the Open method is executed. However, when the POS Printer specified by the registry is not receipt, it is always set to 0.

RecSidewaysMaxChars Property

Syntax	LONG RecSidewaysMaxChars;
Remarks	Holds the maximum number of characters that may be printed on each line in sideways mode (rotated 90° to the left or right). This property is determined by the following calculation based on the PageModeArea property and print font. However, when the POS Printer specified by the registry is not receipt, it is always set to 0. RecSidewaysMaxChars Property = Maximum height of PageModeArea property / (Print font width + Character space) Example: When the PageModeArea property is "432,4110" and the RecLineChars property is 36, RecSidewaysMaxChars Property is: = 4110 / (12 + 0) = 342 (The decimal part is truncated.)

RecLinesToPaperCut Property

Syntax	LONG RecLinesToPaperCut;
Remarks	Holds the number of lines that must be advanced before the receipt paper is cut. This is the line count before reaching the paper cut mechanism. The distance between the print head and the cutter is about 11 mm. The value which is obtained by dividing this distance by the amount of line count shown on RecLineSpacing is set. Therefore, this property can be changed by changing RecLineSpacing . This property is initialized to 3 by the Open method. However, when the POS Printer specified by the registry is not receipt, it is always set to 0.

RecBarCodeRotationList Property

Syntax	BSTR RecBarCodeRotationList;										
Remarks	Holds the directions in which a receipt bar code may be rotated. This property is initialized to "0,R90,L90,180" by the Open method. The string consists of rotation strings separated by commas. The legal rotation strings are as follows: <table><tr><th>Value</th><th>Meaning</th></tr><tr><td>0</td><td>Bar code may be printed in the normal orientation.</td></tr><tr><td>R90</td><td>Bar code may be printed in a rotated 90° to the right.</td></tr><tr><td>L90</td><td>Bar code may be printed in a rotated 90° to the left.</td></tr><tr><td>180</td><td>Bar code may be rotated 180°- upside down.</td></tr></table> However, when the POS Printer specified by the registry is not receipt, it is always set to an "(empty string)."	Value	Meaning	0	Bar code may be printed in the normal orientation.	R90	Bar code may be printed in a rotated 90° to the right.	L90	Bar code may be printed in a rotated 90° to the left.	180	Bar code may be rotated 180°- upside down.
Value	Meaning										
0	Bar code may be printed in the normal orientation.										
R90	Bar code may be printed in a rotated 90° to the right.										
L90	Bar code may be printed in a rotated 90° to the left.										
180	Bar code may be rotated 180°- upside down.										

RecCartridgeState Property

Syntax	LONG RecCartridgeState;				
Remarks	This property contains the status of the currently selected Receipt cartridge (ink, ribbon or toner). It contains the following value. <table><tr><th>Value</th><th>Meaning</th></tr><tr><td>PTR_CART_UNKNOWN (0x10000000)</td><td>Device does not support cartridge state reporting.</td></tr></table> <p>This property is initialized to PTR_CART_UNKNOWN(0x10000000) by the Open method.</p>	Value	Meaning	PTR_CART_UNKNOWN (0x10000000)	Device does not support cartridge state reporting.
Value	Meaning				
PTR_CART_UNKNOWN (0x10000000)	Device does not support cartridge state reporting.				

RecCurrentCartridge Property R/W

Syntax	LONG RecCurrentCartridge;				
Remarks	Receipt Cartridge selection is not supported. This property is initialized to PTR_COLOR_PRIMARY(0x00000001) when the receipt is valid in the registry or to 0 when the receipt is invalid by the Open method.				
Return	When this property is set, the following value is placed in the ResultCode property: <table><tr><th>Value</th><th>Meaning</th></tr><tr><td>OPOS_E_ILLEGAL(106)</td><td>An invalid property value was specified.</td></tr></table>	Value	Meaning	OPOS_E_ILLEGAL(106)	An invalid property value was specified.
Value	Meaning				
OPOS_E_ILLEGAL(106)	An invalid property value was specified.				

RecBitmapRotationList Property

Syntax **BSTR RecBitmapRotationList;**

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

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

However, when the POS Printer specified by the registry is not receipt, it is always set to an "(empty string)."

4.5. Common Methods

Open Method

Syntax	LONG Open (BSTR DeviceName); The <i>DeviceName</i> parameter specifies the device name to open. Specify and execute the registered device name (such as "IFD00x") or "DefaultPOSPrinter."
Remarks	Call this method to open the device. When the Open method is successful, the common property and other class-specific properties are initialized.
Return	One of the following values is returned and stored in the ResultCode property:

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

Note

The value of the **ResultCode** property after calling the **Open** method may not be the same as the **Open** method return value for the following two cases.

- When OPOS Control is closed and the **Open** method is failed:
The **ResultCode** property will continue to return OPOS_E_CLOSED(101).
 - When the OPOS Control is already opened:
The **Open** method will return OPOS_E_ILLEGAL(106), but the **ResultCode** property may continue to return the value it held before the **Open** method.
-

Close Method

Syntax **LONG** Close ();

Remarks Called to release the device and its resources.

If the **DeviceEnabled** property is TRUE, the device is first disabled.

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

Do not execute this while the event is in progress (or in the event handler).

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

Value	Meaning
OPOS_SUCCESS(0)	Device has been disabled and closed.
OPOS_E_BUSY(113)	Asynchronous output is in progress

ClaimDevice Method

Syntax	LONG ClaimDevice (LONG Timeout); The <i>Timeout</i> parameter gives the maximum number of milliseconds to wait for exclusive access to be satisfied. If zero, the method attempts to claim the device, then returns the appropriate status immediately. If OPOS_FOREVER(-1), the method waits as long as needed until exclusive access is satisfied.								
Remarks	Call this method to request exclusive access to the device. The POS Printer device cannot be used until exclusive access is obtained. When successful, the Claimed property is changed to TRUE.								
Return	One of the following values is returned by the method and also placed in the ResultCode property: <table><tr><th>Value</th><th>Meaning</th></tr><tr><td>OPOS_SUCCESS(0)</td><td>Exclusive access is granted. The Claimed property is now TRUE. It is also returned if this application has already gained the exclusive access to the device.</td></tr><tr><td>OPOS_E_ILLEGAL(106)</td><td>This device cannot be claimed for exclusive access, or an invalid <i>Timeout</i> parameter was specified.</td></tr><tr><td>OPOS_E_TIMEOUT(112)</td><td>Another application has exclusive access to the device and the <i>Timeout</i> (in millisecond) has elapsed before the device is released. Or, the POS Printer device is not available before the <i>Timeout</i> (in millisecond) has elapsed.</td></tr></table>	Value	Meaning	OPOS_SUCCESS(0)	Exclusive access is granted. The Claimed property is now TRUE. It is also returned if this application has already gained the exclusive access to the device.	OPOS_E_ILLEGAL(106)	This device cannot be claimed for exclusive access, or an invalid <i>Timeout</i> parameter was specified.	OPOS_E_TIMEOUT(112)	Another application has exclusive access to the device and the <i>Timeout</i> (in millisecond) has elapsed before the device is released. Or, the POS Printer device is not available before the <i>Timeout</i> (in millisecond) has elapsed.
Value	Meaning								
OPOS_SUCCESS(0)	Exclusive access is granted. The Claimed property is now TRUE. It is also returned if this application has already gained the exclusive access to the device.								
OPOS_E_ILLEGAL(106)	This device cannot be claimed for exclusive access, or an invalid <i>Timeout</i> parameter was specified.								
OPOS_E_TIMEOUT(112)	Another application has exclusive access to the device and the <i>Timeout</i> (in millisecond) has elapsed before the device is released. Or, the POS Printer device is not available before the <i>Timeout</i> (in millisecond) has elapsed.								

ReleaseDevice Method

Syntax	LONG ReleaseDevice ();								
Remarks	Call this method to release exclusive access to the device. If the DeviceEnabled property is TRUE and the device is an exclusive-use device, then the device is first disabled. Do not execute this while the event is in progress (or in the event handler).								
Return	One of the following values is returned by the method and also placed in the ResultCode property:								
<table><tr><th>Value</th><th>Meaning</th></tr><tr><td>OPOS_SUCCESS(0)</td><td>Exclusive access has been released. The Claimed property is now FALSE.</td></tr><tr><td>OPOS_E_ILLEGAL(106)</td><td>The application does not have exclusive access to the device.</td></tr><tr><td>OPOS_E_BUSY(113)</td><td>Asynchronous output is in progress</td></tr></table>		Value	Meaning	OPOS_SUCCESS(0)	Exclusive access has been released. The Claimed property is now FALSE.	OPOS_E_ILLEGAL(106)	The application does not have exclusive access to the device.	OPOS_E_BUSY(113)	Asynchronous output is in progress
Value	Meaning								
OPOS_SUCCESS(0)	Exclusive access has been released. The Claimed property is now FALSE.								
OPOS_E_ILLEGAL(106)	The application does not have exclusive access to the device.								
OPOS_E_BUSY(113)	Asynchronous output is in progress								

CheckHealth Method

Syntax **LONG CheckHealth (LONG Level);**

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

Value	Meaning
OPOS_CH_INTERNAL(1)	Not supported.
OPOS_CH_EXTERNAL(2)	Perform a complete test using the device. IFD version, ServiceObjectVersion and DeviceName are printed on the printer.
OPOS_CH_INTERACTIVE(3)	Perform an interactive test of the device. This Service Object displays the modal dialog box and prints the IFD version, ServiceObjectVersion and DeviceName on the station specified by the device.

Remarks Called to test the state of a device. A text description of the results of this method is placed in the **CheckHealthText** property. The **CheckHealth** method is always synchronous.

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

Value	Meaning
OPOS_SUCCESS(0)	Indicates that the health checking procedure was initiated properly and, when possible to determine, indicates that the device is healthy. However, the health of many devices can only be determined by a visual inspection of the test results.
OPOS_E_NOTCLAIMED(103)	Exclusive access is not available. Call the ClaimDevice method to gain exclusive access.
OPOS_E_DISABLED(105)	Not enabled. Call after setting DeviceEnabled property to TRUE.
OPOS_E_ILLEGAL(106)	The specified health check level is not supported by the Service Object.
OPOS_E_NOHARDWARE(107)	The printer is powered off or the cable is not connected.
OPOS_E_FAILURE(111)	A communication error has occurred.
OPOS_E_TIMEOUT(112)	Data transmission timeout or data response timeout has occurred.
OPOS_E_BUSY(113)	Cannot perform while output is in progress or an error occurs.
OPOS_E_EXTENDED(114)	One of the errors defined by the ResultCodeExtended property except for OPOS_EPTR_TOOBIG(206), OPOS_EPTR_BADFORMAT(207), OPOS_ESTATS_ERROR(280) is notified.

ClearOutput Method

Syntax	LONG ClearOutput ();												
Remarks	Called to clear all buffered output data. Any output error events that were enqueued – usually waiting for that FreezeEvents to be set to FALSE – are also cleared.												
Return	One of the following values is returned by the method and also placed in the ResultCode property:												
<table><tr><th>Value</th><th>Meaning</th></tr><tr><td>OPOS_SUCCESS(0)</td><td>Output has been cleared.</td></tr><tr><td>OPOS_E_NOTCLAIMED(103)</td><td>Exclusive access is not available. Call the ClaimDevice method to gain exclusive access.</td></tr><tr><td>OPOS_E_DISABLED(105)</td><td>Not enabled. Call after setting DeviceEnabled property to TRUE.</td></tr><tr><td>OPOS_E_TIMEOUT(112)</td><td>Data transmission timeout or data response timeout has occurred.</td></tr><tr><td>OPOS_E_EXTENDED(114)</td><td>One of the errors defined by the ResultCodeExtended property except for OPOS_EPTR_TOOBIG(206), OPOS_EPTR_BADFORMAT(207), OPOS_ESTATS_ERROR(280) is notified.</td></tr></table>		Value	Meaning	OPOS_SUCCESS(0)	Output has been cleared.	OPOS_E_NOTCLAIMED(103)	Exclusive access is not available. Call the ClaimDevice method to gain exclusive access.	OPOS_E_DISABLED(105)	Not enabled. Call after setting DeviceEnabled property to TRUE.	OPOS_E_TIMEOUT(112)	Data transmission timeout or data response timeout has occurred.	OPOS_E_EXTENDED(114)	One of the errors defined by the ResultCodeExtended property except for OPOS_EPTR_TOOBIG(206), OPOS_EPTR_BADFORMAT(207), OPOS_ESTATS_ERROR(280) is notified.
Value	Meaning												
OPOS_SUCCESS(0)	Output has been cleared.												
OPOS_E_NOTCLAIMED(103)	Exclusive access is not available. Call the ClaimDevice method to gain exclusive access.												
OPOS_E_DISABLED(105)	Not enabled. Call after setting DeviceEnabled property to TRUE.												
OPOS_E_TIMEOUT(112)	Data transmission timeout or data response timeout has occurred.												
OPOS_E_EXTENDED(114)	One of the errors defined by the ResultCodeExtended property except for OPOS_EPTR_TOOBIG(206), OPOS_EPTR_BADFORMAT(207), OPOS_ESTATS_ERROR(280) is notified.												

CompareFirmwareVersion Method

Syntax	LONG CompareFirmwareVersion (BSTR <i>FirmwareFileName</i>, Long <i>result</i>);	
Remarks	This method is not supported.	
Return	The following value is returned by the method and also 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);**

Parameter	Description
<i>Command</i>	Command number. Specific values assigned by the Service Object.
<i>pData</i>	Pointer to additional numeric data. Specific values vary by Command and Service Object.
<i>pString</i>	Pointer to additional string data. Specific values vary by Command and Service Object. In the OPOS environment, the format of this data depends upon the value of the BinaryConversion property. See BinaryConversion Property for details.

Remarks The following functions are supported.

- Execution Response request setting
- Remaining memory capacity response
- Status response
- International character select

1. Execution Response request setting

<i>Command</i>	PTR_DI_SET_RESPONSE_REQUEST (2)
<i>pData</i>	IN Response code (1 - 15)
<i>pString</i>	Not used
	Informs the fact that Execution Response, which is set in the Execution Response request setting was received. Asynchronous output is also possible. If the registry "ProcessCompletionTiming" is set in "1", OPOS_E_ILLEGAL(106) is returned as the return value because the performance is not supported.

2. Remaining memory capacity response

<i>Command</i>	PTR_DI_GET_REMAINING_MEMORY (3)
<i>pData</i>	Not used
<i>pString</i>	OUT Remaining memory
	Issues the remaining memory response command and returns its response in character (numeral) strings. Asynchronous output is not possible. When AsyncMode =TRUE, synchronous output is possible if the State property is OPOS_S_IDLE(2).

3. Status response

<i>Command</i>	PTR_DI_GET_STATUS_DATA (501)			
<i>pData</i>	IN Status type 1: Status response of paper sensor 3: Send response of presenter			
<i>pString</i>	OUT Status			
	1: Paper sensor status (n=1) Out-of-paper Paper-near-end "00" Paper Paper (when NearEnd is disable) "01" Paper No paper (when NearEnd is enable) "04" No paper Paper (when NearEnd is disable) "05" No paper No paper (when NearEnd is enable) 3: Presenter status (n=3) Paper detection Feed error Jam error "00" No No No "01" Yes No No "04" No Yes No "05" Yes Yes No "20" No No Yes "21" Yes No Yes "24" No Yes Yes "25" Yes Yes Yes When AsyncMode =TRUE, synchronous output is possible if the State property is OPOS_S_IDLE(2). Since the paper end state of paper sensor status is informed as a method error, the status cannot be obtained. If the retrieval of presenter errors is set when the presenter status is obtained, the presenter status cannot be obtained because a method error is informed.			

