

Information of the Manual

This SDK manual provides the dll file information for Windows application development.

We continuously promote and update the function and quality of all our products. Any change to the product specification and the manual will be without any further notice.

1.1 Operation Platform

- Windows 2003/XP/7/8

1.2 Remark

- When error code return value is greater than 0, it is the internal error of Windows system, please refer to related help file.

PrtPrinterCreator

Set up the target printer of specified model (should set up target printer before operating any printer).

Int **PrtPrinterCreator**(

void** *phandle*,

const TCHAR* *model*

);

Parameter:

void handle*

[in,out] The created target printer object.

const TCHAR model*

[in] Specify the model of target printer.

Model Name	Description
TP801	HPRT TP801 3" Thermal Receipt Printer
TP805	HPRT TP805 3" Thermal Receipt Printer
TP806	HPRT TP806 3" Thermal Receipt Printer

Return:

Error Code	Value	Description
HPRT_SUCCESS	0	Normal
HPRT_E_INVALID_MODEL	-8	Invalid model name
HPRT_E_NO_ENOUGH_MEMORY	-2	No enough memory

PrtPrinterDestroy

Release the resource of specified model printer that has set up (after operation completed and no more operation for printer, it should release the printer that has set up).

```
int PrinterDestroy(
```

```
    void* handle
```

```
);
```

Parameter:

void handle*

[in] Target printer that needs to release.

Return Value:

Error Code	Value	Description
HPRT_SUCCESS	0	Normal
HPRT_E_BAD_HANDLE	-6	Invalid handle

PrtPortOpen

Open the communication port to operate printing. Currently it supports USB, internet, serial interface and LPT.

```
int PrtPortOpen(  
  
    void* handle,  
  
    const TCHAR* ioSettings  
  
);
```

Parameter:

void handle*

[in,out] The created target printer object.

const TCHAR ioSettings*

[in] Set up the parameter of communication port that connected to the target printer.

Configuration List:

Type	Configuration	Description	Sample
USB	USB[,Position]	USB: connect any USB printer of our company	USB USB,Port_#0004.Hub_#0003
		USB[,Position]: When connecting to multi printers of our company, can specify connecting to one particular USB printer through USB position information (Position parameter)	
NET	NET, IP Add (IPV4)[,Port]	Specify the IP add and port of internet printer. If not specifying port, the default port is 9100.	NET,192.168.0.36 NET,192.168.0.36,9100
COM	COMn,BAUDRATE_rate	Specify the number and baud rate of connected serial port .	COM5,BAUDRATE_19200
LPT	LPTn	Specify the number of connected parallel port.	LPT1

Note: [] indicates selective parameter

How to check the information of USB printer position (Position parameter):

In Windows device manager, unfold “USB controller” , select “USB print support” device, select “Property” on right click menu, click “Detail Information” .

The value of Property “Position Information” is the position information of USB printer. (Position parameter)

Return Value:

Error Code	Value	Description
HPRT_SUCCESS	0	Normal
HPRT_E_INVALID_PARAMETER	-1	Invalid parameter
HPRT_E_BAD_HANDLE	-6	Invalid handle
HPRT_E_IO_PORT_OPEN_FAILED	-311	Port open failed

PrtPortClose

Operation for closing the port.

```
int PrtPortClose(  
    void* handle  
);
```

Parameter:

void handle*
[in,out] The created target printer object.

Return Value:

Error Code	Value		Description
HPRT_SUCCESS	0		Normal
HPRT_E_BAD_HANDLE	-6		Invalid handle

PrtPrinterInitialize

Initialize the printer(Clear the data of print buffer and reset printer temporary setting items).

```
int PrtPrinterInitialize(  
    void* handle  
);
```

Parameter:

void handle*
[in,out] The created target printer object.

Return Value:

Error Code	Value	Description
HPRT_SUCCESS	0	Normal
HPRT_E_BAD_HANDLE	-6	Invalid handle
HPRT_E_IO_PORT_NOT_OPEN	-309	Communication port not open
HPRT_E_IO_WRITE_FAILED	-321	Write failed
HPRT_E_IO_WRITE_TIMEOUT	-322	Write timeout

PrtSetTextLineSpace

Set print text line spacing.

```
int PrtSetTextLineSpace(  
    Void* handle,  
    int lineSpace  
);
```

Parameter:

void handle*

[in,out] The created target printer object.

int lineSpace

[in] Set text line spacing. $0 \leq \text{linespace} \leq 255$

Return Value:

Error Code	Value	Description
HPRT_SUCCESS	0	Normal
HPRT_E_INVALID_PARAMETER	-1	Invalid parameter
HPRT_E_BAD_HANDLE	-6	Invalid handle
HPRT_E_IO_PORT_NOT_OPEN	-309	Communication port not open
HPRT_E_IO_WRITE_FAILED	-321	Write failed
HPRT_E_IO_WRITE_TIMEOUT	-322	Write timeout

PrtCancelPrintDataInPageMode

In page mode, clear all the print data of current printing area.

```
int PrtCancelPrintDataInPageMode(  
    void* handle  
);
```

Parameter:

void handle*
[in,out] The created target printer object.

Return Value:

Error Code	Value	Description
HPRT_SUCCESS	0	Normal
HPRT_E_INVALID_MODEL_TYPE	-3	The model type does not support this function
HPRT_E_BAD_HANDLE	-6	Invalid handle
HPRT_E_IO_PORT_NOT_OPEN	-309	Communication port not open
HPRT_E_IO_WRITE_FAILED	-321	Write failed
HPRT_E_IO_WRITE_TIMEOUT	-322	Write timeout

PrtCheckPrinterState

Operation for getting the printer real-time state.

```
int PrtCheckPrinterState(  
    void* handle,  
    unsigned int* printerStatus  
);
```

Parameter:

void handle*

[in,out] The created target printer object.

unsigned int printerStatus*

[in,out] Pinter real-time state, when returning to multi-state, value is showed as accumulation, return state please refer to below:

Error Code	Value	Description
HPRT_STS_Normal	0	Normal
HPRT_STS_PAPEREMPTY	1	Paper out
HPRT_STS_COVEROPEN	2	Upper cover opened
HPRT_STS_PAPERNEAREND	4	Paper near end
HPRT_STS_ERROR	32	Error ocured when getting state
HPRT_STS_NOT_OPEN	64	Port not open
HPRT_STS_OFFLINE	128	Printer off line

Return Value:

Error Code	Value	Description
HPRT_SUCCESS	0	Normal
HPRT_E_INVALID_MODEL_TYPE	-3	The model type does not support this function
HPRT_E_BAD_HANDLE	-6	Invalid handle
HPRT_E_IO_PORT_NOT_OPEN	-309	Communication port not open
HPRT_E_IO_WRITE_FAILED	-321	Write failed
HPRT_E_IO_WRITE_TIMEOUT	-322	Write timeout
HPRT_E_IO_READ_FAILED	-331	Read failed
HPRT_E_IO_READ_TIMEOUT	-332	Read timeout

PrtSetCharacterSet

Set character set.

```
int PrtSetCharacterSet(  
    void* handle,  
    int characterSet  
);
```

Parameter:

void handle*

[in,out] The created target printer object.

int characterSet

[in] Select character set setting.

Default: Chinese character set.

