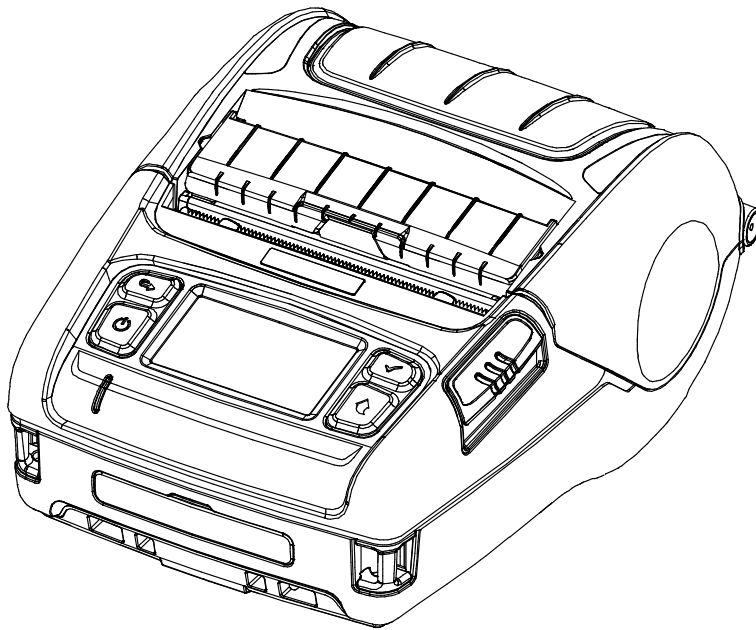




Application Programming Guide

OPOS Driver

Rev. 1.00



Introduction

This is a manual about how to set and configure device and to develop applications to use SATO label printer with SATO OPOS driver.

Before use SATO label printer, set and configure printer by using SATO OPOS driver Setup Utility.

Throughout this manual, “OLE for Retail POS” is called “OPOS”.
Throughout this manual, “Unified for Retail POS” is called “UPOS”.

[Further Source]

OLE for Retail POS committee available from <http://monroecs.com/opos.htm>

Table of Contents

1. Development environment	4
1-1 Communication Configuration	4
2. Properties / Methods	5
2-1 Printer Type.....	5
2-2 Properties Range / Default Value	6
2-2-1 Capability properties setting value	6
2-2-2 Properties default value / range	7
2-2-3 Description of Main Properties	8
2-2-4 Methods	10
2-2-5 Description of Main Methods.....	11
2-2-6 Escape Sequences	15
3. Extended Functions	17
3-1 DirectIO Method	17
3-1-1 SATO OPOS DirectIO Commands Description	17
3-2 DirectIO Command	17
3-2-1 Direct Output.....	17
3-2-2 International charset setting	18
3-3 DirectIOEvent.....	18
4. Error Information.....	19
4-1 ResultCode List.....	19
4-1-1 ClaimDevice method	19
4-1-2 CheckHealth method.....	19
4-1-3 ClearOutput method.....	19
4-1-4 DirectIO method	20
4-1-5 PrintNormal method	20
4-1-6 PrintImmediate method	21
4-1-7 CutPaper method.....	21
4-1-8 RotatePrint method	22
4-1-9 PrintBitmap method.....	22
4-1-10 SetBitmap method.....	23
4-1-11 SetLogo method.....	23
4-1-12 TransactionPrint method	23
5. Resources	24
5-1 OPOS Value (defines).....	24
5-1-1 Barcode type.....	24
5-1-2 Barcode alignment	24
5-1-3 Barcode HRI alignment	24
5-1-4 Image alignment.....	24
5-1-5 TransactionPrint	25
5-2 Code page.....	25
5-2-1 Basic code page.....	25
5-2-2 International charset code table	26
5-2-3 Examples of international charset	27

1. Development environment

1-1 Communication Configuration

- Communication Configuration for serial interface.
- You will set communication configuration in SATO OPOS Setup Utility.

Printer Model	Baud Rate	Printer Type	etc
PV3, PV4	9600/19200/38400/57600/115200	Direct Thermal	203 dpi



- Check the default communication setting.
For more details refer to the user manual included in the printer package.
- A printer which does not support Serial Interface is not indicated.

2. Properties / Methods

2-1 Printer Type

- Property value will be determined by the printer type, as shown in the below table.
(Some property values related to Receipt may be different depending on printer model.)