4. International character selection

<i>Command</i>	PTR_DI_SET_INTERNATIONAL_CHARACTER(201)
<i>pData</i>	IN International character number: n, $0 \leq n \leq 10$
<i>pString</i>	Not used
	<p>Select the character set of each country. International character select by this function continues until the CharacterSet property is changed. When the CharacterSet property is changed, the default value is set. When AsyncMode=TRUE, synchronous output is possible if the State property is OPOS_S_IDLE(2).</p>

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

Value	Meaning
OPOS_SUCCESS(0)	DirectIO successful.
OPOS_E_NOTCLAIMED(103)	Exclusive access is not available. Call the ClaimDevice method to gain exclusive access.

OPOS_E_DISABLED(105)	Not enabled. Call after setting DeviceEnabled property to TRUE.
OPOS_E_ILLEGAL(106)	Parameter has an error or an invalid command number is specified.
OPOS_E_NOHARDWARE(107)	The printer is powered off or the cable is not connected.
OPOS_E_FAILURE(111)	A communication error has occurred.
OPOS_E_TIMEOUT(112)	Data transmission timeout or data response timeout has occurred.
OPOS_E_BUSY(113)	Cannot be performed while output is in progress or an error occurs.
OPOS_E_EXTENDED(114)	One of the errors defined by the ResultCodeExtended property except for OPOS_EPTR_TOOBIG(206), OPOS_EPTR_BADFORMAT(207), OPOS_ESTATS_ERROR(280) is notified.

ResetStatistics Method

Syntax	LONG ResetStatistics (BSTR <i>StatisticsBuffer</i>);	
Remarks	This method is not supported.	
Return	The following value is returned by the method and also placed in the ResultCode property:	
	Value	Meaning
	OPOS_E_ILLEGAL (106)	This method is not supported.

RetrieveStatistics Method

Syntax	LONG RetrieveStatistics (BSTR* <i>pStatisticsBuffer</i>);	
Remarks	This method is not supported.	
Return	The following value is returned by the method and also placed in the ResultCode property:	
	Value	Meaning
	OPOS_E_ILLEGAL(106)	This method is not supported.

UpdateFirmware Method

Syntax	LONG UpdateFirmware (BSTR <i>FirmwareFileName</i>);	
Remarks	This method is not supported.	
Return	The following value is returned by the method and also placed in the ResultCode property:	
	Value	Meaning
	OPOS_E_ILLEGAL(106)	This method is not supported.

UpdateStatistics Method

Syntax	LONG UpdateStatistics (BSTR <i>StatisticsBuffer</i>);	
Remarks	This method is not supported.	
Return	The following value is returned by the method and also placed in the ResultCode property:	
	Value	Meaning
	OPOS_E_ILLEGAL(106)	This method is not supported.

4.6. Specific Methods

PrintNormal Method

Syntax	LONG PrintNormal (LONG <i>Station</i>, BSTR <i>Data</i>);																					
	Parameter	Description																				
	<i>Station</i>	The POS printer to be used. Either PTR_S_JOURNAL(1) or PTR_S_RECEIPT(2) is specified.																				
	<i>Data</i>	The characters to be printed. Consists of printable characters, escape sequences, and line feeds (10 decimal). See BinaryConversion Property for details.																				
Remarks	<p>Call this method to print the <i>Data</i> on POS Printer. The print data that exceeds the maximum number of characters per line is printed on the next print line. If printing data remains in the printer buffer, printing is executed after all the buffered data is printed. This method is performed synchronously if AsyncMode is FALSE, and asynchronously if AsyncMode is TRUE.</p> <p>The special character values within the <i>Data</i> are as follows.</p> <table><tr><th>Value</th><th>Meaning</th></tr><tr><td>Line Feed (10)</td><td>Print any data in the line buffer, and feed to the next print line.</td></tr><tr><td>Carriage Return(13)</td><td>In order to transmit this data, specify "Pass through embedded data" by escape sequence, and execute.</td></tr></table>		Value	Meaning	Line Feed (10)	Print any data in the line buffer, and feed to the next print line.	Carriage Return(13)	In order to transmit this data, specify "Pass through embedded data" by escape sequence, and execute.														
Value	Meaning																					
Line Feed (10)	Print any data in the line buffer, and feed to the next print line.																					
Carriage Return(13)	In order to transmit this data, specify "Pass through embedded data" by escape sequence, and execute.																					
Return	<p>One of the following values is returned by the method and also placed in the ResultCode property:</p> <table><tr><th>Value</th><th>Meaning</th></tr><tr><td>OPOS_SUCCESS(0)</td><td>The method was successful.</td></tr><tr><td>OPOS_E_NOTCLAIMED(103)</td><td>Exclusive access is not available. Call the ClaimDevice method to gain exclusive access.</td></tr><tr><td>OPOS_E_DISABLED(105)</td><td>Not enabled. Call after setting DeviceEnabled property to TRUE.</td></tr><tr><td>OPOS_E_ILLEGAL(106)</td><td>The specified printer does not exist (See the CapJrnPresent and CapRecPresent property).</td></tr><tr><td>OPOS_E_NOHARDWARE(107)</td><td>The printer is powered off or the cable is not connected.</td></tr><tr><td>OPOS_E_FAILURE(111)</td><td>A communication error has occurred.</td></tr><tr><td>OPOS_E_TIMEOUT(112)</td><td>Data transmission timeout or data response timeout has occurred.</td></tr><tr><td>OPOS_E_BUSY(113)</td><td>Cannot be performed while output is in progress or an error occurs. (The State property is set to OPOS_S_BUSY(3) or OPOS_S_ERROR(4) if AsyncMode is FALSE, and to OPOS_S_ERROR(4) if AsyncMode is TRUE.)</td></tr><tr><td>OPOS_E_EXTENDED(114)</td><td>One of the errors defined by the ResultCodeExtended property except for OPOS_EPTR_TOOBIG(206), OPOS_EPTR_BADFORMAT(207), OPOS_ESTATS_ERROR(280) is notified.</td></tr></table>		Value	Meaning	OPOS_SUCCESS(0)	The method was successful.	OPOS_E_NOTCLAIMED(103)	Exclusive access is not available. Call the ClaimDevice method to gain exclusive access.	OPOS_E_DISABLED(105)	Not enabled. Call after setting DeviceEnabled property to TRUE.	OPOS_E_ILLEGAL(106)	The specified printer does not exist (See the CapJrnPresent and CapRecPresent property).	OPOS_E_NOHARDWARE(107)	The printer is powered off or the cable is not connected.	OPOS_E_FAILURE(111)	A communication error has occurred.	OPOS_E_TIMEOUT(112)	Data transmission timeout or data response timeout has occurred.	OPOS_E_BUSY(113)	Cannot be performed while output is in progress or an error occurs. (The State property is set to OPOS_S_BUSY(3) or OPOS_S_ERROR(4) if AsyncMode is FALSE, and to OPOS_S_ERROR(4) if AsyncMode is TRUE.)	OPOS_E_EXTENDED(114)	One of the errors defined by the ResultCodeExtended property except for OPOS_EPTR_TOOBIG(206), OPOS_EPTR_BADFORMAT(207), OPOS_ESTATS_ERROR(280) is notified.
Value	Meaning																					
OPOS_SUCCESS(0)	The method was successful.																					
OPOS_E_NOTCLAIMED(103)	Exclusive access is not available. Call the ClaimDevice method to gain exclusive access.																					
OPOS_E_DISABLED(105)	Not enabled. Call after setting DeviceEnabled property to TRUE.																					
OPOS_E_ILLEGAL(106)	The specified printer does not exist (See the CapJrnPresent and CapRecPresent property).																					
OPOS_E_NOHARDWARE(107)	The printer is powered off or the cable is not connected.																					
OPOS_E_FAILURE(111)	A communication error has occurred.																					
OPOS_E_TIMEOUT(112)	Data transmission timeout or data response timeout has occurred.																					
OPOS_E_BUSY(113)	Cannot be performed while output is in progress or an error occurs. (The State property is set to OPOS_S_BUSY(3) or OPOS_S_ERROR(4) if AsyncMode is FALSE, and to OPOS_S_ERROR(4) if AsyncMode is TRUE.)																					
OPOS_E_EXTENDED(114)	One of the errors defined by the ResultCodeExtended property except for OPOS_EPTR_TOOBIG(206), OPOS_EPTR_BADFORMAT(207), OPOS_ESTATS_ERROR(280) is notified.																					

PrintTwoNormal Method

Syntax	LONG PrintTwoNormal (LONG <i>Stations</i>, BSTR <i>Data1</i>, BSTR <i>Data2</i>);	
Remarks	This method is not supported.	
Return	The following value is returned by the method and also placed in the ResultCode property:	
	Value	Meaning
	OPOS_E_ILLEGAL(106)	This method is not supported.

PrintImmediate Method

Syntax	LONG PrintImmediate (LONG <i>Station</i>, BSTR <i>Data</i>);	
	Parameter	Description
	<i>Station</i>	The POS printer to be used. Either PTR_S_JOURNAL(1) or PTR_S_RECEIPT(2) is specified.
	<i>Data</i>	The characters to be printed. Consists of printable characters, escape sequences and Line feeds (10 decimal). See BinaryConversion Property for details.
Remarks	Call this method to print the <i>Data</i> on the POS Printer immediately. The print data that exceeds the maximum number of characters per line is printed on the next print line. This method tries to print its data immediately without buffering by TransactionPrint method and RotatePrint method.	
	If printing data remains in the printer buffer, printing is executed after all the buffered data is printed. The special character values within the <i>Data</i> are as follows.	
	Value	Meaning
	Line Feed (10)	Print any data in the line buffer, and feed to the next print line.
	Carriage Return(13)	In order to transmit this data, specify the embedded data by escape sequence, and execute.
Return	One of the following values is returned by the method and also placed in the ResultCode property:	
	Value	Meaning
	OPOS_SUCCESS(0)	The method was successful.
	OPOS_E_NOTCLAIMED(103)	Exclusive access is not available. Call the ClaimDevice method to gain exclusive access.
	OPOS_E_DISABLED(105)	Not enabled. Call after setting DeviceEnabled property to TRUE.
	OPOS_E_ILLEGAL(106)	The specified printer does not exist (See the CapJrnPresent and CapRecPresent property).
	OPOS_E_FAILURE(111)	A communication error has occurred.
	OPOS_E_TIMEOUT(112)	Data transmission timeout has occurred.

BeginInsertion Method

Syntax	LONG BeginInsertion (LONG Timeout);	
Remarks	This method is not supported.	
Return	The following value is returned by the method and also placed in the ResultCode property:	
	Value	Meaning
	OPOS_E_ILLEGAL(106)	This method is not supported.

EndInsertion Method

Syntax	LONG EndInsertion ();	
Remarks	This method is not supported.	
Return	The following value is returned by the method and also placed in the ResultCode property:	
	Value	Meaning
	OPOS_E_ILLEGAL(106)	This method is not supported.

BeginRemoval Method

Syntax	LONG BeginRemoval (LONG Timeout);	
Remarks	This method is not supported.	
Return	The following value is returned by the method and also placed in the ResultCode property:	
	Value	Meaning
	OPOS_E_ILLEGAL(106)	This method is not supported.

EndRemoval Method

Syntax	LONG EndRemoval ();	
Remarks	This method is not supported.	
Return	The following value is returned by the method and also placed in the ResultCode property:	
	Value	Meaning
	OPOS_E_ILLEGAL(106)	This method is not supported.

CutPaper Method

Syntax	LONG CutPaper (LONG <i>Percentage</i>);	
	Parameter	Meaning
	<i>Percentage</i>	Indicates the percentage of the paper to be cut. The value 100 causes a full paper cut. Other values between 1 and 99 request a partial cut percentage.
Remarks	<p>Cuts the receipt paper.</p> <p>This method is performed synchronously if AsyncMode is FALSE, and asynchronously if AsyncMode is TRUE.</p> <p>Paper cut can also be performed by using cut paper escape sequence when calling PrintNormal or PrintImmediate methods.</p> <p>If printing data remains in the printer buffer, paper cut is executed after all buffered data is printed.</p> <p>During rotated 90° right/left mode by RotatePrint method and while page mode by PageModePrint method is selected, paper cut is not executed even if the method is not successful. When Journal is specified by the registry, paper cut is not supported.</p> <p>Due to the positions of printer head and cutter, paper cut might be executed at the middle of printing data. To avoid this, call this method after feeding paper for the value of RecLinesToPaperCut property.</p>	
Return	One of the following values is returned by the method and also placed in the ResultCode property:	
	Value	Meaning
	OPOS_SUCCESS(0)	The method was successful.
	OPOS_E_NOTCLAIMED(103)	Exclusive access is not available. Call the ClaimDevice method to gain exclusive access.
	OPOS_E_DISABLED(105)	Not enabled. Call after setting DeviceEnabled property to TRUE.
	OPOS_E_ILLEGAL(106)	An invalid percentage was specified. Select number between 1 and 100. Or, the POS Printer used by the registry is set to one other than Receipt. Or, it is also notified when the method is executed when it is not supported during rotated 90° right/left mode by RotatePrint method or while in the page mode by PageModePrint method.
	OPOS_E_NOHARDWARE(107)	The printer is powered off or the cable is not connected.
	OPOS_E_TIMEOUT(112)	Data transmission timeout has occurred.

OPOS_E_BUSY(113)	Cannot be performed while output is in progress or an error occurs. (The State property is set to OPOS_S_BUSY(3) or OPOS_S_ERROR(4) if AsyncMode is FALSE, and to OPOS_S_ERROR(4) if AsyncMode is TRUE.)
OPOS_E_EXTENDED(114)	One of the errors defined by the ResultCodeExtended property except for OPOS_EPTR_TOOBIG(206), OPOS_EPTR_BADFORMAT(207), OPOS_ESTATS_ERROR(280) is notified.

RotatePrint Method

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

Parameter	Description
<i>Station</i>	The POS printer to be used. PTR_S_RECEIPT(2) is specified.
<i>Rotation</i>	Direction of rotation. See values below.
The values of <i>Rotation</i> are as follows.	
Value	Meaning
PTR_RP_RIGHT90(257)	Start rotated printing 90° to the right (clockwise)
PTR_RP_LEFT90(258)	Start rotated printing 90° to the left (counterclockwise)
PTR_RP_ROTATE180(259)	Start rotated printing 180°, that is, print upside-down
PTR_RP_BARCODE(4096)	Start rotated bar code printing. This value is ORed with one of the above start rotated print values.
PTR_RP_BITMAP(8192)	Start rotated bitmap printing. This value is ORed with one of the above start rotated print values. Rotates the bitmap printed by PrintBitmap .
PTR_RP_NORMAL(1)	End of rotated printing.