Return Value:

Character Set	Value	Description
HPRT_CHARACTERSET_DEFAULT	0	Default
HPRT_CHARACTERSET_USA	437	English character set
HPRT_CHARACTERSET_MULTILINGUAL	850	Multilingual character set
HPRT_CHARACTERSET_LATIN2	852	Latin character set
HPRT_CHARACTERSET_EURO	858	European character set
HPRT_CHARACTERSET_PORTUGUESE	860	Portuguese character set
HPRT_CHARACTERSET_CANADIAN_FRE	863	Canadian-French character set
HPRT_CHARACTERSET_NORDIC	865	Nordic character set
HPRT_CHARACTERSET_CYRILLIC2	866	Cyrillic character set
HPRT_CHARACTERSET_WPC1252	1252	WPC character set

Return Value:

Error Code	Value	Description
HPRT_SUCCESS	0	Normal
HPRT_E_INVALID_PARAMETER	-1	Invalid parameter
HPRT_E_INVALID_MODEL_TYPE	-3	The model type does not support this function
HPRT_E_BAD_HANDLE	-6	Invalid handle
HPRT_E_IO_PORT_NOT_OPEN	-309	Communication port not open
HPRT_E_IO_WRITE_FAILED	-321	Write failed
HPRT_E_IO_WRITE_TIMEOUT	-322	Write timeout

PrtCutPaper

Operation for printer paper feed and cut paper.

```
int PrtCutPaper(  
    void* handle,  
    int cutMode,  
    int distance  
);
```

Parameter:

void handle*

[in,out] The created target printer object.

int cutMode

[in] Paper cut mode.

Paper Cut Mode	Value	Description
HPRT_FULL_CUT	0	Full cut
HPRT_PARTIAL_CUT	1	Partial cut

int distance

[in] Paper feed distance.

Return Value:

Error Code	Value	Description
HPRT_SUCCESS	0	Normal
HPRT_E_INVALID_PARAMETER	-1	Invalid parameter
HPRT_E_INVALID_MODEL_TYPE	-3	The model type does not support this function
HPRT_E_BAD_HANDLE	-6	Invalid handle
HPRT_E_IO_PORT_NOT_OPEN	-309	Communication port not open
HPRT_E_IO_WRITE_FAILED	-321	Write failed
HPRT_E_IO_WRITE_TIMEOUT	-322	Write timeout

PrtFeedLine

Operation for printer paper feed.

```
int PrtFeedLine(  
    void* handle,  
    int lines  
);
```

Parameter:

void handle*

[in,out] The created target printer object.

int lines

[in] Set lines of paper feed. $0 \leq \text{lines} \leq 255$

Return Value:

Error Code	Value	Description
HPRT_SUCCESS	0	Normal
HPRT_E_INVALID_PARAMETER	-1	Invalid parameter
HPRT_E_BAD_HANDLE	-6	Invalid handle
HPRT_E_IO_PORT_NOT_OPEN	-309	Communication port not open
HPRT_E_IO_WRITE_FAILED	-321	Write failed
HPRT_E_IO_WRITE_TIMEOUT	-322	Write timeout

PrtOpenDrawer

Operation for opening drawer(printer should be connected with drawer).

```
int PrtOpenDrawer(  
    void* handle,  
    int pinMode,  
    int onTime,  
    int offTime  
);
```

Parameter:

void handle*

[in,out] The created target printer object.

int cutMode

[in] Select the pin which drawer connected.

Pin	Value	Description
HPRT_CASH_DRAWER_1	0	Pin 2
HPRT_CASH_DRAWER_2	1	Pin 5

int onTime

[in] Set the start time of pulse, onTime*2ms.

int offTime

[in] Set the end time of pulse, offTime*2ms.

Remark: When the setting value of end time is less than that of start time, end time is equal to the start time.

Return Value:

Error Code	Value	Description
HPRT_SUCCESS	0	Normal
HPRT_E_INVALID_PARAMETER	-1	Invalid parameter
HPRT_E_INVALID_MODEL_TYPE	-3	The model type does not support this function
HPRT_E_BAD_HANDLE	-6	Invalid handle
HPRT_E_IO_PORT_NOT_OPEN	-309	Communication port not open
HPRT_E_IO_WRITE_FAILED	-321	Write failed
HPRT_E_IO_WRITE_TIMEOUT	-322	Write timeout

PrtPrintText

Operation for printing printer text.

```
int PrtPrintText(  
    void* handle,  
    const TCHAR* data,  
    int alignment,  
    int attribute,  
    int textSize  
);
```

void handle*
[in,out] The created target printer object.

const TCHAR data*,
[in] The text data which need to print.

int alignment
[in] The alignment method of text.

Alignment Method	Value	Description
HPRT_ALIGNMENT_LEFT	0	Left alignment
HPRT_ALIGNMENT_CENTER	1	Center alignment
HPRT_ALIGNMENT_RIGHT	2	Right alignment

int attribute
[in] Set text font property, property setting can be accumulated.

Text Property	Value	Description
HPRT_TEXT_NORMAL_MODE	0	Default setting font A
HPRT_TEXT_FONT_BOLD	2	Set font bold
HPRT_TEXT_FONT_UNDERLINE_MODE	4	Set font underline mode
HPRT_TEXT_FONT_REVERSE	8	Set font reverse
HPRT_TEXT_FONT_DW_DH_MODE	48	Set font double-width and double-height

int textSize
[in] Set the text size (It will not be printed if the text size exceeds print paper area).

Set text width:

Text Width	Value	Description
HPRT_TEXT_SIZE_0WIDTH	0	Text width × 1
HPRT_TEXT_SIZE_1WIDTH	16	Text width × 2
HPRT_TEXT_SIZE_2WIDTH	32	Text width × 3
HPRT_TEXT_SIZE_3WIDTH	48	Text width × 4
HPRT_TEXT_SIZE_4WIDTH	64	Text width × 5
HPRT_TEXT_SIZE_5WIDTH	80	Text width × 6
HPRT_TEXT_SIZE_6WIDTH	96	Text width × 7
HPRT_TEXT_SIZE_7WIDTH	112	Text width × 8

Set text height:

Text Height	Value	Description
HPRT_TEXT_SIZE_0HEIGHT	0	Text height × 1
HPRT_TEXT_SIZE_1HEIGHT	1	Text height × 2
HPRT_TEXT_SIZE_2HEIGHT	2	Text height × 3
HPRT_TEXT_SIZE_3HEIGHT	3	Text height × 4
HPRT_TEXT_SIZE_4HEIGHT	4	Text height × 5
HPRT_TEXT_SIZE_5HEIGHT	5	Text height × 6
HPRT_TEXT_SIZE_6HEIGHT	6	Text height × 7
HPRT_TEXT_SIZE_7HEIGHT	7	Text height × 8

Return Value:

Error Code	Value	Description
HPRT_SUCCESS	0	Normal
HPRT_E_INVALID_PARAMETER	-1	Invalid parameter
HPRT_E_BAD_HANDLE	-6	Invalid handle
HPRT_E_IO_PORT_NOT_OPEN	-309	Communication port not open
HPRT_E_IO_WRITE_FAILED	-321	Write failed
HPRT_E_IO_WRITE_TIMEOUT	-322	Write timeout

PrtPrintBarCode

Operation for printing bar code. In standard mode, only when bar code print position is in new line or without data in buffer can print normally. In page mode, when the command of printing bar code is not received, the bar code data will be saved in buffer and print will not be executed.

```
int PrtPrintBarCode(  
    void* handle,  
    int bcType,  
    const TCHAR* bcData,  
    int width,  
    int height,  
    int alignment,  
    int hriPosition  
);
```

Parameter:

void handle*

[in,out] The created target printer object.

int bcType

[in] Set bar code type.

const TCHAR bcData,*

[in] Bar code data.