Direct thermal printer list	PV3, PV4
-----------------------------	----------

2-2 Properties Range / Default Value

2-2-1 Capability properties setting value

Capability Property	Label Printer
CapCompareFirmwareVersion	FALSE
CapPowerReporting	TRUE
CapStatisticsReporting	FALSE
CapUpdateFirmware	FALSE
CapUpdateStatistics	FALSE
CapTransaction	TRUE
CapCoverSensor	TRUE
CapConcurrentRecSlp	FALSE
CapConcurrentJrnSlp	FALSE
CapConcurrentJrnRec	FALSE
CapCharacterSet	TRUE
CapRecUnderline	FALSE
CapRecPageMode	TRUE
CapCuncurrentPageMode	FALSE
CapRecStamp	FALSE
CapRecRotate180	TRUE
CapRecRight90	TRUE
CapRecPapercut	FALSE
CapRecNearEndSensor	FALSE
CapRecMarkFeed	TRUE
CapRecLeft90	TRUE
CapRecltalic	TRUE
CapRecEmptySensor	TRUE
CapRecDwideDhigh	TRUE
CapRecDwide	TRUE
CapRecDhigh	TRUE
CapRecColor	FALSE
CapRecCartridgeSensor	FALSE
CapRecBold	TRUE
CapRecBitmap	TRUE
CapRecBarCode	TRUE
CapRec2Color	FALSE
CapRecPresent	TRUE



Caution

May be capability setting values are different depending on the printer model.

2-2-2 Properties default value / range**1) List Properties**

Property	Label Printer
FontTypefaceList	0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17, 18,19,20,21,22
RecBitmapRotationList	0,L90,R90,180
RecBarcodeRotationList	0,L90,R90,180
CharacterSetList	437,850,852,860,863,865,1252,8651252 (Combined),857,737,1250,1253,1254,855, 862,866,1251,1255,928,864,775,1257,858, 949,932,950,936

2) Paper width properties

Model	Max Width
PV3	576
PV4	832

2-2-3 Description of Main Properties

1) DeviceEnabled

- Description: Indicates whether to use the printer. You should set this value to TRUE to use the printer after calling the Open, ClaimDevice method.
- Related method: void SetDeviceEnabled(BOOL)

2) AsyncMode

- Description: When it is set to TRUE, the print-related method operates in asynchronous mode.
If set to FALSE, the print-related method operates in synchronous mode.
In asynchronous mode, you can check the completion of printing with OutputCompleteEvent.
- Related method : void SetAsyncMode(BOOL)

3) CharacterSetList

- Description: Gets a list of the character sets supported by the printer.
- Related method: CString GetCharacterSetList()

4) CharacterSet

- Description: The character set to use when printing the printer.
You can set any of the values in the CharacterSetList list.
Set after setting DeviceEnabled to TRUE.
- Related method: void SetCharacterSet(long)

• Examples

```
posPrinter.SetCharacterSet(932);           // Shift-JIS – Set to Japanese  
  
posPrinter.SetCharacterSet(949);           // KSC5601 – Set to Korean  
  
posPrinter.SetCharacterSet(936);           // GB2312 – Set of Simplified Chinese  
  
posPrinter.SetCharacterSet(950);           // BIG5 – Set to Traditional Chinese
```



Note

Refer to '5-2-1 Basic Code Page' for information on supported code pages.

5) RecLineCharsList

- Description: Gets a list of the number of characters that can be printed on one line.
- Related method : CString GetRecLineCharsList()

6) RecLineChars

- Description: The number of characters printed on one line

If the value is less than or equal to the preceding value in the list of

RecLineCharsList, it is set to A Font(12x24), and if it is greater than the preceding value, it is set to B Font (9x17).

You should set after setting DeviceEnabled to TRUE.

- Related method: void SetRecLineChars(long)

- Examples

```
/: Gets a list of the number of characters that can be printed on one line.  
CString strRecLineCharsList = posPrinter.GetRecLineCharsList();  
  
//If RecLineCharsList value is 48,64  
posPrinter.SetRecLineChars(48);           //Set to A Font  
  
posPrinter.SetRecLineChars(64);           //Set to B Font
```