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

If *Rotation* is PTR_RP_ROTATE180(259), then upside-down print mode is entered. Subsequent calls to **PrintNormal** or **PrintImmediate** will print the data upside-down until **RotatePrint** is called with *Rotation* set to PTR_RP_NORMAL(1). Each print line is rotated by 180°. Lines are printed in the order that they are sent, with the start of each line justified at the right margin of the printer. If PTR_RP_BARCODE(4096) is set as OR of PTR_RP_ROTATE180(259) in *Rotation*, the bar code printing by **PrintBarCode** method makes upside-down bar code. Also, if PTR_RP_BITMAP(8192) is set as OR of PTR_RP_ROTATE180(259) in *Rotation*, the bitmap printing by **PrintBitmap** method makes upside-down bitmap.

* Caution for rotating 180°

1. If the transaction is performed by **TransactionPrint** method, upside-down print mode is cleared as well as **ClearOutput** method.

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

If PTR_RP_BARCODE (4096) is set in the *Rotation*, bar code printing by **PrintBarCode** method is buffered. If PTR_RP_BITMAP(8192) is set in the *Rotation*, bitmap printing by **PrintBitmap** method is buffered.

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

Service object calculate so that the width in the horizontal writing mode becomes best size. The maximum width in the horizontal writing mode differs depending on the specified paper size. If the print data per line exceeds this range, the width differs depending on the model (Refer to **RecSidewaysMaxChars** property) and non-printed data is printed by feeding to the next print line.

* Caution for rotating 90°

1. If the specific escape sequence is specified to **PrintNormal** method during the horizontal writing mode, its escape sequence is buffered regardless of whether or not PTR_RP_BITMAP(8192) is specified in the *Rotation* parameter as OR.
2. If the transaction by **TransactionPrint** method is started, the horizontal writing mode is cleared as well as **ClearOutput** method. This is because buffering by **TransactionPrint** method is prior to buffering by **RotatePrint** method. In order to batch print in the horizontal writing mode, perform **RotatePrint** method after performing **TransactionPrint** method, then stop rotated print mode by **RotatePrint** method and print after clearing batch print by **TransactionPrint** method.

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

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

Value	Meaning
OPOS_SUCCESS(0)	The method was successful.
OPOS_E_NOTCLAIMED(103)	Exclusive access is not available. Call the ClaimDevice method to gain exclusive access.
OPOS_E_DISABLED(105)	Not enabled. Call after setting DeviceEnabled property to TRUE.
OPOS_E_ILLEGAL(106)	The specified printer does not exist (See CapRecPresent property). Or, rotation specified by <i>Station</i> is not supported. Or, it is also notified when rotated 90° mode is specified during rotated 180° mode or when rotated 180° mode is specified during rotated 90° mode.
OPOS_E_NOHARDWARE(107)	The printer is powered off or the cable is not connected.
OPOS_E_BUSY(113)	Cannot be performed while output is in progress or an error occurs. (The State property is set to OPOS_S_BUSY(3) or OPOS_S_ERROR(4) if AsyncMode is FALSE, and to OPOS_S_ERROR(4) if AsyncMode is TRUE.)
OPOS_E_EXTENDED(114)	One of the errors defined by the ResultCodeExtended property except for OPOS_EPTR_TOOBIG(206), OPOS_EPTR_BADFORMAT(207), OPOS_ESTATS_ERROR(280) is notified.

PrintBarCode Method

Syntax	LONG PrintBarCode (LONG Station, BSTR Data, LONG Symbology, LONG Height, LONG Width, LONG Alignment, LONG TextPosition);	
	Parameter	Description
	<i>Station</i>	The POS printer to be used. PTR_S_RECEIPT(2) is specified.
	<i>Data</i>	Character string of bar code. The format of this data depends upon the value of the BinaryConversion property. See BinaryConversion Property for details.
	<i>Symbology</i>	Bar code type to be used (See the values below).
	<i>Height</i>	Bar code height. Expressed in the unit given by MapMode . For PTR_MM_DOTS(1), specify the value between 1 and 255. For QR Code, although <i>Height</i> is ignored, specify the value between 1 and 255 as well. During page mode by PageModePrint method, specify the value within the range of print area defined by the PageModePrintArea property and the PageModeVerticalPosition property.
	<i>Width</i>	Bar code width. Expressed in the unit given by MapMode . The actual printed bar code width is the best size that fits into the width specified by <i>Width</i> . For PTR_MM_DOTS(1), specify the value between 1 and RecLineWidth . When the RotateSpecial property and printing bar code in a rotated 90° to right/left mode by RotatePrint method are executed, specify the value not exceeding the maximum value of the connected devices (see the PageModeArea property for the maximum value). During page mode by PageModePrint method, specify the value within the range of print area defined by the PageModePrintArea property and the PageModeHorizontalPosition property.
	<i>Alignment</i>	Placement of the bar code. See values below.
	<i>TextPosition</i>	Placement of character string. See values below.
	The values of the <i>Alignment</i> parameter are as follows.	
	Value	Meaning
	PTR_BC_LEFT(-1)	Left justification
	PTR_BC_CENTER(-2)	Centering
	PTR_BC_RIGHT(-3)	Right justification
	* When bar code is rotated 90° to the right/left by the RotateSpecial property, RotatePrint method and PageModePrint method, these setting are ignored and the data is always printed with left justification.	
	The values of the <i>TextPosition</i> parameter are as follows.	
	Value	Meaning
	PTR_BC_TEXT_NONE(-11)	No text is printed. Only print 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.

The values of the *Symbology* parameter are as follows.

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_EAN13_S(119)	EAN13 with supplemental bar code
PTR_BCS_ITF(106)	Interleaved 2 of 5
PTR_BCS_Codabar(107)	Codabar(NW-7)
PTR_BCS_Code39(108)	Code 39
PTR_BCS_Code93(109)	Code 93
PTR_BCS_Code128(110)	Code 128
PTR_BCS_PDF417(201)	PDF417
PTR_BCS_OTHER+1(502)	QR Code (Numeric mode)
PTR_BCS_OTHER+2(503)	QR Code (Alphanumeric mode)
PTR_BCS_OTHER+3(504)	QR Code (8-bit byte mode)
PTR_BCS_OTHER+4(505)	QR Code (Kanji mode)
PTR_BCS_OTHER+5(506)	QR Code (Mixed mode)

Remarks Call this method in order to print the bar code at the specified POS Printer.
This method is performed synchronously if **AsyncMode** is FALSE, and asynchronously if **AsyncMode** is TRUE.

Hereinafter, the limitations for each bar code are described.

[UPC-A]

Allowable Character	Specify 11 or 12 letters consisting of '0' - '9'. The 12th letter does not affect the bar code printing data.
Width Calculation Formula	Bar code length = 113×Narrow width

[UPC-E]

Allowable Character	Specify 11 or 12 letters consisting of '0' - '9'. The 12th letter does not affect the bar code printing data.
Width Calculation Formula	Bar code length = 65×Narrow width

Additionally, the allowable character must follow the rules below.

1. The 1st letter is '0.'
2. The UPC-A left code indicates the 2nd to the 6th characters, the UPC-A right code indicates the 7th to the 11th characters, and the code to be abbreviated is actually printed as UPC-E. If the specified UPC-A initial character is other than 0 or the character which is not included in the following list is, OPOS_E_ILLEGAL(106) is returned.

Maker code UPC-A left code					Item code UPC-A right code					Abbreviated code					
F1	F2	F3	F4	F5	A1	A2	A3	A4	A5	Z1	Z2	Z3	Z4	Z5	Z6
0-9	0-9	0	0	0	0	0	0-9	0-9	0-9	F1	F2	A3	A4	A5	0
0-9	0-9	1	0	0	0	0	0-9	0-9	0-9	F1	F2	A3	A4	A5	1
0-9	0-9	2	0	0	0	0	0-9	0-9	0-9	F1	F2	A3	A4	A5	2
0-9	0-9	3-9	0	0	0	0	0	0-9	0-9	F1	F2	F3	A4	A5	3
0-9	0-9	0-9	1-9	0	0	0	0	0	0-9	F1	F2	F3	F4	A5	4
0-9	0-9	0-9	0-9	1-9	0	0	0	0	5-9	F1	F2	F3	F4	F5	A5

[EAN 8 (= JAN 8)]

Allowable Character	Specify 7 or 8 letters consisting of '0' - '9'. The 8th letter does not affect the bar code printing data.
Width Calculation Formula	Bar code length = $81 \times \text{Narrow width}$

[EAN 13 (= JAN 13)]

Allowable Character	Specify 12 or 13 letters consisting of '0' - '9'. The 13th letter does not affect the bar code printing data.
Width Calculation Formula	Bar code length = $113 \times \text{Narrow width}$

[EAN 13 (= JAN 13) with supplemental bar code]

Allowable Character	Specify 14, 15, 17, or 18 letters consisting of '0' - '9'. When 15 letters or 18 letters are inputted, the 13th character does not affect the printing data.
Width Calculation Formula	Bar code length = $140 \times \text{Narrow width}$ (for 14 or 15 letters) Bar code length = $167 \times \text{Narrow width}$ (for 17 or 18 letters)

[Interleaved 2 of 5]

Allowable Character	Specify any value consisting of '0' - '9'. Note that the number of specified letters must be an even number except for 0.
Width Calculation Formula	Bar code length = $((\text{length} \times 2 + 1) \times \text{wide}) + (((\text{length} \times 3) + 6 + (10 \times 2)) \times \text{narrow})$ * length=number of bar code character, wide=wide width, narrow=narrow width

[Codabar(NW-7)]

Allowable Character	One of 'A' - 'D' must be specified for the end of the 1st line and at least one of '0' - '9', '\$', '+', ':', '-', '.', '/' must be specified.
Width Calculation Formula	Bar code length = $((6 \times \text{narrow}) + (2 \times \text{wide})) \times \text{length} + ((\text{wide} - \text{narrow}) \times \text{wlen}) + (\text{narrow} \times (10 \times 2 - 1))$ *length=number of bar code character, wide=wide width, narrow=narrow width, wlen=the number of the characters 'A' - 'D', '+', ':', '-', '/', and '.' included in the bar code characters.

[Code 39]

Allowable Character	At least 1 letter consisting of '0' - '9', 'A' - 'Z', 'space', '\$', '%', '+', '-', '!', '/' must be specified.
Width Calculation Formula	Bar code length = (((narrow×7) + (wide×3))×(length + 2)) + ((10×2 - 1) ×narrow) * length=number of bar code character, wide=wide width, narrow=narrow width

[Code 93]

Allowable Character	See the parameter list below.
Width Calculation Formula	Bar code length = narrow×((10×2) + ((length + 2 + 2)×9) + 1) * length= number of bar code character, narrow=narrow width

* When printing Code93, if all the configurable data parameter are specified, the value of the **BinaryConversion** property must be specified by OPOS_BC_NIBBLE(1) or OPOS_BC_DECIMAL(2). This is because the data includes the symbology character. Regarding the decimal numbers specified for *Data*, 0x00 represents 0 and 0x01 represents 1.

Decimal number for specified for Data	Printing character	Decimal number for specified for Data	Printing character
0	0	24	O
1	1	25	P
2	2	26	Q
3	3	27	R
4	4	28	S
5	5	29	T
6	6	30	U
7	7	31	V
8	8	32	W
9	9	33	X
10	A	34	Y
11	B	35	Z
12	C	36	-
13	D	37	.
14	E	38	SPACE
15	F	39	\$
16	G	40	/
17	H	41	+
18	I	42	%
19	J	43	(\$)
20	K	44	(%)
21	L	45	(/)
22	M	46	(+)
23	N		

[Code 128]

Allowable Character	See the parameter list below. The first letter of the first line should be decimal number, 103(0x67),104(0x68),105(0x69), after that at least one letter should be included.
Width Calculation Formula	Bar code length = narrow×((10×2) + ((length + 2 + 1 + 1)×11) + 2) * length = number of bar code character minus 1, narrow = narrow width

* When printing Code128, if all the configurable data parameter are specified, the value of the **BinaryConversion** property must be specified by OPOS_BC_NIBBLE(1) or OPOS_BC_DECIMAL(2). This is because the data includes symbology character. Regarding the decimal numbers specified for *Data*, 0x00 represents 0 and 0x01 represents 1.

Decimal Number for Specified for Data	Printing Character			Decimal Number for Specified for Data	Printing Character		
	CODE-A	CODE-B	CODE-C		CODE-A	CODE-B	CODE-C
0	SPACE	SPACE	00	53	U	U	53
1	!	!	01	54	V	V	54
2	"	"	02	55	W	W	55
3	#	#	03	56	X	X	56
4	\$	\$	04	57	Y	Y	57
5	%	%	05	58	Z	Z	58
6	&	&	06	59	[[59
7	'	'	07	60	/	/	60
8	((08	61]]	61
9))	09	62	^	^	62
10	*	*	10	63	_	_	63
11	+	+	11	64	NULL	`	64
12	,	,	12	65	SOH	a	65
13	-	-	13	66	STX	b	66
14	.	.	14	67	ETX	c	67
15	/	/	15	68	EOT	d	68
16	0	0	16	69	ENG	e	69
17	1	1	17	70	ACK	f	70
18	2	2	18	71	BEL	g	71
19	3	3	19	72	BS	h	72
20	4	4	20	73	HT	i	73
21	5	5	21	74	LF	j	74
22	6	6	22	75	VT	k	75
23	7	7	23	76	FF	l	76
24	8	8	24	77	CR	m	77
25	9	9	25	78	SO	n	78
26	:	:	26	79	SI	o	79
27	;	;	27	80	DLE	p	80
28	<	<	28	81	DC1	q	81
29	=	=	29	82	DC2	r	82
30	>	>	30	83	DC3	s	83
31	?	?	31	84	DC4	t	84
32	@	@	32	85	NAK	u	85
33	A	A	33	86	SYN	v	86
34	B	B	34	87	ETB	w	87

35	C	C	35	88	CAN	x	88
36	D	D	36	89	EM	y	89
37	E	E	37	90	SUB	z	90
38	F	F	38	91	ESC	{	91
39	G	G	39	92	FS		92
40	H	H	40	93	GS	}	93
41	I	I	41	94	RS	~	94
42	J	J	42	95	US	DEL	95
43	K	K	43	96	FNC 3	FNC 3	96
44	L	L	44	97	FNC 2	FNC 2	97
45	M	M	45	98	SHIFT	SHIFT	98
46	N	N	46	99	CODE C	CODE C	99
47	O	O	47	100	CODE B	FNC 4	CODE B
48	P	P	48	101	FNC 4	CODE A	CODE A
49	Q	Q	49	102	FNC 1	FNC 1	FNC 1
50	R	R	50	103	START(CODE A)		
51	S	S	51	104	START(CODE B)		
52	T	T	52	105	START(CODE C)		

[QR Code]

Allowable Character	Acceptable character string on each mode of <i>Symbology</i> parameter is as below											
	<table> <tr> <th>Mode</th><th>Printable character</th></tr> <tr> <td>Numeric mode</td><td>10 numbers (0 – 9)</td></tr> <tr> <td>Alphanumeric mode</td><td>10 numbers (0 – 9), 26 alphabets (A – Z), 9 symbols (Space, \$, %, *, +, -, ., /, :)</td></tr> <tr> <td>8-bit byte mode</td><td>8 bits Latin/Kana characters based on JIS X 0201 (ASCII 0x00 - 0xFF).</td></tr> <tr> <td>Kanji mode</td><td>Shift JIS code based on JIS X 0208</td></tr> <tr> <td>Mixed mode</td><td>Characters of a combination of any modes above.</td></tr> </table>	Mode	Printable character	Numeric mode	10 numbers (0 – 9)	Alphanumeric mode	10 numbers (0 – 9), 26 alphabets (A – Z), 9 symbols (Space, \$, %, *, +, -, ., /, :)	8-bit byte mode	8 bits Latin/Kana characters based on JIS X 0201 (ASCII 0x00 - 0xFF).	Kanji mode	Shift JIS code based on JIS X 0208	Mixed mode
Mode	Printable character											
Numeric mode	10 numbers (0 – 9)											
Alphanumeric mode	10 numbers (0 – 9), 26 alphabets (A – Z), 9 symbols (Space, \$, %, *, +, -, ., /, :)											
8-bit byte mode	8 bits Latin/Kana characters based on JIS X 0201 (ASCII 0x00 - 0xFF).											
Kanji mode	Shift JIS code based on JIS X 0208											
Mixed mode	Characters of a combination of any modes above.											
Width Calculation Formula	<p>Bar code length = $(4V + 17) \times M + (4M \times 2)$</p> <p>V: Version of QR Code (1 - 40)</p> <p>M: Module size (2 - 16)</p> <p>* For version, the smallest value that input data can be converted to bar code is selected. For module size, the maximum size that does not exceed the width parameter is selected after the version is determined.</p>											

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

For each mode, if the data other than the printable characters is specified,

OPOS_E_ILLEGAL(106) is informed.