Bar Code Type	Value	Bar Code Data Length	Effective Data Range
HPRT_BARCODE_UPC_A	65	$11 \leq n \leq 12$	$48 \leq \text{data} \leq 57$
HPRT_BARCODE_UPC_E	66	$11 \leq n \leq 12$	$48 \leq \text{data} \leq 57$
HPRT_BARCODE_EAN13	67	$12 \leq n \leq 13$	$48 \leq \text{data} \leq 57$
HPRT_BARCODE_JAN13			
HPRT_BARCODE_EAN8	68	$7 \leq n \leq 8$	$48 \leq \text{data} \leq 57$
HPRT_BARCODE_JAN8			
HPRT_BARCODE_CODE39	69	$1 \leq n \leq 255$	$48 \leq \text{data} \leq 57, 65 \leq \text{data} \leq 90,$ $\text{data}=32,36,37,43,45,46,47$
HPRT_BARCODE_ITF	70	$1 \leq n \leq 255$ (even number)	$48 \leq \text{data} \leq 57$
HPRT_BARCODE_CODABAR	71	$1 \leq n \leq 255$	$48 \leq \text{data} \leq 57, 65 \leq \text{data} \leq 68,$

			data =36,43,45,46,47,58
HPRT_BARCODE_CODE93	72	$1 \leq n \leq 255$	$0 \leq \text{data} \leq 127$
HPRT_BARCODE_CODE128	73	$2 \leq n \leq 255$	$0 \leq \text{data} \leq 127$
HPRT_BARCODE_STANDARD_PDF417	101	$2 \leq n \leq 928$	$0 \leq \text{data} \leq 255$
HPRT_BARCODE_TRUNCATED_PDF417	102	$2 \leq n \leq 928$	$0 \leq \text{data} \leq 255$
HPRT_BARCODE_QRCODE1	103	$2 \leq n \leq 928$	$0 \leq \text{data} \leq 255$
HPRT_BARCODE_QRCODE2	104	$2 \leq n \leq 928$	$0 \leq \text{data} \leq 255$

int width

[in] Bar code width effective value range: 2-7, when bar code print width exceeds printable area, bar code print will not be executed. The parameter is ineffective for 2D code.

int height,

[in] Set bar code print height. Effective range:1-255, this parameter is ineffective for 2D code.

int alignment,

[in] Set bar code alignment method.

Alignment Method	Value	Description
HPRT_ALIGNMENT_LEFT	0	Left alignment
HPRT_ALIGNMENT_CENTER	1	Center alignment
HPRT_ALIGNMENT_RIGHT	2	Right alignment

int hriPosition

[in] Set bar code visible character position.

Position	Value	Description
HPRT_BRACODE_HRI_NONE	0	Not print visible character
HPRT_BRACODE_HRI_ABOVE	1	Print visible character above bar code
HPRT_BRACODE_HRI_BELOW	2	Print visible character below bar code
HPRT_BRACODE_HRI_BOTH	3	Print visible character above/below bar code

Return Value:

Error Code	Value	Description
HPRT_SUCCESS	0	Normal
HPRT_E_INVALID_PARAMETER	-1	Invalid parameter
HPRT_E_INVALID_MODEL_TYPE	-3	The model type does not support this function
HPRT_E_BAD_HANDLE	-6	Invalid handle
HPRT_E_IO_PORT_NOT_OPEN	-309	Communication port not open
HPRT_E_IO_WRITE_FAILED	-321	Write failed
HPRT_E_IO_WRITE_TIMEOUT	-322	Write timeout

PrtPrintSymbol

Operation for printing 2D code.

```
int PrtPrintSymbol(  
    void* handle,  
    int type,  
    const TCHAR* data,  
    int errLevel,  
    int width,  
    int height,  
    int alignment  
);
```

Parameter:

void handle*
[in,out] The created target printer object.

int type
[in] 2D code type.

Type	Value	Description
HPRT_BARCODE_STANDARD_PDF417	101	Standard sample PDF417 code
HPRT_BARCODE_TRUNCATED_PDF417	102	Simple sample PDF417 code
HPRT_BARCODE_QRCODE1	103	QR Code sample 1
HPRT_BARCODE_QRCODE2	104	QR Code sample 2

const TCHAR data,*
[in] 2Dcode data.

Data Length	Data Range
$1 \leq n \leq 7089$	$0 \leq \text{data} \leq 255$

int errLevel
[in] 2D code setting error correction level.

Error Correction	Value	Code or Error-tolerant Rate
HPRT_PDF417_ERROR_CORRECTION_LEVEL_0	48	2
HPRT_PDF417_ERROR_CORRECTION_LEVEL_1	49	4
HPRT_PDF417_ERROR_CORRECTION_LEVEL_2	50	8
HPRT_PDF417_ERROR_CORRECTION_LEVEL_3	51	16
HPRT_PDF417_ERROR_CORRECTION_LEVEL_4	52	32
HPRT_PDF417_ERROR_CORRECTION_LEVEL_5	53	64
HPRT_PDF417_ERROR_CORRECTION_LEVEL_6	54	128
HPRT_PDF417_ERROR_CORRECTION_LEVEL_7	55	256
HPRT_PDF417_ERROR_CORRECTION_LEVEL_8	56	512
HPRT_PDF417_ERROR_CORRECTION_LEVEL_L	48	7%
HPRT_PDF417_ERROR_CORRECTION_LEVEL_M	49	15%
HPRT_PDF417_ERROR_CORRECTION_LEVEL_Q	50	25%
HPRT_PDF417_ERROR_CORRECTION_LEVEL_H	51	30%

int width

[in] 2D code width. $0 \leq n \leq 255$

int height

[in] 2D code height. $0 \leq n \leq 255$ (This parameter is ineffective for QR Code.)

int alignment,

[in] 2D code alignment method

Alignment Method	Value	Description
HPRT_ALIGNMENT_LEFT	0	Left alignment
HPRT_ALIGNMENT_CENTER	1	Center alignment
HPRT_ALIGNMENT_RIGHT	2	Right alignment

Return Value:

Error Code	Value	Description
HPRT_SUCCESS	0	Normal
HPRT_E_INVALID_PARAMETER	-1	Invalid parameter
HPRT_E_INVALID_MODEL_TYPE	-3	The model type does not support this function
HPRT_E_BAD_HANDLE	-6	Invalid handle
HPRT_E_IO_PORT_NOT_OPEN	-309	Communication port not open
HPRT_E_IO_WRITE_FAILED	-321	Write failed
HPRT_E_IO_WRITE_TIMEOUT	-322	Write timeout

PrtPrintTwoQRCode

Operation for printing QR code.

```
int PrtPrintTwoQRCode(  
    void* handle,  
    TCHAR* data1,  
    int data1Len,  
    int width1,  
    int hAlign1,  
    int vAlign1,  
    TCHAR* data2,  
    int data2Len,  
    int width2,  
    int hAlign2,  
    int vAlign2  
);
```

Parameter:

void handle*
[in,out] The created target printer object.