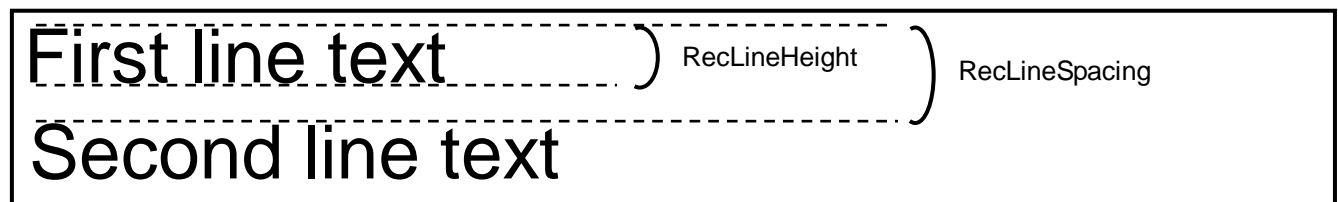
7) RecLineSpacing

- Description: You can set the interval between line and line.

This value can change depending on the MapMode or the selected font.

- Related method : long GetRecLineSpacing(), long SetRecLineSpacing(int)

※ RecLineHeight vs. RecLineSpacing



2-2-4 Methods

Method	Value
beginInsertion	X
beginRemoval	X
changePrintSide	X
clearPrintArea	O
cutPaper	O
drawRuledLine	X
endInsertion	X
endRemoval	X
markFeed	O
pageModePrint	O
printBarCode	O
printBitmap	O
printImmediate	O
printMemoryBitmap	O
printNormal	O
printTwoNormal	X
rotatePrint	O
setBitmap	O
setLogo	O
transactionPrint	O
validateData	O

O: Supported X: Not supported

2-2-5 Description of Main Methods

1) cutPaper

- Description: In the model provided with Auto Cutter, this function cuts the paper.
- Method prototype : void CutPaper(int percentage)
- Example

```
posPrinter.CutPaper(90); //Cut the paper
```

2) printBarcode

- Description: Prints the barcode
- Method prototype : long PrintBarCode(long station, LPCTSTR data, long symbology, long height, long width, long alignment, long textPostition)
- parameter information
 - long station: The type of printer. Set to PTR_S_RECEIPT
 - LPCTSTR data: Sets the data of the barcode to be printed
 - long symbology: Specifies the type of barcode ([See OPOS constant value](#))
 - long height: Specifies the height of the barcode
 - long width: Specifies the width of the barcode
 - long alignment: Specifies the alignment position of the bar code ([See OPOS constant value](#))
 - long textPosition: Specifies the position of the Human Readable Interpretation (HRI). ([See OPOS constant value](#))
- Example

```
//Print a 100 x 100 QR Code in the center of the paper  
posPrinter.PrintBarCode(PTR_S_RECEIPT, "www. satoworldwide.com",  
PTR_BCS_QRCODE, 100, 100, PTR_BC_CENTER, PTR_BC_TEXT_NONE);
```

3) printNormal

- Description: Prints the text. The Escape Sequence string is also available
- Method prototype : long PrintNormal(long station, LPCTSTR data)
- parameter information
 - long station: The type of printer. Set to PTR_S_RECEIPT.
 - LPCTSTR data: Sets the text to be printed
- Example

```
//Print Test Text print  
CString CRLF = "\r\n";  
posPrinter.PrintNormal(PTR_S_RECEIPT, "Print test" + CRLF);
```

4) transactionPrint

- Description: Starts or ends transaction mode.

A series of actions consisting of transactions are performed as if they were a single action.

- Method prototype: long TransactionPrint(long station, long control)
- parameter information
 - long station: The type of printer. Set to PTR_S_RECEIPT
 - long control: Sets the start and end of transaction mode ([See OPOS constant value](#))
- Examples

```
//Transaction mode start
posPrinter.TransactionPrint(PTR_S_RECEIPT, PTR_TP_TRANSACTION);

// Stack text print, image print, paper cutting actions in transaction buffer
posPrinter.PrintNormal(PTR_S_RECEIPT, "Transaction test" + CRLF);
posPrinter.PrintBitmap(PTR_S_RECEIPT, (LPCTSTR)("Logo.bmp"),
    PTR_BM_ASIS, PTR_BM_CENTER);
posPrinter.CutPaper(90);

//Transaction mode end. This time, all actions are performed
posPrinter.TransactionPrint(PTR_S_RECEIPT, PTR_TP_NORMAL);
```