[PDF417]

Allowable Character	0x00 to 0x7F must follow the ASCII code and 0x80 to 0xFF must follow the extended character set of PC437 English list.
Width Calculation Formula	<p>Bar code length = $(17C + 69) \times X + (2X \times 2)$</p> <p>Bar code height = $RY + (2X \times 2)$</p> <p>C: number of column</p> <p>X: nominal fine element width</p> <p>R: number of row</p> <p>Y: height of row</p> <p>* For the number of row and the number of column, the smallest value that input data can be converted to bar code is selected. The maximum size that does not exceed the <i>Width</i> and <i>Height</i> parameter is selected after the nominal fine element width, the height of the row, the number of the row, and the number of the column are determined.</p>

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

Return

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

Value	Meaning
OPOS_SUCCESS(0)	The method was successful.
OPOS_E_NOTCLAIMED(103)	Exclusive access is not available. Call the ClaimDevice method to gain exclusive access.
OPOS_E_DISABLED(105)	Not enabled. Call after setting DeviceEnabled property to TRUE.
OPOS_E_ILLEGAL(106)	One of the following errors has occurred. <ul style="list-style-type: none"> - <i>Station</i> does not exist. - <i>Station</i> does not support bar code printing. - <i>Height</i> or <i>Width</i> is 0 or too large. - This <i>Symbology</i> is not supported. - The values of <i>Alignment</i> is improper. - The values of <i>TextPosition</i> is improper. - The specified data is not printable. - The data stored in the specified <i>Width</i> parameter cannot be printed.
OPOS_E_NOHARDWARE(107)	The printer is powered off or the cable is not connected.
OPOS_E_TIMEOUT(112)	Data transmission timeout has occurred.
OPOS_E_BUSY(113)	Cannot be performed while output is in progress or an error occurs. (The State property is set to OPOS_S_BUSY(3) or OPOS_S_ERROR(4) if AsyncMode is FALSE, and to OPOS_S_ERROR(4) if AsyncMode is TRUE.)
OPOS_E_EXTENDED(114)	One of the errors defined by the ResultCodeExtended property except for OPOS_EPTR_TOOBIG(206), OPOS_EPTR_BADFORMAT(207), OPOS_ESTATS_ERROR(280) is notified.

PrintBitmap Method

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

Parameter	Description
<i>Station</i>	The POS printer to be used. PTR_S_RECEIPT(2) is specified.
<i>FileName</i>	The name of Windows bitmap file. The compressed format is not supported (Full path or relative path must be specified). Monochrome or full color (24 bit) format file can be specified.
<i>Width</i>	Print width of bitmap. See values below.
<i>Alignment</i>	Position of bitmap. See values below.

The values of Width parameter are as follows.

Value	Meaning
PTR_BM_ASIS(-11)	Print the bitmap with one bitmap pixel per printer dot of the POS Printer.
<i>Other values</i>	Bitmap width. Expressed in the unit given by MapMode . If MapMode is PTR_MM_DOTS(1), specify between 1 and RecLineWidth property. When printing bitmap in a rotated 90° to right/left mode by RotatePrint method is executed, specify the value not exceeding the maximum value of the connected devices (see the PageModeArea property for the maximum value). During page mode by PageModePrint method, specify the value within the range of print area defined by the PageModePrintArea property and the PageModeHorizontalPosition property.

* The value of bitmap width can be rounded up to a multiple of 8 within OPOS OCX. Specify the bitmap width within the range so that the converted value does not exceed the print area.

The values of the *Alignment* parameter are as follows.

Value	Meaning
PTR_BM_LEFT(-1)	Left justification
PTR_BM_CENTER(-2)	Centering
PTR_BM_RIGHT(-3)	Right justification
<i>Other values</i>	Distance from the left edge where bitmap printing starts. Expressed in the unit given by MapMode .

* During rotated 90° right/left mode by **RotatePrint** method and **PageModePrint** method, the setting of *Alignment* parameter is invalid and the data is always aligned with left justification.

Remarks This method is called when bitmap is printed on the specified printer.
The highest performance cannot be achieved since the Bitmap data is transferred to the printer after **PrintBitmap** is called. It is recommended to print the Bitmap data using **SetBitmap** and escape sequence.
This method is performed synchronously if **AsyncMode** is FALSE, and asynchronously if **AsyncMode** is TRUE.
The *Width* parameter controls the transformation of bitmap data. If *Width* is

PTR_BM_ASIS(-11), then no transformation is performed. The bitmap is printed with one bitmap pixel per dot of the POS Printer.

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

Return 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 successful.
OPOS_E_NOTCLAIMED(103)	Exclusive access is not available. Call the ClaimDevice method to gain exclusive access.
OPOS_E_DISABLED(105)	Not enabled. Call after setting DeviceEnabled property to TRUE.
OPOS_E_ILLEGAL(106)	One of the following parameter errors occurred. - <i>Station</i> does not exist. - <i>Station</i> does not support bitmap printing - <i>Width</i> is too large. - <i>Alignment</i> is invalid or too big. When position is specified during normal or upside-down mode, the total including print bitmap width exceeds the range of RecLineWidth property. - 32 bit colored bitmap is set.
OPOS_E_NOHARDWARE(107)	The printer is powered off or the cable is not connected.
OPOS_E_NOEXIST(109)	The file specified by <i>FileName</i> was not found.
OPOS_E_TIMEOUT(112)	Data transmission timeout has occurred.
OPOS_E_BUSY(113)	Cannot be performed while output is in progress or an error occurs. (The State property is set to OPOS_S_BUSY(3) or OPOS_S_ERROR(4) if AsyncMode is FALSE, and to OPOS_S_ERROR(4) if AsyncMode is TRUE.)
OPOS_E_EXTENDED(114)	One of the errors defined by the ResultCodeExtended property except for OPOS_ESTATS_ERROR(280) are notified.

TransactionPrint Method

Syntax	LONG TransactionPrint(LONG Station, LONG Control);																	
	Parameter	Description																
	Station	The POS printer to be used. Either PTR_S_JOURNAL(1) or PTR_S_RECEIPT(2) is specified.																
	Control	Transaction control. See values below.																
	The values of Control are as follows.																	
	Value	Meaning																
	PTR_TP_TRANSACTION(11)	Start of transaction.																
	PTR_TP_NORMAL(12)	Ends a transaction by printing the buffered data.																
Remarks	<p>Call this method to enter or exit transaction mode.</p> <p>If Control is PTR_TP_TRANSACTION, then transaction mode is entered. Subsequent calls to PrintNormal, CutPaper, RotatePrint, PrintBarCode, and PrintBitmap will buffer the print data at the Service Object until TransactionPrint is called with the Control parameter set to PTR_TP_NORMAL(12). (In this case, the print methods only validate the method parameters and buffer the data – they do not initiate printing. Also, the value of AsyncMode property does not affect its operation. In other words, no OutputID is assigned and no OutputCompleteEvent is informed.)</p> <p>If Control is PTR_TP_NORMAL(12), then transaction mode is exited. If some data was buffered by calls to the methods PrintNormal, CutPaper, RotatePrint, PrintBarCode, and PrintBitmap, then the buffered data is printed. This method is performed synchronously if AsyncMode is FALSE, and asynchronously if AsyncMode is TRUE.</p> <p>Calling ClearOutput method cancels transaction mode. Any buffered print lines are also cleared.</p> <p>For the combination with RotatePrint method, refer to RotatePrint method.</p>																	
Return	<p>One of the following values is returned by the method and also placed in the ResultCode property:</p> <table><tr><td>Value</td><td>Meaning</td></tr><tr><td>OPOS_SUCCESS(0)</td><td>The method is successful.</td></tr><tr><td>OPOS_E_NOTCLAIMED(103)</td><td>Exclusive access is not available. Call the ClaimDevice method to gain exclusive access.</td></tr><tr><td>OPOS_E_DISABLED(105)</td><td>Not enabled. Call after setting DeviceEnabled property to TRUE.</td></tr><tr><td>OPOS_E_ILLEGAL(106)</td><td>The specified printer does not exist (See the CapJrnPresent and CapRecPresent property).</td></tr><tr><td>OPOS_E_NOHARDWARE(107)</td><td>The printer is powered off or the cable is not connected.</td></tr><tr><td>OPOS_E_BUSY(113)</td><td>Cannot be performed while output is in progress or an error occurs. (The State property is set to OPOS_S_BUSY(3) or OPOS_S_ERROR(4) if AsyncMode is FALSE, and to OPOS_S_ERROR(4) if AsyncMode is TRUE.)</td></tr><tr><td>OPOS_E_EXTENDED(114)</td><td>One of the errors defined by the ResultCodeExtended property except for OPOS_EPTR_TOOBIG(206), OPOS_EPTR_BADFORMAT(207), OPOS_ESTATS_ERROR(280) is notified.</td></tr></table>		Value	Meaning	OPOS_SUCCESS(0)	The method is successful.	OPOS_E_NOTCLAIMED(103)	Exclusive access is not available. Call the ClaimDevice method to gain exclusive access.	OPOS_E_DISABLED(105)	Not enabled. Call after setting DeviceEnabled property to TRUE.	OPOS_E_ILLEGAL(106)	The specified printer does not exist (See the CapJrnPresent and CapRecPresent property).	OPOS_E_NOHARDWARE(107)	The printer is powered off or the cable is not connected.	OPOS_E_BUSY(113)	Cannot be performed while output is in progress or an error occurs. (The State property is set to OPOS_S_BUSY(3) or OPOS_S_ERROR(4) if AsyncMode is FALSE, and to OPOS_S_ERROR(4) if AsyncMode is TRUE.)	OPOS_E_EXTENDED(114)	One of the errors defined by the ResultCodeExtended property except for OPOS_EPTR_TOOBIG(206), OPOS_EPTR_BADFORMAT(207), OPOS_ESTATS_ERROR(280) is notified.
Value	Meaning																	
OPOS_SUCCESS(0)	The method is successful.																	
OPOS_E_NOTCLAIMED(103)	Exclusive access is not available. Call the ClaimDevice method to gain exclusive access.																	
OPOS_E_DISABLED(105)	Not enabled. Call after setting DeviceEnabled property to TRUE.																	
OPOS_E_ILLEGAL(106)	The specified printer does not exist (See the CapJrnPresent and CapRecPresent property).																	
OPOS_E_NOHARDWARE(107)	The printer is powered off or the cable is not connected.																	
OPOS_E_BUSY(113)	Cannot be performed while output is in progress or an error occurs. (The State property is set to OPOS_S_BUSY(3) or OPOS_S_ERROR(4) if AsyncMode is FALSE, and to OPOS_S_ERROR(4) if AsyncMode is TRUE.)																	
OPOS_E_EXTENDED(114)	One of the errors defined by the ResultCodeExtended property except for OPOS_EPTR_TOOBIG(206), OPOS_EPTR_BADFORMAT(207), OPOS_ESTATS_ERROR(280) is notified.																	

ValidateData Method

Syntax	LONG ValidateData(LONG Station, BSTR Data);	
	Parameter	Description
	<i>Station</i>	The POS printer to be used. Either PTR_S_JOURNAL(1) or PTR_S_RECEIPT(2) is specified.
	<i>Data</i>	The data to be validated. May include printable data and escape sequences. See BinaryConversion Property for details.
Remarks	Before calling the PrintImmediate or PrintNormal methods, this method is called when determining whether a data sequence, which possibly including one or more escape sequences, is valid for the specified POS Printer. This method does not cause any printing but is used to determine the capability of the POS Printer.	
Return	One of the following values is returned by the method and also placed in the ResultCode property:	
	Value	Meaning
	OPOS_SUCCESS(0)	The data is valid.
	OPOS_E_NOTCLAIMED(103)	Exclusive access is not available. Call the ClaimDevice method to gain exclusive access.
	OPOS_E_DISABLED(105)	Not enabled. Call after setting DeviceEnabled property to TRUE.
	OPOS_E_ILLEGAL(106)	At least one of the escape sequences is out of the range, but Control can select valid alternatives. Also, this value is stored when the escape sequence is not supported by the Page Mode function or rotated 90° to the left or right print mode.
	OPOS_E_NOHARDWARE(107)	The printer is powered off or the cable is not connected.
	OPOS_E_FAILURE(111)	At least one of the escape sequences is not supported. No alternatives can be selected.
	The cases in which ResultCode of OPOS_E_ILLEGAL(106) is returned are as follows.	
	Escape Sequence	Condition
	Paper Cut	One of the following statuses occurs. - Percentage '#' is not precisely supported. - It is not supported during rotated 90° right/left mode by RotatePrint method. - It is not supported during page mode by PageModePrint method.
	Feed and Paper cut	One of the following statuses occurs. - Percentage '#' is not precisely supported. - It is not supported during rotated 90° right/left mode by RotatePrint method. - It is not supported during page mode by PageModePrint method.
	Bitmap print	The bitmap number '#' is not precisely supported.

Feed lines	<p>One of the following statuses occurs.</p> <ul style="list-style-type: none"> - The line number '#' is not precisely supported. - It is not supported during rotated 90° right/left mode by RotatePrint method. - It is not supported during page mode by PageModePrint method.
Feed unit	<ul style="list-style-type: none"> - Feed unit number '#' is not precisely supported due to occurrence of rounding error of one dot depending on the setting of the MapMode property. - The feed unit number '#' is not precisely supported. - It is not supported during rotated 90° right/left mode by RotatePrint method. - It is not supported during page mode by PageModePrint method.
Pass through embedded data	The number of bytes of embedded data '#' is not precisely supported.
Underline	The thickness '#' is not precisely supported.
Vertical scale	The scale factor '#' is not precisely supported
Horizontal scale	The scale factor '#' is not precisely supported
Centering	<p>One of the following statuses occurs.</p> <ul style="list-style-type: none"> - It is not supported during rotated 90° right/left mode by RotatePrint method. - It is not supported during page mode by PageModePrint method.
Right justification	<p>One of the following statuses occurs.</p> <ul style="list-style-type: none"> - It is not supported during rotated 90° right/left mode by RotatePrint method. - It is not supported during page mode by PageModePrint method.