TCHAR data1*
[in] QR code data 1.

int data1Len
[in] QR code data 1 length.

Data Length	Data Range
$1 \leq n \leq 7089$	$0 \leq \text{data} \leq 255$

int width1
[in] QR code module 1 width. $0 \leq n \leq 255$

int hAlign1
[in] QR code 1 horizontal alignment method.
[in] When value of hAlign1 is greater than 2, horizontal position of QR code 1 can be defined by user(cannot over current page width).

Alignment Method	Value	Description
HPRT_ALIGNMENT_LEFT	0	Left alignment
HPRT_ALIGNMENT_CENTER	1	Center alignment
HPRT_ALIGNMENT_RIGHT	2	Right alignment

int vAlign1

[in] QR code 1 vertical alignment method.

[in] When value of vAlign1 is greater than 2, vertical position of QR code 1 can be defined by user(cannot over current height, otherwise QR code cannot be printed).

Alignment Method	Value	Description
HPRT_ALIGNMENT_TOP	0	Top alignment
HPRT_ALIGNMENT_CENTER	1	Center alignment
HPRT_ALIGNMENT_BOTTOM	2	Bottom alignment

TCHAR data2*

[in] QR code data 2.

int data2Len

[in] QR code data 2 length.

Data Length	Data Range
$1 \leq n \leq 7089$	$0 \leq \text{data} \leq 255$

int width2

[in] QR code module 2 width. $0 \leq n \leq 255$

int hAlign2

[in] QR code 2 horizontal alignment.

[in] When value of hAlign1 is greater than 2, horizontal position of QR code 1 can be defined by user(cannot over current page width).

Alignment Method	Value	Description
HPRT_ALIGNMENT_LEFT	0	Left alignment
HPRT_ALIGNMENT_CENTER	1	Center alignment
HPRT_ALIGNMENT_RIGHT	2	Right alignment

int vAlign2

[in] QR code 2 vertical alignment method.

[in] When value of vAlign1 is greater than 2, vertical position of QR code 1 can be defined by user(cannot over current height, otherwise QR code cannot be printed).

Alignment Method	Value	Description
HPRT_ALIGNMENT_TOP	0	Top alignment
HPRT_ALIGNMENT_CENTER	1	Center alignment
HPRT_ALIGNMENT_BOTTOM	2	Bottom alignment

Return Value:

Error Code	Value		Description
HPRT_SUCCESS	0	Normal	
HPRT_E_INVALID_PARAMETER	-1	Invalid parameter	
HPRT_E_INVALID_MODEL_TYPE	-3	The model type does not support this function	
HPRT_E_BAD_HANDLE	-6	Invalid handle	
HPRT_E_IO_PORT_NOT_OPEN	-309	Communication port not open	
HPRT_E_IO_WRITE_FAILED	-321	Write failed	
HPRT_E_IO_WRITE_TIMEOUT	-322	Write timeout	

PrtPrintImage

Print specified image (support bmp, jpg, gif, etc.).

```
int PrtPrintImage(  
    void* handle,  
    const TCHAR* filePath,  
    int scaleMode  
);
```

Parameter:

void handle*

[in,out] The created target printer object.

const TCHAR filePath*

[in] The complete path of image.

int scaleMode

[in] The scale mode for printing image.

Mode	Value	Description
HPRT_PRINT_IMAGE_NORMAL	0	Normal mode
HPRT_PRINT_IMAGE_DOUBLE_WIDTH	1	Double-width mode
HPRT_PRINT_IMAGE_DOUBLE_HEIGHT	2	Double-height mode
HPRT_PRINT_IMAGE_QUADRUPLE	3	Quadruple mode

Return Value:

Error Code	Value	Description
HPRT_SUCCESS	0	Normal
HPRT_E_INVALID_PARAMETER	-1	Invalid parameter
HPRT_E_INVALID_MODEL_TYPE	-3	The model type does not support this function
HPRT_E_BAD_HANDLE	-6	Invalid handle
HPRT_E_NOT_ENOUGH_MEMORY	-9	Not enough memory
HPRT_E_IMAGE_BAD_SIZE	-25	Image size error
HPRT_E_IO_PORT_NOT_OPEN	-309	Communication port not open
HPRT_E_IO_WRITE_FAILED	-321	Write failed
HPRT_E_IO_WRITE_TIMEOUT	-322	Write timeout

PrtDefineNVImageCompatible

Define specified NV bit image in NV image buffer area.

```
int PrtDefineNVImageCompatible(  
    void* handle,  
    const TCHAR** filePathList,  
    int imageQty  
);
```

Parameter:

void handle*

[in,out] The created target printer object.

*const TCHAR**] filePathList*

[in] The specified image path list.

int imageQty

[in] The specified image quantity.

Return Value:

Error Code	Value	Description
HPRT_SUCCESS	0	Normal
HPRT_E_INVALID_PARAMETER	-1	Invalid parameter
HPRT_E_INVALID_MODEL_TYPE	-3	The model type does not support this function
HPRT_E_BAD_HANDLE	-6	Invalid handle
HPRT_E_NOT_ENOUGH_MEMORY	-9	Not enough memory
HPRT_E_IMAGE_BAD_SIZE	-25	Image size error
HPRT_E_IO_PORT_NOT_OPEN	-309	Communication port not open
HPRT_E_IO_WRITE_FAILED	-321	Write failed
HPRT_E_IO_WRITE_TIMEOUT	-322	Write timeout

PrtPrintNVImageCompatible

Print the specified NV image in NV image buffer area.

```
int PrintNVImageCompatible(  
    void* handle,  
    int imgNo,  
    int scaleMode  
);
```

Parameter:

void handle*

[in,out] The created target printer object.

int imgNo,

[in] Print specified nth image (the image serial number which undefined in NV buffer area will not be printed). $1 \leq n \leq 255$

int scaleMode,

[in] The scale mode for printing image.

Return Value:

Error Code	Value	Description
HPRT_SUCCESS	0	Normal
HPRT_E_INVALID_PARAMETER	-1	Invalid parameter
HPRT_E_INVALID_MODEL_TYPE	-3	The model type does not support this function
HPRT_E_BAD_HANDLE	-6	Invalid handle
HPRT_E_IO_PORT_NOT_OPEN	-309	Communication port not open
HPRT_E_IO_WRITE_FAILED	-321	Write failed
HPRT_E_IO_WRITE_TIMEOUT	-322	Write timeout

PrtDefineDownloadedImageCompatible

Define downloaded image in image buffer area.

```
int PrtDefineDownloadedImageCompatible(  
    void* handle,  
    const TCHAR* filePath  
);
```

Parameter:

void handle*

[in,out] The created target printer object.

const TCHAR filePath*

[in] The complete path of image.

Return Value:

Error Code	Value	Description
HPRT_SUCCESS	0	Normal
HPRT_E_INVALID_PARAMETER	-1	Invalid parameter
HPRT_E_INVALID_MODEL_TYPE	-3	The model type does not support this function
HPRT_E_BAD_HANDLE	-6	Invalid handle
HPRT_E_NOT_ENOUGH_MEMORY	-9	Not enough memory
HPRT_E_IMAGE_BAD_SIZE	-25	Image size error
HPRT_E_IO_PORT_NOT_OPEN	-309	Communication port not open
HPRT_E_IO_WRITE_FAILED	-321	Write failed
HPRT_E_IO_WRITE_TIMEOUT	-322	Write timeout

PrtPrintDownloadedImageCompatible

Print downloaded image in image buffer area.

```
int PrtPrintDownloadedImageCompatible(  
    void* handle,  
    int scaleMode  
);
```

Parameter:

void handle*

[in,out] The created target printer object.

int scalemode

[in] The scale mode for printing image.