5) printBitmap

- Description: Prints the image
- Method prototype : long PrintBitmap(long station, LPCTSTR fileName, long width, long alignment)
- parameter information
 - long station: The type of printer. Set to PTR_S_RECEIPT
 - LPCTSTR fileName: Specifies the path to the image file to be printed
 - long width: Specifies the width of the image to be printed
 - long alignment: Specifies the alignment position of the image ([See OPOS constant value](#))
- Example

```
// Print the image of Logo.bmp file in the center of the paper
//PTR_BM_ASIS: Print the image at the given size
posPrinter.PrintBitmap(PTR_S_RECEIPT, (LPCTSTR)("Logo.bmp"),
    PTR_BM_ASIS, PTR_BM_CENTER);
```

6) printMemoryBitmap

- Description: Prints images stored in memory
- Method prototype : long PrintMemoryBitmap(long station, LPCTSTR data, long type, Long width, long alignment)
- parameter information
 - long station: The type of printer. Set to PTR_S_RECEIPT
 - LPCTSTR data: Specifies the byte array where image information is stored. BinaryConversion must be used
 - long type: Specifies the format of the image stored in memory
 - long width: Specifies the width of the image to be printed
 - long alignment: Specifies the alignment position of the image ([See OPOS constant value](#))
- Examples

```
CFile  BitmapFile;
BitmapFile.Open("Logo.bmp", CFile::modeRead);           //Open Logo.bmp file

CString  strBitmapData;
strBitmapData.Empty();                                 //Initialize the buffer

// Read 1 byte at a time and apply BinaryConversion
while (BitmapFile.Read(&byBuffer, sizeof(BYTE)) == sizeof(BYTE))
{
    strBitmapData += TCHAR(0x30 + ((byBuffer >> 4) & 0x0F));
    strBitmapData += TCHAR(0x30 + (byBuffer & 0x0F));
}

BitmapFile.Close();

// Print images stored in memory at half the size of paper in the center of the paper
posPrinter.SetBinaryConversion(OPOS_BC_NIBBLE);
posPrinter.PrintMemoryBitmap(PTR_S_RECEIPT, strBitmapData,
    PTR_BMT_BMP, posPrinter.GetRecLineWidth() / 2, PTR_BM_CENTER);
posPrinter.SetBinaryConversion(OPOS_BC_NONE);
```



Note

Refer to the UPOS 1.13 protocol document for a detailed description of BinaryConversion.

7) setBitmap

- Description: Stores the image information to be printed later
- Method prototype : long SetBitmap(long bitmapNumber, long station, LPCTSTR fileName, long width, long alignment)
- parameter information
 - long bitmapNumber: Assigns a number to the image to be stored
 - long station: The type of printer. Set to PTR_S_RECEIPT
 - LPCTSTR fileName: Specifies the path to the image to be stored
 - long width: Specifies the width of the image
 - long alignment : Specifies the alignment position of the image ([See OPOS constant value](#))
- Examples

```
CString CRLF = "\r\n";
CString ESC = "\x1b";

//Store Logo.bmpfile as no. 1
posPrinter.SetBitmap(1, PTR_S_RECEIPT, (LPCTSTR)("Logo.bmp"),
    PTR_BM_ASIS, PTR_BM_CENTER);

// Print the image stored as no.1 using Escape Sequence
posPrinter.PrintNormal(PTR_S_RECEIPT, ESC + "|1B" + CRLF);
```

2-2-6 Escape Sequences

Escape Sequence	Setting value	Description
[#]P	O (value range: 0~100)	Cut the paper
[#]fP	O (value range: 0~100)	Feed and cut the paper
[#]sP	X	This function is not supported
sL	X	This function is not supported
[#]B	O (value range: 1~20)	Print images saved with the setBitmap method
tL	O	Print the saved logo with the setLogo method
bL	O	Print the saved logo with the setLogo method
[#]IF	O (value range: 0~9999)	Feed the paper by the set value. The # value is in line units
[#]uF Base Pitch [inch]	O	Feed the paper by the set value. The # value is in dot units.
[#]rF Maximum [inch]	X	This function is not supported
#E	O	Transmit the data as long as the length of the # number.
#R	X	This function is not supported
#dL	X	This function is not supported
#fT	O	Select the font
[!]bC	O	Print in bold font (! for cancellation)
[!][#]uC	O	Print with underline (! for cancellation)
[!]iC	X	This function is not supported
[#]rC	O	Print using custom colors
[!]rvC	O	Print in reverse style (! for cancellation)
[#]sC	X	This function is not supported
1C	O	Print in normal size
2C	O	Print in double width
3C	O	Print in double height
4C	O	Print in double width and height
#hC	O	Prints in the width of the font at # times magnification
#vC	O	Prints in the height of the font at # times magnification
[#]fC	X	This function is not supported
[!]tbC	X	This function is not supported
[!]tpC	X	This function is not supported
cA	O	Align in the center
rA	O	Align to the right
lA	O	Align to the left
[!][#]stC	X	This function is not supported
N	O	Initialize the status of the printer

O: Supported X: Not supported



To know more about Escape Sequence, please refer to UPOS 1.13.