The cases in which OPOS_E_FAILURE(111) is returned are as follows.

Escape Sequence	Condition
Paper Cut	<i>Station</i> does not support paper cut.
Feed and Paper cut	<i>Station</i> does not support paper cut.
Feed, Paper cut, and Stamp	Not supported.
Bitmap print	<i>Station</i> does not support bitmap printing
Stamp print	Not supported.
Reverse feed	Not supported.
Font typeface	Not supported.
Italic	Not supported.
Custom color	Not supported.
Red color	Not supported.
Shaded character	Not supported.
Color option	Not supported.
SubScript	Not supported.
SuperScript	Not supported.

SetBitmap Method

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

Parameter	Description
<i>BitmapNumber</i>	The number to be assigned to this bitmap. The valid bitmap numbers are 1 through 20.
<i>Station</i>	The POS Printer to be used. PTR_S_RECEIPT(2) is specified.
<i>FileName</i>	The name of Windows bitmap file. The compressed format is not supported Monochrome and full color (24bit) format is also available. If "(empty string)" is set, the bitmap setting is canceled.
<i>Width</i>	Print width of bitmap. See PrintBitmap for values.
<i>Alignment</i>	Position of bitmap. See PrintBitmap for values.

The values of the *Width* parameter are as follows.

Value	Meaning
PTR_BM_ASIS(-11)	Print the bitmap with one bitmap pixel per printer dot of the POS Printer.
<i>Other values</i>	Bitmap width. Expressed in the unit given by MapMode . If MapMode is PTR_MM_DOTS(1), specify between 1 and RecLineWidth property.

The values of the *Alignment* parameter are as follows.

Value	Meaning
PTR_BM_LEFT(-1)	Left justification
PTR_BM_CENTER(-2)	Centering
PTR_BM_RIGHT(-3)	Right justification
<i>Other values</i>	Distance from the left edge where bitmap printing starts. Expressed in the unit given by MapMode .

* During rotating 90° to the right/left by **RotatePrint** and **PageModePrint** methods, the setting of *Alignment* parameter is invalid and the data is always aligned with left justification.

Remarks Call this method to save the information about the bitmap to be printed soon. The bitmap may then be printed by calling the **PrintNormal** or **PrintImmediate** methods with the print bitmap escape sequence in the print data. Service object prepares printing with downloading bitmap data in the downloaded bit image area of the printer and NV bit image area. When bitmap print is specified by escape sequence, only command which conducts printing is transmitted to provide better performance.

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

Value	Meaning
OPOS_SUCCESS(0)	The method was successful.
OPOS_E_NOTCLAIMED(103)	Exclusive access is not available. Call the ClaimDevice method to gain exclusive access.

OPOS_E_DISABLED(105)	Not enabled. Call after setting DeviceEnabled property to TRUE.
OPOS_E_ILLEGAL(106)	One of the following parameter errors occurred. <ul style="list-style-type: none"> - The values of <i>BitmapNumber</i> is improper. - <i>Station</i> does not exist. - <i>Station</i> does not support bitmap printing - <i>Width</i> is too large. - <i>Alignment</i> is invalid or too big. When absolute position is specified, the total including print bitmap width exceeds the range of RecLineWidth. - 32-bit colored bitmap is set.
OPOS_E_NOHARDWARE(107)	The printer is powered off or the cable is not connected.
OPOS_E_NOEXIST(109)	<i>FileName</i> was not found.
OPOS_E_FAILURE(111)	Failed to register bitmap due to the shortage of memory on the printer.
OPOS_E_TIMEOU(112)	Data transmission timeout has occurred or any response from the printer is not made before timeout.
OPOS_E_BUSY(113)	Cannot be perform while output is in progress (This can be set when the State property is OPOS_S_BUSY(3) or OPOS_S_ERROR(4)).
OPOS_E_EXTENDED(114)	One of the errors defined by the ResultCodeExtended property except for OPOS_ESTATS_ERROR(280) is notified.

SetLogo Method

Syntax	LONG SetLogo (LONG <i>Location</i>, BSTR <i>Data</i>);	
	Parameter	Description
	<i>Location</i>	The logo to be set. PTR_L_TOP(1) or PTR_L_BOTTOM(2).
	<i>Data</i>	The characters that produce the logo. Consists of printable characters, escape sequences and line feeds (10 decimal). See BinaryConversion Property for details.
Remarks	Call this method to save a data string as a top or bottom logo. The logo is printed by calling the PrintNormal or PrintImmediate methods including the escape sequence of the top or bottom logo into the print data. The <i>Data</i> registered by this method is kept by the character of CharacterSet property at the time when the method is performed. Therefore, SetLogo method is executed by the CharacterSet property of 932 to register the data including Kanji character(s). After that, kanji character(s) will be printed even if the CharacterSet property is set to 999 when the data is printed by escape sequence.	
Return	One of the following values is returned by the method and also placed in the ResultCode property:	
	Value	Meaning
	OPOS_SUCCESS(0)	The method was successful.
	OPOS_E_ILLEGAL(106)	Improper <i>Location</i> is specified.
	<i>Other values</i>	See ResultCode .

ChangePrintSide Method

Syntax	LONG ChangePrintSide (LONG <i>Side</i>);	
Remarks	This method is not supported.	
Return	The following value is returned by the method and also placed in the ResultCode property:	
	Value	Meaning
	OPOS_E_ILLEGAL(106)	This method is not supported.

MarkFeed method

Syntax	LONG MarkFeed (LONG <i>Type</i>);	
Remarks	This method is not supported.	
Return	The following value is returned by the method and also placed in the ResultCode property:	
	Value	Meaning
	OPOS_E_ILLEGAL(106)	This method is not supported.

ClearPrintArea Method

Syntax	LONG ClearPrintArea ();	
Remarks	Clears the print data on the page mode print area defined by the PageModePrintArea property. The entire page mode area can be cleared by calling ClearPrintArea method. A valid station must be set to the PageModeStation property before calling this method.	
Return	One of the following values is returned by the method and also placed in the ResultCode property:	
	Value	Meaning
	OPOS_SUCCESS (0)	The method was successful.
	<i>Other values</i>	See ResultCode .

PageModePrint Method

Syntax **LONG PageModePrint (LONG Control);**

Control parameters are as follows.

Value	Meaning
PTR_PM_PAGE_MODE(1)	Starts page mode.
PTR_PM_PRINT_SAVE(2)	Prints the print data of the page mode print area and save the data. This is used for repeated printings.
PTR_PM_NORMAL(3)	Prints the print data of the page mode print area, clears the data and ends page mode.
PTR_PM_CANCEL(4)	Clear the print data of the page mode print and ends the page mode without any printing.

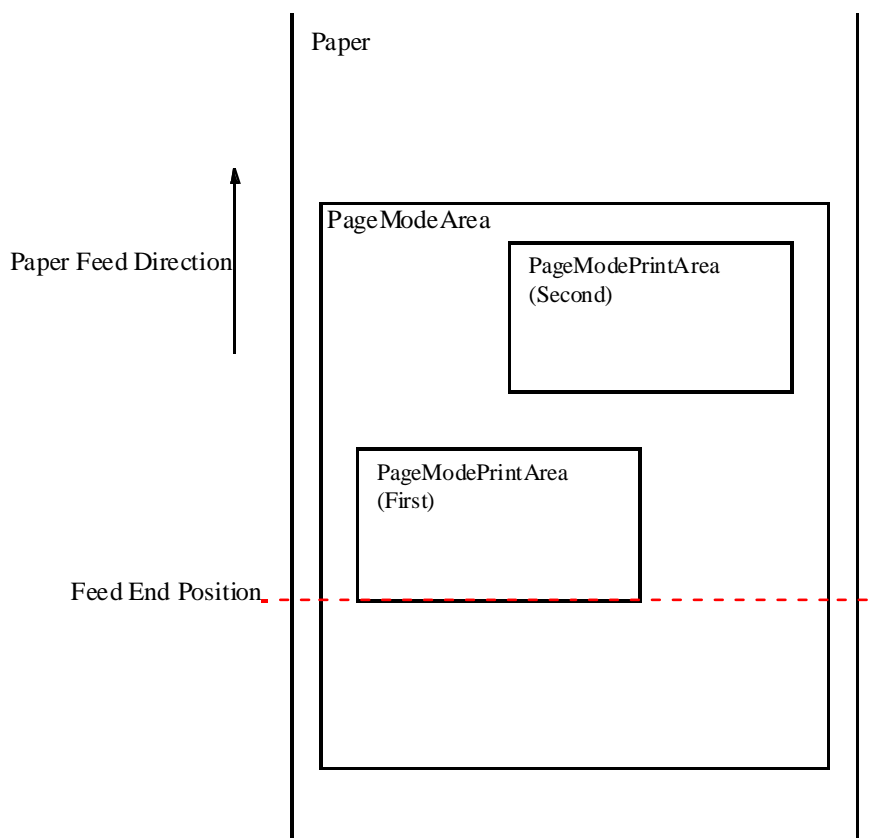
Remarks Starts or ends Page Mode for the station specified for the **PageModeStation** property. If PTR_PM_PAGE_MODE(1) is specified for *Control*, then Page Mode is started. After that, the print data can be buffered using **PrintNormal**, **PrintBarCode**, or **PrintBitmap** methods (either in the printer or the Service Object, depending on the printer capabilities) until **PageModePrint** is called by specifying PTR_PM_PRINT_SAVE(2), PTR_PM_NORMAL(3), or PTR_PM_CANCEL(4). (In this case, the print method called during this only buffers the print data and does not start printing. Also, the setting of the **AsyncMode** property does not affect the page mode function. No **OutputID** will be assigned and no **OutputCompleteEvent** will be notified according to each operation.)

If PTR_PM_PRINT_SAVE(2) is specified for *Control*, then Page Mode is continued. If some data is buffered by one of **PrintNormal**, **PrintBarCode** and **PrintBitmap** methods, then the data is saved and printed. This control is used to print the data with the same page layout by adding the print data into the page mode area.

If PTR_PM_NORMAL(3) is specified for *Control*, then Page Mode is ended to return to the normal state. If some data is buffered by one of **PrintNormal**, **PrintBarCode**, and **PrintBitmap** methods, then the data is printed. The buffered data will not be saved.

If PTR_PM_CANCEL(4) is specified for *Control*, then Page Mode is ended to return to the normal state. If some data is buffered by one of **PrintNormal**, **PrintBarCode**, and **PrintBitmap** methods, the buffered data is not printed and saved.

Note that when the **PageModePrint** method is called, all of the print data on the page mode print area defined by **PageModePrintArea** is printed and the paper is fed to the end of the area. If multiple page mode print areas are defined, then all the print data on each page mode print area is printed and the paper is fed to the end of the area located at the lower side (see figure below).



The entire page mode area is treated as one transaction. This method is performed synchronously if the **AsyncMode** property is FALSE or asynchronously if the property is TRUE.

Calling the **ClearOutput** method cancels page mode to return to the normal state. The buffered print data are also cleared.

Although the page mode function can be used within transaction print, it cannot be used within rotated print.

A valid station must be set to the **PageModeStation** property before calling this method.

Return

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

Value	Meaning
OPOS_SUCCESS(0)	The method was successful.
OPOS_E_ILLEGAL(106)	The station specified by the PageModeStation property does not exist (see the CapRecPresent property) or the CapRecPageMode property is FALSE. Or, while the station specified by the PageModeStation property is in the state before the transition to the page mode, PTR_PM_NORMAL(3), PTR_PM_PRINT_SAVE(4), or PTR_PM_CANCEL(4) is specified for the <i>Control</i> parameter.
OPOS_E_NOHARDWARE(107)	The printer is powered off or the cable is not connected.

OPOS_E_BUSY(113)	Cannot be performed because output is in progress (It can only be returned when AsyncMode is FALSE and PTR_PM_NORMAL(3), PTR_PM_PRINT_SAVE(2), or PTR_PM_CANCEL(4) is specified for the <i>Control</i> parameter).
OPOS_E_EXTENDED(114)	One of the errors defined by the ResultCodeExtended property except for OPOS_EPTR_TOOBIG(206), OPOS_EPTR_BADFORMAT(207), OPOS_ESTATS_ERROR(280) is notified.
<i>Other values</i>	See ResultCode .

4.7. Events

DirectIOEvent Event

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

Parameter	Description
<i>EventNumber</i>	Event number. Specific values assigned by the Service Object.
<i>pData</i>	Pointer to additional numeric data. This specific value varies according to the event number and the Service Object.
<i>pString</i>	The pointer of character string.

Remarks This event is notified by Service Object to directly communicate with the application. This event allows Service Object to provide events that are not supported by Control Object to the application.

If the registry "ProcessCompletionTiming" is set in "1", this event is not notified.

The followings are supported.

Execution Response request reception

EventNumber	2
pData	Response code (1 - 15)
pString	Not used
	Informs the reception of execution response, which is set in Execution Response request setting on DirectIO .

ErrorEvent Event

Syntax	void ErrorEvent (LONG ResultCode, LONG ResultCodeExtended, LONG ErrorLocus, LONG* pErrorResponse);	
	Parameter	Description
	ResultCode	Factor code causing the error event. See Result Code for the values.
	ResultCodeExtended	Extended code of the factor causing the error event. See the values below.
	ErrorLocus	Set to OPOS_EL_OUTPUT(1). The error occurred while processing asynchronous output.
	pErrorResponse	Pointer for error event response. See the values below.
	If ResultCode is OPOS_E_EXTENDED(114), then ResultCodeExtended is set to one of the following values.	
	Value	Meaning
	OPOS_EPTR_COVER_OPEN(201)	Cover is open.
	OPOS_EPTR_JRN_EMPTY(202)	The journal is out of paper.
	OPOS_EPTR_REC_EMPTY(203)	The receipt is out of paper.
	OPOS_EPTR_VPPower(1001)	Vp voltage error has occurred.
	OPOS_EPTR_AUTOCUTTER(1002)	Autocutter error has occurred.
	OPOS_EPTR_HEADHOT(1003)	Head-temperature error has occurred.
	The content of the location specified by the <i>pErrorResponse</i> is preset to the default value of OPOS_ER_RETRY(11). The application sets one of the following values.	
	Value	Meaning
	OPOS_ER_RETRY(11)	Retry the asynchronous output. The error state is exited. If "ProcessCompletionTiming" is set in "0", this event prints the data not being transmitted to the printer. If "ProcessCompletionTiming" is set in "1", this event prints the unprinted data.
	OPOS_ER_CLEAR(12)	Clear the asynchronous output. The error state is exited.
Remarks	The error is detected and is notified when the POS Printer control state shifts to the error state.	

OutputCompleteEvent Event

Syntax	void OutputCompleteEvent (LONG OutputID); The <i>OutputID</i> parameter indicates the ID number of the completed asynchronous output request.
Remarks	This event is reported when the previously started asynchronous output request is completed successfully.

StatusUpdateEvent Event

Syntax **void StatusUpdateEvent (LONG Status);**

Status is set to one of the following values.