Mode	Value	Description
HPRT_PRINT_IMAGE_NORMAL	0	Normal mode
HPRT_PRINT_IMAGE_DOUBLE_WIDTH	1	Double-width mode
HPRT_PRINT_IMAGE_DOUBLE_HEIGHT	2	Double-height mode
HPRT_PRINT_IMAGE_QUADRUPLE	3	Quadruple mode

Return Value:

Error Code	Value	Description
HPRT_SUCCESS	0	Normal
HPRT_E_INVALID_PARAMETER	-1	Invalid parameter
HPRT_E_INVALID_MODEL_TYPE	-3	The model type does not support this function
HPRT_E_BAD_HANDLE	-6	Invalid handle
HPRT_E_IO_PORT_NOT_OPEN	-309	Communication port not open
HPRT_E_IO_WRITE_FAILED	-321	Write failed
HPRT_E_IO_WRITE_TIMEOUT	-322	Write timeout

PrtGetPrinterVersion

Get the current printer version.

```
int PrtGetPrinterVersion(  
    void* handle,  
    int* version,  
    int versionLen  
);
```

Parameter:

void handle*

[in,out] The created target printer object.

int version*

[in,out] Printer firmware version number, e.g. 1.3.12.

int versionLen

[in] Firmware version data length.

Return Value:

Error Code	Value	Description
HPRT_SUCCESS	0	Normal
HPRT_E_INVALID_MODEL_TYPE	-3	The model type does not support this function
HPRT_E_BAD_HANDLE	-6	Invalid handle
HPRT_E_IO_PORT_NOT_OPEN	-309	Communication port not open
HPRT_E_IO_WRITE_FAILED	-321	Write failed
HPRT_E_IO_WRITE_TIMEOUT	-322	Write timeout
HPRT_E_IO_READ_FAILED	-331	Read failed
HPRT_E_IO_READ_TIMEOUT	-332	Read timeout

PrtSelectPageMode

Printer select page mode (only printer supports page mode and effective in standard mode).

```
int PrtSelectPageMode(  
    void* handle  
);
```

Parameter:

void handle*
[in,out] The created target printer object.

Return Value:

Error Code	Value	Description
HPRT_SUCCESS	0	Normal
HPRT_E_INVALID_MODEL_TYPE	-3	The model type does not support this function
HPRT_E_BAD_HANDLE	-6	Invalid handle
HPRT_E_IO_PORT_NOT_OPEN	-309	Communication port not open
HPRT_E_IO_WRITE_FAILED	-321	Write failed
HPRT_E_IO_WRITE_TIMEOUT	-322	Write timeout

PrtSelectStandardMode

Printer select standard mode and clear the data in print buffer (only effective in page mode).

```
int PrtSelectStandardMode(  
    void* handle  
);
```

Parameter:

void handle*
[in,out] The created target printer object.

Return Value:

Error Code	Value	Description
HPRT_SUCCESS	0	Normal
HPRT_E_INVALID_MODEL_TYPE	-3	The model type does not support this function
HPRT_E_BAD_HANDLE	-6	Invalid handle
HPRT_E_IO_PORT_NOT_OPEN	-309	Communication port not open
HPRT_E_IO_WRITE_FAILED	-321	Write failed
HPRT_E_IO_WRITE_TIMEOUT	-322	Write timeout

PrtSelectPrintDirectionInPageMode

In page mode, select printer print direction.

```
int PrtSelectPrintDirectionInPageMode(  
    void* handle,  
    int direction  
);
```

Parameter:

void handle*

[in,out] The created target printer object.

int direction

[in] Select print direction.

Print Direction	Value	Description	Start Position
HPRT_PRINT_DIRECTION_LEFT_TO_RIGHT	0	Left->Right	Top left corner
HPRT_PRINT_DIRECTION_BOTTOM_TO_TOP	1	Bottom->Top	Bottom left corner
HPRT_PRINT_DIRECTION_RIGHT_TO_LEFT	2	Right->Left	Bottom right corner
HPRT_PRINT_DIRECTION_TOP_TO_BOTTOM	3	Top->Bottom	Top right corner

Return Value:

Error Code	Value	Description
HPRT_SUCCESS	0	Normal
HPRT_E_INVALID_PARAMETER	-1	Invalid parameter
HPRT_E_INVALID_MODEL_TYPE	-3	The model type does not support this function
HPRT_E_BAD_HANDLE	-6	Invalid handle
HPRT_E_IO_PORT_NOT_OPEN	-309	Communication port not open
HPRT_E_IO_WRITE_FAILED	-321	Write failed
HPRT_E_IO_WRITE_TIMEOUT	-322	Write timeout

PrtSetTextPosition

Set print start position (in standard mode, the command will set horizontal print position. In page mode, when print start position is top left corner or bottom right corner, is horizontal position setting; when print start position is bottom left corner or top right corner, is vertical position setting).

```
int PrtSetTextPosition(  
    void* handle,  
    int position  
);
```

Parameter:

void handle*
[in,out] The created target printer object.

int position
[in] Set print start position.

Return Value:

Error Code	Value	Description
HPRT_SUCCESS	0	Normal
HPRT_E_INVALID_PARAMETER	-1	Invalid parameter
HPRT_E_INVALID_MODEL_TYPE	-3	The model type does not support this function
HPRT_E_BAD_HANDLE	-6	Invalid handle
HPRT_E_IO_PORT_NOT_OPEN	-309	Communication port not open
HPRT_E_IO_WRITE_FAILED	-321	Write failed
HPRT_E_IO_WRITE_TIMEOUT	-322	Write timeout

PrtSetAbsoluteVerticalPrintPositionInPageMode

In page mode, set the vertical print position (when print start position is top left corner or bottom right corner, is vertical position setting. When print start position is bottom left corner or top right corner, is horizontal setting).

```
int PrtSetAbsoluteVerticalPrintPositionInPageMode(  
    void* handle,  
    int position  
);
```

Parameter:

void handle*
[in,out] The created target printer object.

int position
[in] Set vertical position.

Return Value:

Error Code	Value	Description
HPRT_SUCCESS	0	Normal
HPRT_E_INVALID_PARAMETER	-1	Invalid parameter
HPRT_E_INVALID_MODEL_TYPE	-3	The model type does not support this function
HPRT_E_BAD_HANDLE	-6	Invalid handle
HPRT_E_IO_PORT_NOT_OPEN	-309	Communication port not open
HPRT_E_IO_WRITE_FAILED	-321	Write failed
HPRT_E_IO_WRITE_TIMEOUT	-322	Write timeout

PrtSetPrintAndReturnStandardMode

Print and return standard mode (only effective in page mode).

```
int PrtSetPrintAndReturnStandardMode(  
    void* handle,  
);
```

Parameter:

void handle*

[in,out] The created target printer object.