3. Extended Functions

This section is to explain DirectIO method.
This method will operate properly only with SATO label printer.

3-1 DirectIO Method

Argument	Explanation	Type
Command	Pre-define constant	Long
pData	Number of output data / Value defined by command	Long
pString	Output Data	String

3-1-1 SATO OPOS DirectIO Commands Description

Command	Constant	Description
PTR_DI_OUTPUT	0	Output the pString data
PTR_DI_INTERNATIONAL_CHAR	1	Define the international character set

* The constant of Command can be set at inf file of Printer model in SATO OPOS Driver installation folder.

3-2 DirectIO Command

3-2-1 Direct Output

Argument	Command	PTR_DI_OUTPUT
	pData	Null
	pString	Output data
Description	Sends data without any process after checking printer status. "pString" is not affected by "BinaryConversion"	
Return	Result Code	ResultCodeExtended
	OPOS_SUCCESS	0
	OPOS_E_CLOSED	0
	OPOS_E_CLAIMED	0
	OPOS_E_NOTCLAIMED	0
	OPOS_E_DISABLED	0
	OPOS_E_BUSY	0
	OPOS_E_ILLEGAL	0
	OPOS_E_OFFLINE	0
	OPOS_E_FAILURE	0

3-2-2 International charset setting

Argument	Command	PTR_DI_INTERNATIONAL_CHAR
	pData	constant value which pre-defined PRN_DI_CHAR_USA: 0 PRN_DI_CHAR_FRANCE: 1 PRN_DI_CHAR_GERMANY: 2 PRN_DI_CHAR_UK: 3 PRN_DI_CHAR_DENMARK1: 4 PRN_DI_CHAR_SWEDEN: 5 PRN_DI_CHAR_ITALY: 6 PRN_DI_CHAR_SPAIN: 7 PRN_DI_CHAR_JAPAN: 8 PRN_DI_CHAR_NORWAY: 9 PRN_DI_CHAR_DENMARK2: 10
Description	pString	Empty string
	Sets International charset. Certain characters change according to the pData argument value. Refer to International charset code table and related method. If CharSet property value is changed, International character set will be reset.	
Return	Result Code	ResultCodeExtended
	OPOS_SUCCESS	0
	OPOS_E_CLOSED	0
	OPOS_E_CLAIMED	0
	OPOS_E_NOTCLAIMED	0
	OPOS_E_DISABLED	0
	OPOS_E_BUSY	0
	OPOS_E_ILLEGAL	0
	OPOS_E_OFFLINE	0
	OPOS_E_FAILURE	0

3-3 DirectIOEvent

- Not used.

4. Error Information

This section is to explain returned error information when use POSPrinter methods. For more details, please refer to the UPOS specifications.

4-1 ResultCode List

4-1-1 ClaimDevice method

Method	Result Code	ResultCodeExtended
ClaimDevice	OPOS_E_ILLEGAL	OPOS_EPTR_UNRECOVERABLE
		OPOS_EPTR_MECHANICAL
		OPOS_EPTR_CUTTER
		OPOS_EPTR_OVERHEAT
		OPOS_EPTR_REC_EMPTY
		OPOS_EPTR_JRN_EMPTY

4-1-2 CheckHealth method

Method	Result Code	ResultCodeExtended
CheckHealth	OPOS_SUCCESS	0
	OPOS_E_CLOSED	0
	OPOS_E_CLAIMED	0
	OPOS_E_DISABLED	0
	OPOS_E_ILLEGAL	0
	OPOS_E_OFFLINE	0
	OPOS_E_FAILURE	OPOS_EPTR_UNRECOVERABLE
		OPOS_EPTR_CUTTER
		OPOS_EPTR_MECHANICAL
		OPOS_EPTR_OVERHEAT