Value	Meaning
PTR_SUE_COVER_OPEN(11)	The printer cover is open. Or the lever position is not correct.
PTR_SUE_COVER_OK(12)	The printer cover is closed and the lever position is correct.
PTR_SUE_JRN_EMPTY(21)	The journal is out of paper.
PTR_SUE_JRN_NEAREMPTY(22)	The journal paper is low.
PTR_SUE_JRN_PAPEROK(23)	The journal paper is prepared.
PTR_SUE_REC_EMPTY(24)	The receipt is out of paper.
PTR_SUE_REC_NEAREMPTY(25)	The receipt paper is low.
PTR_SUE_REC_PAPEROK(26)	The receipt paper is prepared.
PTR_SUE_IDLE(1001)	All the asynchronous outputs have finished either successfully or by cleared. The printer's State is now OPOS_S_IDLE(2). The FlagWhenIdle property must be TRUE for this event to be notified. And, the POS Printer control automatically resets the property to FALSE before the event is notified.
OPOS_SUE_POWER_ONLINE(2001)	The device is powered on and ready. (This can only be notified when PowerNotify = OPOS_PN_ENABLED(1)).
OPOS_SUE_POWER_OFF_OFFLINE(2004)	The device is powered off or offline (This can only be notified when PowerNotify = OPOS_PN_ENABLED(1)).

* The IFD OPOS Control handles a platen position sensor as a cover open sensor.

Remarks This event is reported when the printer device has an important state change.

Chapter5: OPOS INTERFACE SPECIFICATION (Cash Drawer)

5.1. Summary

Common Properties

Property Name	Type	Access	Availability Condition	Initial value after Open
BinaryConversion	Long	R/W	Open	OPOS_BC_NONE (0)
CapCompareFirmwareVersion	Boolean	R	Open	FALSE
CapPowerReporting	Long	R	Open	OPOS_PR_STANDARD (1)
CapStatisticsReporting	Boolean	R	Open	FALSE
CapUpdateFirmware	Boolean	R	Open	FALSE
CapUpdateStatistics	Boolean	R	Open	FALSE
CheckHealthText	String	R	Open	""
Claimed	Boolean	R	Open	FALSE
DeviceEnabled	Boolean	R/W	Open	FALSE
FreezeEvents	Boolean	R/W	Open	FALSE
OpenResult	Long	R	--	OPOS_SUCCESS(0)
PowerNotify	Long	R/W	Open	OPOS_PN_DISABLED (0)
PowerState	Long	R	Open	OPOS_PS_UNKNOWN (2000)
ResultCode	Long	R	--	OPOS_SUCCESS(0)
ResultCodeExtended	Long	R	Open	0
State	Long	R	--	OPOS_S_IDLE (2)
ControlObjectDescription	String	R	--	"SII Cash Drawer Control Object, Copyright (C) 2009 Seiko Instruments Inc."
ControlObjectVersion	Long	R	--	1009004
ServiceObjectDescription	String	R	Open	"SII IFD00x (2inch) Cash Drawer Service Object, Copyright (C) 2010 Seiko Instruments Inc." *1
ServiceObjectVersion	Long	R	Open	1009006
DeviceDescription	String	R	Open	"SII IFD00x (2inch) Cash Drawer" *1
DeviceName	String	R	Open	"IFD00x (2inch) Cash Drawer" *1

*1: Variable item depends on the printer driver to be used.

Specific Properties

Property Name	Type	Access	Availability Condition	Initial Value after Open
CapStatus	Boolean	R	Open	TRUE
CapStatusMultiDrawerDetect	Boolean	R	Open	FALSE
DrawerOpened	Boolean	R	Open	FALSE

Common Methods

Method Name	Availability Condition
Open	--
Close	Open
ClaimDevice	Open
ReleaseDevice	Open & Claim
CheckHealth	Open & Enable
CompareFirmwareVersion	Open, Claim, & Enable
DirectIO	Open & Enable ^{*1}
ResetStatistics	Open & Enable
RetrieveStatistics	Open & Enable
UpdateFirmware	Open, Claim, & Enable
UpdateStatistics	Open & Enable

*1: Item for which the necessary condition differs from that in "UnifiedPOS Retail Peripheral Architecture, Ver. 1.9."

Specific Methods

Method Name	Availability Condition
OpenDrawer	Open & Enable
WaitForDrawerClose	Open & Enable

Events

Event Name	Occurrence Condition
DirectIOEvent	Open & Enable ^{*1}
StatusUpdateEvent	Open & Enable

*1: Item for which the necessary condition differs from that in "UnifiedPOS Retail Peripheral Architecture, Ver. 1.9."

5.2. Common Properties

The properties commonly provided for the Cash Drawer are described.

There are two types of properties, read-only properties and readable/writable properties. The readable/writable properties are indicated by R/W after their name. The return value is described only when it has a special meaning. For errors when the property is accessed without satisfying the initialization condition, refer to the **ResultCode** property.

BinaryConversion Properties R/W

Syntax	LONG BinaryConversion;								
Remarks	The value of BinaryConversion can be modified to one of the following values, but this affects no method. <table><tr><th>Value</th><th>Meaning</th></tr><tr><td>OPOS_BC_NONE(0)</td><td><i>Data</i> is not converted and one <i>Bstring</i> character is placed as one byte. (Default)</td></tr><tr><td>OPOS_BC_NIBBLE(1)</td><td>Each byte is converted into two characters.</td></tr><tr><td>OPOS_BC_DECIMAL(2)</td><td>Each byte is converted into three characters.</td></tr></table>	Value	Meaning	OPOS_BC_NONE(0)	<i>Data</i> is not converted and one <i>Bstring</i> character is placed as one byte. (Default)	OPOS_BC_NIBBLE(1)	Each byte is converted into two characters.	OPOS_BC_DECIMAL(2)	Each byte is converted into three characters.
Value	Meaning								
OPOS_BC_NONE(0)	<i>Data</i> is not converted and one <i>Bstring</i> character is placed as one byte. (Default)								
OPOS_BC_NIBBLE(1)	Each byte is converted into two characters.								
OPOS_BC_DECIMAL(2)	Each byte is converted into three characters.								
Return	This property is initialized to OPOS_BC_NONE(0) by the Open method. When this property is set, one of the following values is placed in the ResultCode property: <table><tr><th>Value</th><th>Meaning</th></tr><tr><td>OPOS_SUCCESS(0)</td><td>The property was set successfully.</td></tr><tr><td>OPOS_E_ILLEGAL(106)</td><td>Improper value is specified.</td></tr></table>	Value	Meaning	OPOS_SUCCESS(0)	The property was set successfully.	OPOS_E_ILLEGAL(106)	Improper value is specified.		
Value	Meaning								
OPOS_SUCCESS(0)	The property was set successfully.								
OPOS_E_ILLEGAL(106)	Improper value is specified.								

CapCompareFirmwareVersion Property

Syntax	BOOL CapCompareFirmwareVersion;
Remarks	FALSE: The function that compares firmware versions is not supported. This property is initialized to FALSE by the Open method.

CapPowerReporting Property

Syntax	LONG CapPowerReporting;				
Remarks	Identifies the power notification capabilities of the Device. The values indicating power notification capabilities are as follows. <table><tr><th>Value</th><th>Meaning</th></tr><tr><td>OPOS_PR_STANDARD(1)</td><td>Two types of power states can be determined and notified.</td></tr></table> This property is initialized to OPOS_PR_STANDARD(1) by the Open method.	Value	Meaning	OPOS_PR_STANDARD(1)	Two types of power states can be determined and notified.
Value	Meaning				
OPOS_PR_STANDARD(1)	Two types of power states can be determined and notified.				

CapStatisticsReporting Property

Syntax	BOOL CapStatisticsReporting;
Remarks	FALSE: No statistical data regarding the device is available. This property is initialized to FALSE by the Open method.

CapUpdateFirmware Property

Syntax	BOOL CapCompareFirmwareVersion;
Remarks	FALSE: Firmware update is not supported. This property is initialized to FALSE by the Open method.

CapUpdateStatistics Property

Syntax	BOOL CapUpdateStatistic;
Remarks	FALSE: None of the statistical data can be reset/updated by the application. This property is initialized to FALSE by the Open method.

CheckHealthText Property

Syntax	BSTR CheckHealthText;
Remarks	Holds the results of the most recent call to the CheckHealth method. The results of diagnosis are as follows. "Internal HCheck: Successful" Internal check is successful. "External HCheck : Successful" External check is successful. "External HCheck : Failure" External check is failed. "Interactive HCheck : Not Supported" Interactive check is not supported. This value is initialized to an "(empty string)" before the first call to CheckHealth method.

Claimed Property

Syntax	BOOL Claimed;
Remarks	TRUE: The exclusive access to the device is obtained. FALSE: The device is released for sharing with other applications. The value of Claimed property is initialized to FALSE by the Open method.

ControlObjectDescription Property

Syntax	BSTR ControlObjectDescription;
Remarks	"SII Cash Drawer Control Object, Copyright (C) 2009 Seiko Instruments Inc." is set. The property identifies the Control Object. This property is always readable.

ControlObjectVersion Property

Syntax	LONG ControlObjectVersion;
Remarks	Indicates the Control Object Version number. This property is always readable.

DeviceDescription Property

Syntax	BSTR DeviceDescription;
Remarks	This property provides devices and related information. The value to be set differs depending on the printer driver used. Example: "SII IFD00x (2inch) Cash Drawer" etc. This property is initialized by the Open method.

DeviceEnabled Property R/W

Syntax	BOOL DeviceEnabled;	
Remarks	TRUE:	The device is in an operational state. If changed to TRUE, then the device is brought to an operational state.
	FALSE:	The device is disabled. If changed to FALSE, then the device is physically disabled.
	The application must set this property to TRUE before using output devices. This property is initialized to FALSE by the Open method.	
Return	When this property is set, one of the following values is placed in the ResultCode property:	
	Value	Meaning
	OPOS_SUCCESS(0)	The property was set successfully.
	OPOS_E_NOHARDWARE(107)	The printer is powered off or the cable is not connected.
	OPOS_E_FAILURE(111)	The connection port cannot be opened. Try again after checking that the port is not used by other programs or that the port exists.

DeviceName Property

Syntax	BSTR DeviceName;
Remarks	This property provides devices and related information. The value to be set differs depending on the printer driver used. Example: "IFD00x (2inch) Cash Drawer" etc. This property is initialized by the Open method.

FreezeEvents Property R/W

Syntax	BOOL FreezeEvents;
Remarks	<p>TRUE: Events are not notified by the Control. Events will be held by the Control until events are unfrozen.</p> <p>FALSE: Events are notified by the Control. If some events were held while events were frozen and all other conditions are correct for delivering the events, then changing FreezeEvents to FALSE will cause these events to be delivered.</p> <p>An application may choose to freeze events for a specific sequence of code where interruption by an event is not desirable. This property is initialized to FALSE by the Open method.</p>
Return	When this property is set, the following value is placed in the ResultCode property:
Value	Meaning
OPOS_SUCCESS(0)	The property was set successfully.

OpenResult Property

Syntax **LONG OpenResult;**

Remarks Holds additional details about the most recent **Open** method. The values of **OpenResult** are as follows.

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

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

PowerNotify Property R/W

Syntax	LONG PowerNotify;						
Remarks	<p>Contains the type of power notification selection made by the Application. The values indicating power notification capabilities are as follows.</p> <table><tr><th>Value</th><th>Meaning</th></tr><tr><td>OPOS_PN_DISABLED(0)</td><td>The Control will not provide any power notifications to the application. No power notification StatusUpdateEvents will be fired, and the PowerState property may not be set.</td></tr><tr><td>OPOS_PN_ENABLED(1)</td><td>When DeviceEnabled is set to TRUE, the Control will fire power notification StatusUpdateEvents and update the PowerState property. The level of functionality depends upon the value of CapPowerReporting.</td></tr></table> <p>The PowerNotify property may only be set while the device is disabled; that is, while the DeviceEnabled property is FALSE.</p> <p>This property is initialized to OPOS_PN_DISABLED(0) by the Open method. This setting provides compatibility with earlier releases of OPOS.</p>	Value	Meaning	OPOS_PN_DISABLED(0)	The Control will not provide any power notifications to the application. No power notification StatusUpdateEvents will be fired, and the PowerState property may not be set.	OPOS_PN_ENABLED(1)	When DeviceEnabled is set to TRUE, the Control will fire power notification StatusUpdateEvents and update the PowerState property. The level of functionality depends upon the value of CapPowerReporting .
Value	Meaning						
OPOS_PN_DISABLED(0)	The Control will not provide any power notifications to the application. No power notification StatusUpdateEvents will be fired, and the PowerState property may not be set.						
OPOS_PN_ENABLED(1)	When DeviceEnabled is set to TRUE, the Control will fire power notification StatusUpdateEvents and update the PowerState property. The level of functionality depends upon the value of CapPowerReporting .						
Return	<p>When this property is set, one of the following values is placed in the ResultCode property:</p> <table><tr><th>Value</th><th>Meaning</th></tr><tr><td>OPOS_SUCCESS(0)</td><td>This property was set successfully.</td></tr><tr><td>OPOS_E_ILLEGAL(106)</td><td>The device is already enabled.</td></tr></table>	Value	Meaning	OPOS_SUCCESS(0)	This property was set successfully.	OPOS_E_ILLEGAL(106)	The device is already enabled.
Value	Meaning						
OPOS_SUCCESS(0)	This property was set successfully.						
OPOS_E_ILLEGAL(106)	The device is already enabled.						

PowerState Property

Syntax **LONG PowerState;**

Remarks The device's current power state is set when it can be determined.

The **PowerState** values are:

Value	Meaning
OPOS_PS_UNKNOWN(2000)	Cannot determine the device's power state due to one of the following reasons. - PowerNotify = OPOS_PN_DISABLED(0) - DeviceEnabled = FALSE
OPOS_PS_ONLINE(2001)	The device is powered on and ready for use.
OPOS_PS_OFF_OFFLINE(2004)	The device is powered off or offline.

This property is initialized to OPOS_PS_UNKNOWN(2000) by the **Open** method.

ResultCode Property

Syntax **LONG ResultCode;**

Remarks This property is set by each method. It is also set when a writable property is set. This property is always readable. Before the **Open** method is called, it returns the value OPOS_E_CLOSED(101).

The **ResultCode** values are:

Value	Meaning
OPOS_SUCCESS(0)	Successful operation
OPOS_E_CLOSED(101)	Attempt was made to access a closed device. This error is not mentioned in the description of Property and Method.
OPOS_E_CLAIMED(102)	Attempt was made to access a device that is exclusively accessed by another process.
OPOS_E_NOSERVICE(104)	The Control cannot communicate with the Service Object. Most likely, a setup or configuration error must be corrected.
OPOS_E_DISABLED(105)	Cannot perform operation while device is disabled.
OPOS_E_ILLEGAL(106)	Attempt was made to perform an illegal or unsupported operation with the device, or an invalid parameter value was used.
OPOS_E_NOHARDWARE(107)	The printer is powered off or the cable is not connected.
OPOS_E_FAILURE(111)	A communication error has occurred.
OPOS_E_TIMEOUT(112)	Timeout has occurred when the ClaimDevice method is executed. The exclusive access cannot be obtained within the preset time.