Return Value:

Error Code	Value	Description
HPRT_SUCCESS	0	Normal
HPRT_E_INVALID_MODEL_TYPE	-3	The model type does not support this function
HPRT_E_BAD_HANDLE	-6	Invalid handle
HPRT_E_IO_PORT_NOT_OPEN	-309	Communication port not open
HPRT_E_IO_WRITE_FAILED	-321	Write failed
HPRT_E_IO_WRITE_TIMEOUT	-322	Write timeout

PrtSetPrintAreaInPageMode

Set print area in page mode.

```
int PrtSetPrintAreaInPageMode(  
    void* handle,  
    int horizontal,  
    int vertical,  
    int width,  
    int height  
);
```

Parameter:

void handle*

[in,out] The created target printer object.

int horizontal

[in] Set horizontal position of print start.

int vertical

[in] Set vertical position of print start.

int width

[in] Set horizontal width of printable area.

int height

[in] Set vertical height of printable area.

When print width is 80mm: horizontal star point = 0, vertical start point = 0,
width = 576, height = 840.

Return Value:

Error Code	Value	Description
HPRT_SUCCESS	0	Normal
HPRT_E_INVALID_PARAMETER	-1	Invalid parameter
HPRT_E_INVALID_MODEL_TYPE	-3	The model type does not support this function
HPRT_E_BAD_HANDLE	-6	Invalid handle
HPRT_E_IO_PORT_NOT_OPEN	-309	Communication port not open
HPRT_E_IO_WRITE_FAILED	-321	Write failed
HPRT_E_IO_WRITE_TIMEOUT	-322	Write timeout

PrtPrintDataInPageMode

Print data in page mode, and not return standard mode after printing (only effective in page mode).

```
int PrtPrintDataInPageMode(  
    void* handle  
);
```

Parameter:

void handle*
[in,out] The created target printer object.

Return Value:

Error Code	Value	Description
HPRT_SUCCESS	0	Normal
HPRT_E_INVALID_MODEL_TYPE	-3	The model type does not support this function
HPRT_E_BAD_HANDLE	-6	Invalid handle
HPRT_E_IO_PORT_NOT_OPEN	-309	Communication port not open
HPRT_E_IO_WRITE_FAILED	-321	Write failed
HPRT_E_IO_WRITE_TIMEOUT	-322	Write timeout

PrtDirectIO

User-defined send and read printer data.

```
int PrtDirectIO(  
    void* handle,  
    unsigned char* writeData,  
    unsigned int writeNum,  
    unsigned char* readData,  
    unsigned int readNum,  
    unsigned int* preadedNum  
);
```

Parameter:

void handle*

[in,out] The created target printer object.

unsigned char writeData*

[in] Printer write data.

unsigned int writeNum,

[in] The number of write data.

unsigned char readData,*

[in,out] Get the printer return data.

unsigned int readNum,

[in] Pre-set the number of data that need to read.

unsigned int preadedNum*

[in,out] The number of actual read data.

Return Value:

Error Code	Value	Description
HPRT_SUCCESS	0	Normal
HPRT_E_BAD_HANDLE	-6	Invalid handle
HPRT_E_IO_PORT_NOT_OPEN	-309	Communication port not open
HPRT_E_IO_WRITE_FAILED	-321	Write failed
HPRT_E_IO_WRITE_TIMEOUT	-322	Write timeout
HPRT_E_IO_READ_FAILED	-331	Read failed
HPRT_E_IO_READ_TIMEOUT	-332	Read timeout

State (int R)

Get the printer state path. State is read-only, when calling the function of checking printer state, it will return to state automatically. State value can be read repeatedly, every value can be confirmed by bit operation. Call the state value by getting the state function.

Printer state value is as below:

Code	Value	Description
STS_NORMAL	0	Printer state is normal
STS_PAPEREMPTY	1	Paper out
STS_COVEROPEN	2	Upper cover open
STS_PAPERNEAREND	4	Paper near end
STS_ERROR	32	Printer error
STS_NOT_OPEN	64	Printer port not open
STS_OFFLINE	128	Printer off line

CharacterSet (int R/W)

Set the property of code page, default is Chinese, use SetCharacterSet() to set.

Code page please refer to below:

Character Set	Value	Description
CHARACTERSET_DEFAULT	0	Default
CHARACTERSET_USA	437	English character set
CHARACTERSET_MULTILINGUAL	850	Multilingual character set
CHARACTERSET_LATIN2	852	Latin character set
CHARACTERSET_EURO	858	European character set
CHARACTERSET_PORTUGUESE	860	Portuguese character set
CHARACTERSET_CANADIAN_FREN	863	Canadian-French character set
CHARACTERSET_NORDIC	865	Nordic character set
CHARACTERSET_CYRILLIC2	866	Cyrillic character set
CHARACTERSET_WPC1252	1252	WPC character set

PortOpen

Open the communication port to operate printing. Currently it supports USB, internet, serial interface and LPT.

```
int PortOpen(  
    String portSetting  
);
```

Parameter:

string portSetting

[in] Set up the parameter of communication port that connected to the target printer.

Configuration List:

Type	Configuration	Description	Sample
USB	USB [,Position]	USB: connect any USB printer of our company USB[,Position]: When connecting to multi printers of our company, can specify connecting to one particular USB printer through USB position information (Position parameter)	USB USB,Port_#0004.Hub_#0003
NET	NET , IP Add (IPv4)[,Port]	Specify the IP add and port of internet printer. If not specifying port, the default port is 9100.	NET,192.168.0.36 NET,192.168.0.36,9100
COM	COMn ,BAUDRATE_rate	Specify the number and baud rate of connected serial port .	COM5,BAUDRATE_19200
LPT	LPTn	Specify the number of connected parallel port.	LPT1

Note: [] indicates selective parameter

How to check the information of USB printer position (Position parameter):

In Windows device manager, unfold “USB controller” , select “USB print support” device, select “Property” on right click menu, click “Detail Information” .

The value of Property “Position Information” is the position information of USB printer. (Position parameter)

Return Value:

Error Code	Value	Description
E_SUCCESS	0	Normal
E_INVALID_PARAMETER	-1	Invalid parameter
E_BAD_HANDLE	-6	Invalid handle
E_IO_PORT_OPEN_FAILED	-311	Port open failed

PortClose

Operation for closing the port (not using SDK function).

```
int PortClose(  
);
```

Parameter:

N/A.

Return Value:

Error Code	Value	Description
E_SUCCESS	0	Normal
E_BAD_HANDLE	-6	Invalid handle

Initialize

Initialize the printer(Clear the data of printing buffer and reset printer temporary setting items).

```
int Initialize(  
    );
```

Parameter:

N/A.