4-1-3 ClearOutput method

Method	Result Code	ResultCodeExtended
ClearOutput	OPOS_SUCCESS	0
	OPOS_E_CLOSED	0
	OPOS_E_CLAIMED	0
	OPOS_E_NOTCLAIMED	0

4-1-4 DirectIO method

Method	Result Code	ResultCodeExtended	
DirectIO	OPOS_SUCCESS	0	
	OPOS_E_CLOSED	0	
	OPOS_E_CLAIMED	0	
	OPOS_E_NOTCLAIMED	0	
	OPOS_E_DISABLED	0	
	OPOS_E_ILLEGAL	0	
	OPOS_E_OFFLINE	0	
	OPOS_E_FAILURE		OPOS_EPTR_UNRECOVERABLE
			OPOS_EPTR_CUTTER
			OPOS_EPTR_MECHANICAL
			OPOS_EPTR_OVERHEAT
	OPOS_E_EXTENDED		OPOS_EPTR_COVER_OPEN
			OPOS_EPTR_JRN_EMPTY
OPOS_EPTR_REC_EMPTY			

4-1-5 PrintNormal method

Method	Result Code	ResultCodeExtended	
PrintNormal	OPOS_SUCCESS	0	
	OPOS_E_CLOSED	0	
	OPOS_E_CLAIMED	0	
	OPOS_E_NOTCLAIMED	0	
	OPOS_E_DISABLED	0	
	OPOS_E_ILLEGAL	0	
	OPOS_E_OFFLINE	0	
	OPOS_E_FAILURE		OPOS_EPTR_UNRECOVERABLE
			OPOS_EPTR_CUTTER
			OPOS_EPTR_MECHANICAL
			OPOS_EPTR_OVERHEAT
	OPOS_E_EXTENDED		OPOS_EPTR_COVER_OPEN
			OPOS_EPTR_JRN_EMPTY
OPOS_EPTR_REC_EMPTY			

4-1-6 PrintImmediate method

Method	Result Code	ResultCodeExtended	
PrintImmediate	OPOS_SUCCESS	0	
	OPOS_E_CLOSED	0	
	OPOS_E_CLAIMED	0	
	OPOS_E_NOTCLAIMED	0	
	OPOS_E_DISABLED	0	
	OPOS_E_ILLEGAL	0	
	OPOS_E_OFFLINE	0	
	OPOS_E_FAILURE		OPOS_EPTR_UNRECOVERABLE
			OPOS_EPTR_CUTTER
			OPOS_EPTR_MECHANICAL
			OPOS_EPTR_OVERHEAT
	OPOS_E_EXTENDED		OPOS_EPTR_COVER_OPEN
			OPOS_EPTR_JRN_EMPTY
OPOS_EPTR_REC_EMPTY			

4-1-7 CutPaper method

Method	Result Code	ResultCodeExtended	
CutPaper	OPOS_SUCCESS	0	
	OPOS_E_CLOSED	0	
	OPOS_E_CLAIMED	0	
	OPOS_E_NOTCLAIMED	0	
	OPOS_E_DISABLED	0	
	OPOS_E_ILLEGAL	0	
	OPOS_E_OFFLINE	0	
	OPOS_E_FAILURE		OPOS_EPTR_UNRECOVERABLE
			OPOS_EPTR_CUTTER
			OPOS_EPTR_MECHANICAL
			OPOS_EPTR_OVERHEAT
	OPOS_E_EXTENDED		OPOS_EPTR_COVER_OPEN
			OPOS_EPTR_JRN_EMPTY
OPOS_EPTR_REC_EMPTY			

4-1-8 RotatePrint method

Method	Result Code	ResultCodeExtended	
RotatePrint	OPOS_SUCCESS	0	
	OPOS_E_CLOSED	0	
	OPOS_E_CLAIMED	0	
	OPOS_E_NOTCLAIMED	0	
	OPOS_E_DISABLED	0	
	OPOS_E_ILLEGAL	0	
	OPOS_E_OFFLINE	0	
	OPOS_E_FAILURE		OPOS_EPTR_UNRECOVERABLE
			OPOS_EPTR_CUTTER
			OPOS_EPTR_MECHANICAL
			OPOS_EPTR_OVERHEAT
	OPOS_E_EXTENDED		OPOS_EPTR_COVER_OPEN
			OPOS_EPTR_JRN_EMPTY
			OPOS_EPTR_REC_EMPTY