ResultCodeExtended Property

Syntax	LONG ResultCodeExtended;
Remarks	This property is initialized to 0 by the Open method. The extended error code is not supported.
Reference	ResultCode Property

ServiceObjectDescription Property

Syntax	BSTR ServiceObjectDescription;
Remarks	A character string that identifies the Service Objects is set to this property. The string to be set differs depending on the printer driver used. Example: "SII IFD00x (2inch) Cash Drawer Service Object, Copyright (C) 2010 Seiko Instruments Inc." etc. This property is initialized by the Open method.

ServiceObjectVersion Property

Syntax	LONG ServiceObjectVersion;
Remarks	Indicates the Service Object Version number. This property is initialized by the Open method.

State Property

Syntax	LONG State;						
Remarks	Indicates the current status of the control. <table><tr><th>Value</th><th>Meaning</th></tr><tr><td>OPOS_S_CLOSED(1)</td><td>The Control is closed. (Default)</td></tr><tr><td>OPOS_S_IDLE(2)</td><td>The Control is in a good state and is not busy.</td></tr></table> This property is always readable.	Value	Meaning	OPOS_S_CLOSED(1)	The Control is closed. (Default)	OPOS_S_IDLE(2)	The Control is in a good state and is not busy.
Value	Meaning						
OPOS_S_CLOSED(1)	The Control is closed. (Default)						
OPOS_S_IDLE(2)	The Control is in a good state and is not busy.						

5.3. Common Methods

CheckHealth Method

Syntax	LONG CheckHealth (LONG Level); The <i>Level</i> parameter indicates the type of health check to be performed on the device. The following values may be specified: <table><tr><th>Value</th><th>Meaning</th></tr><tr><td>OPOS_CH_INTERNAL(1)</td><td>Perform a health check without using the device physically. OPOS_SUCCESS is always returned.</td></tr><tr><td>OPOS_CH_EXTERNAL(2)</td><td>Perform a complete test using the device. Open the drawer if possible. OPOS_SUCCESS is returned when it is opened successfully. This method fails when another application has exclusive access to the device.</td></tr><tr><td>OPOS_CH_INTERACTIVE(3)</td><td>Not supported.</td></tr></table>	Value	Meaning	OPOS_CH_INTERNAL(1)	Perform a health check without using the device physically. OPOS_SUCCESS is always returned.	OPOS_CH_EXTERNAL(2)	Perform a complete test using the device. Open the drawer if possible. OPOS_SUCCESS is returned when it is opened successfully. This method fails when another application has exclusive access to the device.	OPOS_CH_INTERACTIVE(3)	Not supported.				
Value	Meaning												
OPOS_CH_INTERNAL(1)	Perform a health check without using the device physically. OPOS_SUCCESS is always returned.												
OPOS_CH_EXTERNAL(2)	Perform a complete test using the device. Open the drawer if possible. OPOS_SUCCESS is returned when it is opened successfully. This method fails when another application has exclusive access to the device.												
OPOS_CH_INTERACTIVE(3)	Not supported.												
Remarks	Call this method to test the state of device. A text description of the results of this method is stored in the CheckHealthText property. The CheckHealth method is always synchronous.												
Return	One of the following values is returned by the method and also placed in the ResultCode property: <table><tr><th>Value</th><th>Meaning</th></tr><tr><td>OPOS_SUCCESS(0)</td><td>Indicates that the health checking procedure was initiated properly and, when possible to determine, indicates that the device is healthy. However, the health of many devices can only be determined by a visual inspection of the test results.</td></tr><tr><td>OPOS_E_CLAIMED(102)</td><td>Another device has exclusive access.</td></tr><tr><td>OPOS_E_ILLEGAL(106)</td><td>Unsupported <i>Level</i> parameter is specified.</td></tr><tr><td>OPOS_E_NOHARDWARE(107)</td><td>The POS Printer to which the Cash Drawer is connected is powered off or the cable is not connected. This value is notified only when OPOS_CH_EXTERNAL(2) is set.</td></tr><tr><td>OPOS_E_FAILURE(111)</td><td>A communication error has occurred. This value is notified only when OPOS_CH_EXTERNAL(2) is set.</td></tr></table>	Value	Meaning	OPOS_SUCCESS(0)	Indicates that the health checking procedure was initiated properly and, when possible to determine, indicates that the device is healthy. However, the health of many devices can only be determined by a visual inspection of the test results.	OPOS_E_CLAIMED(102)	Another device has exclusive access.	OPOS_E_ILLEGAL(106)	Unsupported <i>Level</i> parameter is specified.	OPOS_E_NOHARDWARE(107)	The POS Printer to which the Cash Drawer is connected is powered off or the cable is not connected. This value is notified only when OPOS_CH_EXTERNAL(2) is set.	OPOS_E_FAILURE(111)	A communication error has occurred. This value is notified only when OPOS_CH_EXTERNAL(2) is set.
Value	Meaning												
OPOS_SUCCESS(0)	Indicates that the health checking procedure was initiated properly and, when possible to determine, indicates that the device is healthy. However, the health of many devices can only be determined by a visual inspection of the test results.												
OPOS_E_CLAIMED(102)	Another device has exclusive access.												
OPOS_E_ILLEGAL(106)	Unsupported <i>Level</i> parameter is specified.												
OPOS_E_NOHARDWARE(107)	The POS Printer to which the Cash Drawer is connected is powered off or the cable is not connected. This value is notified only when OPOS_CH_EXTERNAL(2) is set.												
OPOS_E_FAILURE(111)	A communication error has occurred. This value is notified only when OPOS_CH_EXTERNAL(2) is set.												

ClaimDevice Method

Syntax	LONG ClaimDevice (LONG <i>Timeout</i>); The <i>Timeout</i> parameter indicates the maximum waiting time (in millisecond) for exclusive access. If it is zero, the method returns the result immediately even if exclusive access cannot be obtained. If OPOS_FOREVER(-1) is set, the method waits until exclusive access is obtained.								
Remarks	Call this method to request exclusive access to the device. Acquisition of exclusive device is not essential since the Cash Drawer device is a sharable device. When it is successful, the Claimed property is set to TRUE.								
Return	One of the following values is returned by the method and also placed in the ResultCode property: <table><tr><th>Value</th><th>Meaning</th></tr><tr><td>OPOS_SUCCESS(0)</td><td>Exclusive access is granted and available device connection is established. The Claimed property is now set to TRUE. It is also returned if this application has already gained the exclusive access to the device.</td></tr><tr><td>OPOS_E_ILLEGAL(106)</td><td>An invalid <i>Timeout</i> parameter is specified.</td></tr><tr><td>OPOS_E_TIMEOUT(112)</td><td>Another application has exclusive access to the device and the <i>Timeout</i> (in millisecond) has elapsed before the device is released. Or, the device is not available before the <i>Timeout</i> (in millisecond) has elapsed.</td></tr></table>	Value	Meaning	OPOS_SUCCESS(0)	Exclusive access is granted and available device connection is established. The Claimed property is now set to TRUE. It is also returned if this application has already gained the exclusive access to the device.	OPOS_E_ILLEGAL(106)	An invalid <i>Timeout</i> parameter is specified.	OPOS_E_TIMEOUT(112)	Another application has exclusive access to the device and the <i>Timeout</i> (in millisecond) has elapsed before the device is released. Or, the device is not available before the <i>Timeout</i> (in millisecond) has elapsed.
Value	Meaning								
OPOS_SUCCESS(0)	Exclusive access is granted and available device connection is established. The Claimed property is now set to TRUE. It is also returned if this application has already gained the exclusive access to the device.								
OPOS_E_ILLEGAL(106)	An invalid <i>Timeout</i> parameter is specified.								
OPOS_E_TIMEOUT(112)	Another application has exclusive access to the device and the <i>Timeout</i> (in millisecond) has elapsed before the device is released. Or, the device is not available before the <i>Timeout</i> (in millisecond) has elapsed.								

Close Method

Syntax	LONG Close ();						
Remarks	Call this method to release the device and its resource. If the DeviceEnabled property is TRUE, the device is first disabled. If the Claimed property is TRUE, exclusive access to the device is first released. Do not execute this while the event is in progress (or in the event handler).						
Return	One of the following values is returned by the method and also placed in the ResultCode property: <table><tr><th>Value</th><th>Meaning</th></tr><tr><td>OPOS_SUCCESS(0)</td><td>Device is disabled and closed.</td></tr><tr><td><i>Other values</i></td><td>Refer to the description of the ResultCode property.</td></tr></table>	Value	Meaning	OPOS_SUCCESS(0)	Device is disabled and closed.	<i>Other values</i>	Refer to the description of the ResultCode property.
Value	Meaning						
OPOS_SUCCESS(0)	Device is disabled and closed.						
<i>Other values</i>	Refer to the description of the ResultCode property.						

CompareFirmwareVersion Method

Syntax	LONG CompareFirmwareVersion (BSTR <i>FirmwareFileName</i>, Long <i>result</i>);				
Remarks	This method is not supported.				
Return	The following value is returned by the method and also placed in the ResultCode property: <table><tr><th>Value</th><th>Meaning</th></tr><tr><td>OPOS_E_ILLEGAL(106)</td><td>This method is not supported.</td></tr></table>	Value	Meaning	OPOS_E_ILLEGAL(106)	This method is not supported.
Value	Meaning				
OPOS_E_ILLEGAL(106)	This method is not supported.				

DirectIO method

Syntax	LONG DirectIO (LONG <i>Command</i>, LONG* <i>pData</i>, BSTR* <i>pString</i>);				
Remarks	Call this method to communicate directly with the Service Object. This method is not supported.				
Return	The following value is returned by the method and also placed in the ResultCode property: <table><tr><th>Value</th><th>Meaning</th></tr><tr><td>OPOS_E_ILLEGAL(106)</td><td>This method is not supported.</td></tr></table>	Value	Meaning	OPOS_E_ILLEGAL(106)	This method is not supported.
Value	Meaning				
OPOS_E_ILLEGAL(106)	This method is not supported.				

Open Method

Syntax	LONG Open (BSTR DeviceName); The <i>DeviceName</i> parameter specifies the device name to open. Specify the registered device name (such as "IFD00x") or "DefaultCashDrawer."												
Remarks	Call this method to open the device. When the Open method is successful, the common property and other class-specific properties are initialized.												
Return	One of the following values is returned by the method and also placed in the ResultCode property: <table><tr><th>Value</th><th>Meaning</th></tr><tr><td>OPOS_SUCCESS(0)</td><td>Open successful.</td></tr><tr><td>OPOS_E_NOSERVICE(104)</td><td>Could not establish a connection to the Service Object.</td></tr><tr><td>OPOS_E_ILLEGAL(106)</td><td>The Control is already open.</td></tr><tr><td>OPOS_E_NOEXIST(109)</td><td>The specified <i>DeviceName</i> is not found. This includes the cases when <i>DeviceName</i> is set using an "(empty string)."</td></tr><tr><td>OPOS_E_FAILURE(111)</td><td>Initialization of the OPOS Driver is failed.</td></tr></table> <p>The value of the ResultCode property after calling the Open method may not be the same as the Open method return values for the following two cases.</p> <ol style="list-style-type: none">1. When OPOS Control is closed and the Open method is failed: The ResultCode property continues to be OPOS_E_CLOSED(101).2. When the OPOS Control is already opened: The Open method return value is OPOS_E_ILLEGAL(106), but the ResultCode property holds the value before the Open method.	Value	Meaning	OPOS_SUCCESS(0)	Open successful.	OPOS_E_NOSERVICE(104)	Could not establish a connection to the Service Object.	OPOS_E_ILLEGAL(106)	The Control is already open.	OPOS_E_NOEXIST(109)	The specified <i>DeviceName</i> is not found. This includes the cases when <i>DeviceName</i> is set using an "(empty string)."	OPOS_E_FAILURE(111)	Initialization of the OPOS Driver is failed.
Value	Meaning												
OPOS_SUCCESS(0)	Open successful.												
OPOS_E_NOSERVICE(104)	Could not establish a connection to the Service Object.												
OPOS_E_ILLEGAL(106)	The Control is already open.												
OPOS_E_NOEXIST(109)	The specified <i>DeviceName</i> is not found. This includes the cases when <i>DeviceName</i> is set using an "(empty string)."												
OPOS_E_FAILURE(111)	Initialization of the OPOS Driver is failed.												

ReleaseDevice Method

Syntax	LONG ReleaseDevice ();								
Remarks	Call this method to release exclusive access to the device. If the DeviceEnabled property is TRUE and the device is an exclusive-use device, then the device is first disabled. Do not execute this while the event is in progress (or in the event handler).								
Return	One of the following values is returned by the method and also placed in the ResultCode property: <table><tr><th>Value</th><th>Meaning</th></tr><tr><td>OPOS_SUCCESS(0)</td><td>Exclusive access is released. The Claimed property is now FALSE.</td></tr><tr><td>OPOS_E_ILLEGAL(106)</td><td>The application does not have exclusive access to the device.</td></tr><tr><td><i>Other values</i></td><td>Refer to the description of the ResultCode property.</td></tr></table>	Value	Meaning	OPOS_SUCCESS(0)	Exclusive access is released. The Claimed property is now FALSE.	OPOS_E_ILLEGAL(106)	The application does not have exclusive access to the device.	<i>Other values</i>	Refer to the description of the ResultCode property.
Value	Meaning								
OPOS_SUCCESS(0)	Exclusive access is released. The Claimed property is now FALSE.								
OPOS_E_ILLEGAL(106)	The application does not have exclusive access to the device.								
<i>Other values</i>	Refer to the description of the ResultCode property.								

ResetStatistics Method

Syntax	LONG ResetStatistics (BSTR <i>StatisticsBuffer</i>);				
Remarks	This method is not supported.				
Return	The following value is returned by the method and also placed in the ResultCode property: <table><tr><th>Value</th><th>Meaning</th></tr><tr><td>OPOS_E_ILLEGAL(106)</td><td>This method is not supported.</td></tr></table>	Value	Meaning	OPOS_E_ILLEGAL(106)	This method is not supported.
Value	Meaning				
OPOS_E_ILLEGAL(106)	This method is not supported.				

RetrieveStatistics Method

Syntax	LONG RetrieveStatistics (BSTR <i>*pStatisticsBuffer</i>);				
Remarks	This method is not supported.				
Return	The following value is returned by the method and also placed in the ResultCode property: <table><tr><th>Value</th><th>Meaning</th></tr><tr><td>OPOS_E_ILLEGAL(106)</td><td>This method is not supported.</td></tr></table>	Value	Meaning	OPOS_E_ILLEGAL(106)	This method is not supported.
Value	Meaning				
OPOS_E_ILLEGAL(106)	This method is not supported.				

UpdateFirmware Method

Syntax	LONG UpdateFirmware (BSTR <i>FirmwareFileName</i>);	
Remarks	This method is not supported. The following value is returned by the method and also placed in the ResultCode property:	
	Value	Meaning
	OPOS_E_ILLEGAL(106)	This method is not supported.

UpdateStatistics Method

Syntax	LONG UpdateStatistics (BSTR <i>pStatisticsBuffer</i>);	
Remarks	This method is not supported.	
Return	The following value is returned by the method and also placed in the ResultCode property:	
	Value	Meaning
	OPOS_E_ILLEGAL(106)	This method is not supported.