Return Value:

Error Code	Value	Description
E_SUCCESS	0	Normal
E_BAD_HANDLE	-6	Invalid handle
E_IO_PORT_NOT_OPEN	-309	Communication port not open
E_IO_WRITE_FAILED	-321	Write failed
E_IO_WRITE_TIMEOUT	-322	Write timeout

SetTextLineSpace

Set print text line spacing.

```
int SetTextLineSpace(  
    int lineSpace  
);
```

Parameter:

int lineSpace

[in] Set text line spacing. $0 \leq \text{lineSpace} \leq 255$

Return Value:

Error Code	Value	Description
E_SUCCESS	0	Normal
E_INVALID_PARAMETER	-1	Invalid parameter
E_BAD_HANDLE	-6	Invalid handle
E_IO_PORT_NOT_OPEN	-309	Communication port not open
E_IO_WRITE_FAILED	-321	Write failed
E_IO_WRITE_TIMEOUT	-322	Write timeout

CheckPrinterState

Operation for getting the printer real-time state.

```
int CheckPrinterState(  
    ref uint printerState  
);
```

Parameter:

ref uint printerState

[in,out] Pinter real-time state, when returning to multi-state, value is showed as accumulation, return state please refer to below:

Error Code	Value	Description
STS_NORMAL	0	Normal
STS_PAPEREMPTY	1	Paper out
STS_COVEROPEN	2	Upper cover opened
STS_PAPERNEAREND	4	Paper near end
STS_ERROR	32	Error occured when gettting state
STS_NOT_OPEN	64	Port not open
STS_OFFLINE	128	Printer off line

Return Value:

Error Code	Value	Description
E_SUCCESS	0	Normal
E_INVALID_MODEL_TYPE	-3	The model type does not support this function
E_BAD_HANDLE	-6	Invalid handle
E_IO_PORT_NOT_OPEN	-309	Communication port not open
E_IO_WRITE_FAILED	-321	Write failed
E_IO_WRITE_TIMEOUT	-322	Write timeout
E_IO_READ_FAILED	-331	Read failed
E_IO_READ_TIMEOUT	-332	Read timeout

SetCharacterSet

Set character set.

```
int SetCharacterSet(  
    int characterSet  
);
```

Parameter:

int characterSet

[in] Select character set setting.
Default: Chinese character set.

Character Set	Value	Description
CHARACTERSET_DEFAULT	0	Default
CHARACTERSET_USA	437	English character set
CHARACTERSET_MULTILINGUAL	850	Multilingual character set
CHARACTERSET_LATIN2	852	Latin character set
CHARACTERSET_EURO	858	European character set
CHARACTERSET_PORTUGUESE	860	Portuguese character set
CHARACTERSET_CANADIAN_FRE	863	Canadian-French character set
CHARACTERSET_NORDIC	865	Nordic character set
CHARACTERSET_CYRILLIC2	866	Cyrillic character set
CHARACTERSET_WPC1252	1252	WPC character set

Return Value:

Error Code	Value	Description
E_SUCCESS	0	Normal
E_INVALID_PARAMETER	-1	Invalid parameter
E_INVALID_MODEL_TYPE	-3	The model type does not support this function
E_BAD_HANDLE	-6	Invalid handle
E_IO_PORT_NOT_OPEN	-309	Communication port not open
E_IO_WRITE_FAILED	-321	Write failed
E_IO_WRITE_TIMEOUT	-322	Write timeout

CutPaper

Operation for printer paper feed and cut paper.

```
int CutPaper(  
    int cutMode,  
    int distance  
);
```

Parameter:

int cutMode

[in] Paper cut mode.

Paper Cut Mode	Value	Description
FULL_CUT	0	Full cut
PARTIAL_CUT	-1	Partial cut

int distance

[in] Paper feed distance.

Return Value:

Error Code	Value	Description
E_SUCCESS	0	Normal
E_INVALID_PARAMETER	-1	Invalid parameter
E_INVALID_MODEL_TYPE	-3	The model type does not support this function
E_BAD_HANDLE	-6	Invalid handle
E_IO_PORT_NOT_OPEN	-309	Communication port not open
E_IO_WRITE_FAILED	-321	Write failed
E_IO_WRITE_TIMEOUT	-322	Write timeout

LineFeed

Operation for printer paper feed.

```
int LineFeed(  
    int nFeed  
);
```

Parameter:

int nFeed
[in] Set lines of paper feed.

Return Value:

Error Code	Value	Description
E_SUCCESS	0	Normal
E_INVALID_PARAMETER	-1	Invalid parameter
E_BAD_HANDLE	-6	Invalid handle
E_IO_PORT_NOT_OPEN	-309	Communication port not open
E_IO_WRITE_FAILED	-321	Write failed
E_IO_WRITE_TIMEOUT	-322	Write timeout

PrintText

Operation for printing printer text.

```
int PrintText(  
    string data,  
    int alignment,  
    int attribute,  
    int textSize  
);
```

string data

[in] The text data which need to print.

int alignment

[in] The alignment method of text.

Alignment Method	Value	Description
ALIGNMENT_LEFT	0	Left alignment
ALIGNMENT_CENTER	1	Center alignment
ALIGNMENT_RIGHT	2	Right alignment

int attribute

[in] Set text font property, property setting can be accumulated.

Text Property	Value	Description
TEXT_NORMAL_MODE	0	Default setting font A
TEXT_FONT_BOLD	2	Set font bold
TEXT_FONT_UNDERLINE_MODE	4	Set font underline mode
TEXT_FONT_REVERSE	8	Set font reverse
TEXT_FONT_DW_DH_MODE	48	Set font double-width and double-height

int textSize

[in] Set the text size (It will not be printed if the text size exceeds print paper area).

Set text width:

Text Width	Value	Description
TEXT_SIZE_0WIDTH	0	Text width × 1
TEXT_SIZE_1WIDTH	16	Text width × 2
TEXT_SIZE_2WIDTH	32	Text width × 3
TEXT_SIZE_3WIDTH	48	Text width × 4
TEXT_SIZE_4WIDTH	64	Text width × 5
TEXT_SIZE_5WIDTH	80	Text width × 6
TEXT_SIZE_6WIDTH	96	Text width × 7
TEXT_SIZE_7WIDTH	112	Text width × 8

Set text height:

Text Height	Value	Description
TEXT_SIZE_0HEIGHT	0	Text height × 1
TEXT_SIZE_1HEIGHT	1	Text height × 2
TEXT_SIZE_2HEIGHT	2	Text height × 3
TEXT_SIZE_3HEIGHT	3	Text height × 4
TEXT_SIZE_4HEIGHT	4	Text height × 5
TEXT_SIZE_5HEIGHT	5	Text height × 6
TEXT_SIZE_6HEIGHT	6	Text height × 7
TEXT_SIZE_7HEIGHT	7	Text height × 8

Return Value:

Error Code	Value	Description
E_SUCCESS	0	Normal
E_INVALID_PARAMETER	-1	Invalid parameter
E_BAD_HANDLE	-6	Invalid handle
E_IO_PORT_NOT_OPEN	-309	Communication port not open
E_IO_WRITE_FAILED	-321	Write failed
E_IO_WRITE_TIMEOUT	-322	Write timeout

PrintBarcode

Operation for printing bar code. In standard mode, only when bar code print position is in new line or without data in buffer can print normally. In page mode, when the command of printing bar code is not received, the bar code data will be saved in buffer and print will not be executed.

```
int PrintBarcode(  
    int bcType,  
    string bcData,  
    int width,  
    int height,  
    int alignment,  
    int hriPosition  
    );
```

Parameter:

int bcType
[in] Set bar code type.

string bcData,
[in] Send printer bar code data.