4-1-9 PrintBitmap method

Method	Result Code	ResultCodeExtended	
PrintBitmap	OPOS_SUCCESS	0	
	OPOS_E_CLOSED	0	
	OPOS_E_CLAIMED	0	
	OPOS_E_NOTCLAIMED	0	
	OPOS_E_DISABLED	0	
	OPOS_E_ILLEGAL	0	
	OPOS_E_OFFLINE	0	
	OPOS_E_FAILURE		OPOS_EPTR_UNRECOVERABLE
			OPOS_EPTR_CUTTER
			OPOS_EPTR_MECHANICAL
			OPOS_EPTR_OVERHEAT
	OPOS_E_NOEXIST		0
	OPOS_E_EXTENDED		OPOS_EPTR_COVER_OPEN
			OPOS_EPTR_JRN_EMPTY
			OPOS_EPTR_REC_EMPTY
			OPOS_EPTR_SLP_EMPTY
			OPOS_EPTR_TOOBIG
		OPOS_EPTR_BADFORMAT	

4-1-10 SetBitmap method

Method	Result Code	ResultCodeExtended	
SetBitmap	OPOS_SUCCESS	0	
	OPOS_E_CLOSED	0	
	OPOS_E_CLAIMED	0	
	OPOS_E_NOTCLAIMED	0	
	OPOS_E_DISABLED	0	
	OPOS_E_ILLEGAL	0	
	OPOS_E_OFFLINE	0	
	OPOS_E_FAILURE		OPOS_EPTR_UNRECOVERABLE
			OPOS_EPTR_CUTTER
			OPOS_EPTR_MECHANICAL
			OPOS_EPTR_OVERHEAT
	OPOS_E_NOEXIST		0
	OPOS_E_EXTENDED		OPOS_EPTR_COVER_OPEN
			OPOS_EPTR_JRN_EMPTY
OPOS_EPTR_REC_EMPTY			
OPOS_EPTR_SLP_EMPTY			
OPOS_EPTR_TOOBIG			
		OPOS_EPTR_BADFORMAT	

4-1-11 SetLogo method

Method	Result Code	ResultCodeExtended
SetLogo	OPOS_SUCCESS	0
	OPOS_E_CLOSED	0
	OPOS_E_CLAIMED	0
	OPOS_E_NOTCLAIMED	0
	OPOS_E_DISABLED	0
	OPOS_E_ILLEGAL	0

4-1-12 TransactionPrint method

Method	Result Code	ResultCodeExtended	
TransactionPrint	OPOS_SUCCESS	0	
	OPOS_E_CLOSED	0	
	OPOS_E_CLAIMED	0	
	OPOS_E_NOTCLAIMED	0	
	OPOS_E_DISABLED	0	
	OPOS_E_ILLEGAL	0	
	OPOS_E_OFFLINE	0	
	OPOS_E_FAILURE		OPOS_EPTR_UNRECOVERABLE
			OPOS_EPTR_CUTTER
			OPOS_EPTR_MECHANICAL
			OPOS_EPTR_OVERHEAT
	OPOS_E_NOEXIST		0
	OPOS_E_EXTENDED		OPOS_EPTR_COVER_OPEN
			OPOS_EPTR_JRN_EMPTY
OPOS_EPTR_REC_EMPTY			
OPOS_EPTR_SLP_EMPTY			

5. Resources

5-1 OPOS Value (defines)

5-1-1 Barcode type

Code define	Description
PTR_BCS_UPCA:	UPC-A
PTR_BCS_UPCE	UPC-E
PTR_BCS_JAN8	JAN 8 (= EAN 8)
PTR_BCS_JAN13	JAN 13 (= EAN 13)
PTR_BCS_ITF	Interleaved 2 of 5
PTR_BCS_Codabar	Codabar
PTR_BCS_Code39	Code 39
PTR_BCS_Code93	Code 93
PTR_BCS_Code128	Code 128
PTR_BCS_Code128_Parsed	Code 128 with parsing
PTR_BCS_GS1DATABAR	GS1 DataBar Omnidirectional
PTR_BCS_GS1DATABAR_E	GS1 DataBar Expanded
PTR_BCS_GS1DATABAR_S	GS1 DataBar Stacked Omnidirectional
PTR_BCS_GS1DATABAR_E_S	GS1 DataBar Expanded Stacked
PTR_BCS_PDF417	PDF 417
PTR_BCS_MAXICODE	MAXI Code
PTR_BCS_DATAMATRIX	Data Matrix
PTR_BCS_QRCODE	QR Code
PTR_BCS_AZTEC	Aztec