5.4. Specific Properties

CapStatus Property

Syntax	BOOL CapStatus;
Remarks	TRUE: The open/close state of the drawer can be reported. FALSE: The open/close state of the drawer cannot be notified. This property is initialized to TRUE by the Open method.

CapStatusMultiDrawerDetect Property

Syntax	BOOL CapStatusMultiDrawerDetect;
Remarks	FALSE: Not supported. This property is initialized by the Open method.

DrawerOpened Property

Syntax	BOOL DrawerOpened;
Remarks	TRUE: The drawer is open.* FALSE: The drawer is closed.* When the CapStatus property is FALSE, the device cannot inform the state change of the device and this DrawerOpened property is always set to FALSE. This property is initialized to an appropriate value when the device is enabled. * The open/close state of the drawer can be reversed and then notified using the setting of the registry key (InvertDrawerStatus).

5.5. Specific Methods

OpenDrawer Method

Syntax	LONG OpenDrawer ();												
Remarks	Opens the drawer. This method fails when another application has exclusive access to the device.												
Return	One of the following values is returned by the method and also placed in the ResultCode property: <table><tr><th>Value</th><th>Meaning</th></tr><tr><td>OPOS_SUCCESS(0)</td><td>The drawer was opened successfully.</td></tr><tr><td>OPOS_E_CLAIMED(102)</td><td>Another device has exclusive access.</td></tr><tr><td>OPOS_E_DISABLED (105)</td><td>Not enabled.</td></tr><tr><td>OPOS_E_NOHARDWARE(107)</td><td>The printer is powered off or the cable is not connected.</td></tr><tr><td>OPOS_E_FAILURE(111)</td><td>Communication with the device is failed.</td></tr></table>	Value	Meaning	OPOS_SUCCESS(0)	The drawer was opened successfully.	OPOS_E_CLAIMED(102)	Another device has exclusive access.	OPOS_E_DISABLED (105)	Not enabled.	OPOS_E_NOHARDWARE(107)	The printer is powered off or the cable is not connected.	OPOS_E_FAILURE(111)	Communication with the device is failed.
Value	Meaning												
OPOS_SUCCESS(0)	The drawer was opened successfully.												
OPOS_E_CLAIMED(102)	Another device has exclusive access.												
OPOS_E_DISABLED (105)	Not enabled.												
OPOS_E_NOHARDWARE(107)	The printer is powered off or the cable is not connected.												
OPOS_E_FAILURE(111)	Communication with the device is failed.												

WaitForDrawerClose Method

Syntax	LONG WaitForDrawerClose (LONG BeepTimeout, LONG BeepFrequency, LONG BeepDuration, LONG BeepDelay);				
Remarks	This method is not supported.				
Return	The following value is returned and stored in the ResultCode property: <table><tr><th>Value</th><th>Meaning</th></tr><tr><td>OPOS_E_ILLEGAL(106)</td><td>This method is not supported.</td></tr></table>	Value	Meaning	OPOS_E_ILLEGAL(106)	This method is not supported.
Value	Meaning				
OPOS_E_ILLEGAL(106)	This method is not supported.				

5.6. Events

DirectIOEvent Event

Syntax	void DirectIOEvent (LONG <i>EventNumber</i> , LONG* <i>pData</i> , BSTR* <i>pString</i>);
---------------	-------------------------------------------------------------------------------------------------------------------------

Remarks	This event is not notified.
----------------	-----------------------------

StatusUpdateEvent Event

Syntax	void StatusUpdateEvent (LONG <i>Status</i>);
---------------	--------------------------------------------------------------

Status is set to one of the following values.

Value	Meaning
--------------	----------------

CASH_SUE_DRAWERCLOSED(0)	
--------------------------	--

The drawer is closed*.

CASH_SUE_DRAWEROPEN(1)	
------------------------	--

The drawer is open*.

OPOS_SUE_POWER_ONLINE(2001)	
-----------------------------	--

The device is powered on and ready (This can only be notified when **PowerNotify** = OPOS_PN_ENABLED(1)).

OPOS_SUE_POWER_OFF_OFFLINE(2004)	
----------------------------------	--

The device is powered off or offline (This can only be notified when **PowerNotify** = OPOS_PN_ENABLED(1)).

Remarks	This event is notified when the printer device has an important state change.
----------------	-------------------------------------------------------------------------------

* The open/close state of the drawer is reversed and then reported using the setting of the registry key (InvertDrawerStatus).

* To notify the open/close state of the drawer, the following conditions must be satisfied.

- The **CapStatus** property is set to TRUE.
- The **DeviceEnabled** property is set to TRUE.

* To notify the power state, the following conditions must be satisfied.

- The **PowerNotify** property is set to OPOS_PN_ENABLED(1).
- The **DeviceEnabled** property is set to TRUE.

Chapter6: Registry for OCX

The registry used by this OCX is described below.

Although the registry can be set manually, set it using the configuration program. The contents of registry are read at the time when OCX performs the **Open** method; therefore, the changes made during OCX operation do not affect the operation. In order to update the new setting, call the **Open** method after calling the **Close** method to OCX once.

6.1. POS Printer

```
HKEY_LOCAL_MACHINE\SOFTWARE\OLEforRetail\ServiceOPOS
\POSPrinter
  "DefaultPOSPrinter"="IFD00x"
  \IFD00x ="OPOS.POSPRINTER.SO.SII.IFD.1"
    "Service"="C:\OPOS\SII\POSPrinterSO_IFD.dll"
    "Version"="1.9.9"
    "Description"=" SII POS Printer Service Object, Copyright (C) 2010 Seiko Instruments Inc."
    "LogFileName"=""
    "LogLevel"="-1"
    "LogFileSize"="0"
    "AutoCutter"="1"
    "CurrentStation"="2"
    "DataRegistrationTimeout"="30000"
    "DefaultCharacterSet"="999"
    "DriverName"="SII IFD00x (2inch)"
    "MCAutoSave"="1"
    "NearEnd"="1"
    "PeripheralDevice"="2"
    "ReceiveTimeout"="10000"
    "SendTimeout"="10000"
    "ProcessCompletionTiming"="1"
  \IFD50x ="OPOS.POSPRINTER.SO.SII.IFD.1"
    "Service"="C:\OPOS\SII\POSPrinterSO_IFD.dll"
    "Version"="1.9.9"
    "Description"=" SII POS Printer Service Object, Copyright (C) 2010 Seiko Instruments Inc."
    "LogFileName"=""
    "LogLevel"="-1"
    "LogFileSize"="0"
    "AutoCutter"="1"
    "CurrentStation"="2"
    "DataRegistrationTimeout"="30000"
    "DefaultCharacterSet"="999"
    "DriverName"="SII IFD50x (2inch)"
    "MCAutoSave"="1"
```



```

    "NearEnd"="1"
    "PeripheralDevice"="1"
    "ReceiveTimeout"="10000"
    "SendTimeout"="10000"
    "ProcessCompletionTiming"="1"
\PTD00 ="OPOS.POSPRINTER.SO.SII.IFD.1"
    "Service"="C:\OPOS\SII\POSPrinterSO_IFD.dll"
    "Version"="1.9.9"
    "Description"=" SII POS Printer Service Object, Copyright (C) 2010 Seiko Instruments Inc."
    "LogFileName"=""
    "LogLevel"="-1"
    "LogFileSize"="0"
    "AutoCutter"="1"
    "CurrentStation"="2"
    "DataRegistrationTimeout"="30000"
    "DefaultCharacterSet"="999"
    "DriverName"="SII PTD00 (2inch)"
    "MCAutoSave"="1"
    "NearEnd"="1"
    "PeripheralDevice"="2"
    "ReceiveTimeout"="10000"
    "SendTimeout"="10000"
    "ProcessCompletionTiming"="1"
\PTD50 ="OPOS.POSPRINTER.SO.SII.IFD.1"
    "Service"="C:\OPOS\SII\POSPrinterSO_IFD.dll"
    "Version"="1.9.9"
    "Description"=" SII POS Printer Service Object, Copyright (C) 2010 Seiko Instruments Inc."
    "LogFileName"=""
    "LogLevel"="-1"
    "LogFileSize"="0"
    "AutoCutter"="1"
    "CurrentStation"="2"
    "DataRegistrationTimeout"="30000"
    "DefaultCharacterSet"="999"
    "DriverName"="SII PTD50 (2inch)"
    "MCAutoSave"="1"
    "NearEnd"="1"
    "PeripheralDevice"="1"
    "ReceiveTimeout"="10000"
    "SendTimeout"="10000"
    "ProcessCompletionTiming"="1"

```

Value	Meaning
DefaultPOSPrinter	Logic device name
Service	SO file name
Version	SO version
Description	SO detailed information
LogFileName	Log file name Full path must be specified
LogLevel	Log output level - 1 : No output 0 : Error 1 : Warning 2 : Information 3: Debug 4 : Trace
LogFileSize	Upper limit of output log level (KB)
AutoCutter	Autocutter function 0 : Enable 1 : Disable
CurrentStation	POS Printer to be used. 1 : Journal 2 : Receipt
DataRegistrationTimeout	User area reduction timeout (millisecond)
DefaultCharacterSet	Initial value of CharacterSet property
DriverName	Printer driver name Used for printer driver setting
MCAutoSave	Storing of maintenance counter 0: Disable 1: Enable
NearEnd	Paper-near-end sensor function 0 : Enable 1 : Disable
PeripheralDevice	Peripheral device selection 0 : Reserved 1 : Drawer is enable (when IFD501 is used.) 2 : Drawer is enable (when IFD001 is used.) 3 : Reserved
ReceiveTimeout	Receive timeout (millisecond)
SendTimeout	Transmission timeout (millisecond)
ProcessCompletionTiming	Timing for process completion 0 : Timing at completion of data transmission 1 : Timing at completion of the data printing

6.2. Cash Drawer

```
HKEY_LOCAL_MACHINE\SOFTWARE\OLEforRetail\ServiceOPOS
\CashDrawer
  "DefaultCashDrawer"="IFD00x"
  \IFD00x = OPOS.CASHDRAWER.SO.SII.IFD.1
    "Service"="C:\OPOS\SII\CashDrawerSO_IFD.dll"
    "Version"="1.9.6"
    "Description"=" SII Cash Drawer Service Object, Copyright (C) 2010 Seiko Instruments Inc."
    "DriverName"="SII IFD00x (2inch) "
    "OnTimer"="50"
    "OffTimer"="500"
    "InvertDrawerStatus"="F"
    "LogFileName"=""
    "LogLevel"="-1"
    "LogFileSize"="0"
  \IFD50x = OPOS.CASHDRAWER.SO.SII.IFD.1
    "Service"="C:\OPOS\SII\CashDrawerSO_IFD.dll"
    "Version"="1.9.6"
    "Description"=" SII Cash Drawer Service Object, Copyright (C) 2010 Seiko Instruments Inc."
    "DriverName"="SII IFD50x (2inch) "
    "OnTimer"="50"
    "OffTimer"="500"
    "InvertDrawerStatus"="F"
    "LogFileName"=""
    "LogLevel"="-1"
    "LogFileSize"="0"
  \PTD00 = OPOS.CASHDRAWER.SO.SII.IFD.1
    "Service"="C:\OPOS\SII\CashDrawerSO_IFD.dll"
    "Version"="1.9.6"
    "Description"=" SII Cash Drawer Service Object, Copyright (C) 2010 Seiko Instruments Inc."
    "DriverName"="SII PTD00 (2inch) "
    "OnTimer"="50"
    "OffTimer"="500"
    "InvertDrawerStatus"="F"
    "LogFileName"=""
    "LogLevel"="-1"
    "LogFileSize"="0"
  \PTD50 = OPOS.CASHDRAWER.SO.SII.IFD.1
    "Service"="C:\OPOS\SII\CashDrawerSO_IFD.dll"
    "Version"="1.9.6"
    "Description"=" SII Cash Drawer Service Object, Copyright (C) 2010 Seiko Instruments Inc."
    "DriverName"="SII PTD50 (2inch) "
```

```

"OnTimer"="50"
"OffTimer"="500"
"InvertDrawerStatus"="F"
"LogFileName"=""
"LogLevel"="-1"
"LogFileSize"="0"

```

Value	Meaning
DefaultCashDrawer	Logic device name
Service	SO file name
Version	SO version
Description	SO detailed information
LogFileName	Log file name Full path must be specified
LogLevel	Log output level -1: No output 0: Error 1: Warning 2: Information 3: Debug 4: Trace
LogFileSize	Upper limit of output log level (KB)
DriverName	Used for printer driver setting
OnTimer	0 - 510: Pulse On time for drawer drive (millisecond)
OffTimer	0 - 510: Pulse Off time for drawer drive (millisecond)
InvertDrawerStatus	Synchronization of drawer sensor status and cash drawer status. F: When the drawer sensor status is "High", the cash drawer is open. T: When the drawer sensor status is "Low", the cash drawer is open.

Chapter7: Header File

7.1. POS Printer Header File

The header file used in IFD OPOS Control is described below.
The constants used from the header file are as follows.

- OPOS_EPTR_VPPower
- OPOS_EPTR_AUTOCUTTER
- OPOS_EPTR_PRS_JAM
- OPOS_EPTR_PRS_RETRACT
- OPOS_EPTR_HEAD_TEMP
- OPOS_EPTR_PRS_PAPER
- OPOS_EPTR_NOCGROM
- OPOS_EPTR_IMAGEAREA_FULL
- PTR_DI_SET_RESPONSE_REQUEST
- PTR_DI_GET_REMAINING_MEMORY
- PTR_DI_SET_INTERNATIONAL_CHARACTER
- PTR_DI_GET_STATUS_DATA

```
Header file: SIIIFDPtr.h
/////////////////////////////////////////////////////////////////
//
// SIIIFDPtr.h
//
// POS Printer header file for OPOS Applications.
//
// Modification history
// -----
//
/////////////////////////////////////////////////////////////////
#if !defined(SIIIFDPTR_H)
#define SIIIFDPTR_H
/////////////////////////////////////////////////////////////////
// OPOS "ResultCodeExtended" Property Base Constants
/////////////////////////////////////////////////////////////////
const LONG PTRERREXT = 1000; // POS Printer specific error base
/////////////////////////////////////////////////////////////////
```

```

// "ResultCodeExtended" Property Constants for Printer
/////////////////////////////////////////////////////////////////
const LONG OPOS_EPTR_VPPower          = PTRErRExt+1;
const LONG OPOS_EPTR_AutoCutter        = PTRErRExt+2;
const LONG OPOS_EPTR_PRS_Jam           = PTRErRExt+3;
const LONG OPOS_EPTR_PRS_Retract       = PTRErRExt+4;
const LONG OPOS_EPTR_Head_Temp         = PTRErRExt+5;
const LONG OPOS_EPTR_PRS_Paper         = PTRErRExt+7;
const LONG OPOS_EPTR_NoCGROM           = PTRErRExt+8;
const LONG OPOS_EPTR_ImageArea_Full    = PTRErRExt+9;

/////////////////////////////////////////////////////////////////
// Parameter Constants of "DirectIO" Method
/////////////////////////////////////////////////////////////////
const LONG PTR_DI_Set_Response_Request = 2;
const LONG PTR_DI_Get_Remaining_Memory = 3;
const LONG PTR_DI_Set_International_Character = 201;
const LONG PTR_DI_Get_Status_Data      = 501;
#endif // !defined(SIIIFDPTR_H)

```