5-1-2 Barcode alignment

Code define	Description
PTR_BC_LEFT	Left alignment
PTR_BC_CENTER	Center alignment
PTR_BC_RIGHT	Right alignment

5-1-3 Barcode HRI alignment

Code define	Description
PTR_BC_TEXT_NONE	Do not print HRI
PTR_BC_TEXT_ABOVE	Print HRI on top of barcode
PTR_BC_TEXT_BELOW	Print HRI at the bottom of barcode

5-1-4 Image alignment

Code define	Description
PTR_BM_LEFT	Left alignment
PTR_BM_CENTER	Center alignment
PTR_BM_RIGHT	Right alignment

5-1-5 TransactionPrint

Code define	Description
PTR_TP_TRANSACTION	Initializes the buffer to empty and starts the transaction mode.
PTR_TP_NORMAL	Exit transaction mode and output the stacked data in Buffer.

5-2 Code page**5-2-1 Basic code page**

Code page number	Description
PC437	U.S.A
PC850	LATIN 1
PC852	LATIN 2
PC860	PORTUGUESE
PC863	CANADIAN FRENCH
PC865	NORDIC
PC1252	WINDOWS LATIN 1
PC865 + PC1252	EUROPEAN COMBINED
PC857	TURKISH
PC737	GREEK
PC1250	WINDOWS LATIN 2
PC1253	GREEK
PC1254	TURKISH
PC855	CYRILLIC
PC862	HEBREW
PC866	CYRILLIC
PC1251	CYRILLIC
PC1255	HEBREW
PC928	GREEK
PC864	Arabic
PC775	Baltic
PC1257	Baltic
PC858	Latin 1 + Euro
PC932	Shift-JIS
PC936	GB2312
PC949	KSC5601
PC950	BIG5

5-2-2 International characterset code table

	Country	ASCII code (hexadecimal number)											
		23	24	40	5B	5C	5D	5E	60	7B	7C	7D	7E
0	U.S.A.	#	\$	@	[\]	^	'	{		}	~
1	France	#	\$	à	°	ç	§	^	'	é	ù	è	¨
2	Germany	#	\$	§	Ä	Ö	Ü	^	'	ä	ö	ü	β
3	U.K.	£	\$	@	[\]	^	'	{		}	~
4	Denmark I	#	\$	@	Æ	Ø	Å	^	'	æ	ø	å	~
5	Sweden	#	¤	É	Ä	Ö	Å	Ü	é	ä	ö	å	ü
6	Italy	#	\$	@	°	\	é	^	ù	à	ò	è	ì
7	Spain	Pt	\$	@	í	Ñ	¿	^	'	¨	ñ	}	~
8	Japan	#	\$	@	[¥]	^	'	{		}	~
9	Norway	#	¤	É	Æ	Ø	Å	Ü	é	æ	ø	å	ü
10	Denmark II	#	\$	É	Æ	Ø	Å	Ü	é	æ	ø	å	ü
11	Spain II	#	\$	á	í	Ñ	¿	é	`	í	ñ	ó	ú
12	Latin America	#	\$	á	í	Ñ	¿	é	ü	í	ñ	ó	ú
13	Korea	#	\$	@	[₩]	^	`	{		}	~

5-2-3 Examples of international character set

```
CString CRLF = _T("\r\n");
CString cStr = _T("");
BSTR pString = cStr.AllocSysString();

//Select French character code
long pData = PRN_DI_CHAR_FRANCE;
posPrinter.DirectIO(PTR_DI_INTERNATIONAL_CHAR, &pData, &pString);

// Print à character
posPrinter.PrintNormal(PTR_S_RECEIPT, _T("\x40") + CRLF);

// Select German character code
pData = PRN_DI_CHAR_GERMANY;
posPrinter.DirectIO(PTR_DI_INTERNATIONAL_CHAR, &pData, &pString);

// Print § character
posPrinter.PrintNormal(PTR_S_RECEIPT, _T("\x40") + CRLF);

// Select Italian character code
pData = PRN_DI_CHAR_ITALY;
posPrinter.DirectIO(PTR_DI_INTERNATIONAL_CHAR, &pData, &pString);

// Print @ character
posPrinter.PrintNormal(PTR_S_RECEIPT, _T("\x40") + CRLF);
```