Bar Code Type	Value	Bar Code Data Length	Effective Data Range
BARCODE_UPC_A	65	$11 \leq n \leq 12$	$48 \leq \text{data} \leq 57$
BARCODE_UPC_E	66	$11 \leq n \leq 12$	$48 \leq \text{data} \leq 57$
BARCODE_EAN13	67	$12 \leq n \leq 13$	$48 \leq \text{data} \leq 57$
BARCODE_JAN13			
BARCODE_EAN8	68	$7 \leq n \leq 8$	$48 \leq \text{data} \leq 57$
BARCODE_JAN8			
BARCODE_CODE39	69	$1 \leq n \leq 255$	$48 \leq \text{data} \leq 57, 65 \leq \text{data} \leq 90,$ $\text{data} = 32, 36, 37, 43, 45, 46, 47$
BARCODE_ITF	70	$1 \leq n \leq 255$ (even number)	$48 \leq \text{data} \leq 57$
BARCODE_CODABAR	71	$1 \leq n \leq 255$	$48 \leq \text{data} \leq 57, 65 \leq \text{data} \leq 68,$ $\text{data} = 36, 43, 45, 46, 47, 58$
BARCODE_CODE93	72	$1 \leq n \leq 255$	$0 \leq \text{data} \leq 127$
BARCODE_CODE128	73	$2 \leq n \leq 255$	$0 \leq \text{data} \leq 127$

BARCODE_STANDARD_PDF 417	101	$2 \leq n \leq 928$	$0 \leq \text{data} \leq 255$
BARCODE_TRUNCATED_PD F417	102	$2 \leq n \leq 928$	$0 \leq \text{data} \leq 255$
BARCODE_QRCODE1	103	$2 \leq n \leq 928$	$0 \leq \text{data} \leq 255$
BARCODE_QRCODE2	104	$2 \leq n \leq 928$	$0 \leq \text{data} \leq 255$

int width

[in] Bar code width effective value range: 2-7, when bar code print width exceeds printerable area, bar code print will not be executed. The parameter is ineffective for 2D code.

int height,

[in] Set bar code print height. Effective range:1-255, this parameter is ineffective for 2D code.

int alignment,

[in] Set bar code alignment method.

Alignment Method	Value	Description
ALIGNMENT_LEFT	0	Left alignment
ALIGNMENT_CENTER	1	Center alignment
ALIGNMENT_RIGHT	2	Right alignment

int hriPosition

[in] Set bar code visible character position.

Position	Value	Description
BRACODE_HRI_NONE	0	Not print visible character
BRACODE_HRI_ABOVE	1	Print visible character above bar code
BRACODE_HRI_BELOW	2	Print visible character below bar code
BRACODE_HRI_BOTH	3	Print visible character above/below bar code

Return Value:

Error Code	Value	Description
E_SUCCESS	0	Normal
E_INVALID_PARAMETER	-1	Invalid parameter
E_INVALID_MODEL_TYPE	-3	The model type does not support this function
E_BAD_HANDLE	-6	Invalid handle
E_IO_PORT_NOT_OPEN	-309	Communication port not open
E_IO_WRITE_FAILED	-321	Write failed
E_IO_WRITE_TIMEOUT	-322	Write timeout

PrintSymbol

Operation for printing 2D code.

```
int PrintSymbol(  
    int bcType,  
    string bcData,  
    int errLevel,  
    int width,  
    int height,  
    int alignment  
);
```

Parameter:

int bcType
[in] 2D code type.

Type	Value	Description
BARCODE_STANDARD_PDF417	101	Standard sample PDF417 code
BARCODE_TRUNCATED_PDF417	102	Simple sample PDF417 code
BARCODE_QRCODE1	103	QR Code sample 1
BARCODE_QRCODE2	104	QR Code sample 2

string bcData,
[in] 2Dcode data.

Data Length	Data Range
$1 \leq n \leq 7089$	$0 \leq \text{data} \leq 255$

int errLevel
[in] 2D code setting error correction level.

Error Correction	Value	Code or Error-tolerant Rate
PDF417_ERROR_CORRECTION_LEVEL_0	48	2
PDF417_ERROR_CORRECTION_LEVEL_1	49	4
PDF417_ERROR_CORRECTION_LEVEL_2	50	8
PDF417_ERROR_CORRECTION_LEVEL_3	51	16
PDF417_ERROR_CORRECTION_LEVEL_4	52	32
PDF417_ERROR_CORRECTION_LEVEL_5	53	64
PDF417_ERROR_CORRECTION_LEVEL_6	54	128
PDF417_ERROR_CORRECTION_LEVEL_7	55	256
PDF417_ERROR_CORRECTION_LEVEL_8	56	512
QRCODE_ERROR_CORRECTION_LEVEL_L	48	7%
QRCODE_ERROR_CORRECTION_LEVEL_M	49	15%
QRCODE_ERROR_CORRECTION_LEVEL_Q	50	25%
QRCODE_ERROR_CORRECTION_LEVEL_H	51	30%

int width

[in] 2D code width. $0 \leq n \leq 255$

int height

[in] 2D code height. $0 \leq n \leq 255$ (This parameter is ineffective for QR Code.)

int alignment,

[in] 2D code alignment method

Alignment Method	Value	Description
ALIGNMENT_LEFT	0	Left alignment
ALIGNMENT_CENTER	1	Center alignment
ALIGNMENT_RIGHT	2	Right alignment

Return Value:

Error Code	Value	Description
E_SUCCESS	0	Normal
E_INVALID_PARAMETER	-1	Invalid parameter
E_INVALID_MODEL_TYPE	-3	The model type does not support this function
E_BAD_HANDLE	-6	Invalid handle
E_IO_PORT_NOT_OPEN	-309	Communication port not open
E_IO_WRITE_FAILED	-321	Write failed
E_IO_WRITE_TIMEOUT	-322	Write timeout

PrintTwoQRCode

Operation for printing QR code.

```
int PrintTwoQRCode(  
    string data1,  
    int data1Len,  
    int width1,  
    int hAlign1,  
    int vAlign1,  
    string data2,  
    int data2Len,  
    int width2,  
    int hAlign2,  
    int vAlign2  
);
```

Parameter:

string data1

[in] QR code data 1.

Data Length

$1 \leq n \leq 7089$

Data Range

$0 \leq \text{data} \leq 255$

int width1

[in] QR code module 1 width. $0 \leq n \leq 255$

int hAlign1

[in] QR code 1 horizontal alignment method.

[in] When value of hAlign1 is greater than 2, horizontal position of QR code 1 can be defined by user(cannot over current page width).

Alignment Method	Value	Description
ALIGNMENT_LEFT	0	Left alignment
ALIGNMENT_CENTER	1	Center alignment
ALIGNMENT_RIGHT	2	Right alignment

int vAlign1

[in] QR code 1 vertical alignment method.

[in] When value of vAlign1 is greater than 2, vertical position of QR code 1 can be defined by user(cannot over current height, otherwise QR code cannot be printed).

Alignment Method	Value	Description
ALIGNMENT_TOP	0	Top alignment
ALIGNMENT_CENTER	1	Center alignment
ALIGNMENT_BOTTOM	2	Bottom alignment

string data2

[in] QR code data 2.

Data Length	Data Range
$1 \leq n \leq 7089$	$0 \leq \text{data} \leq 255$

int width2

[in] QR code module 2 width. $0 \leq n \leq 255$

int hAlign2

[in] QR code 2 horizontal alignment.

[in] When value of hAlign1 is greater than 2, horizontal position of QR code 1 can be defined by user(cannot over current page width).

Alignment Method	Value	Description
ALIGNMENT_LEFT	0	Left alignment
ALIGNMENT_CENTER	1	Center alignment
ALIGNMENT_RIGHT	2	Right alignment

int vAlign2

[in] QR code 2 vertical alignment method.

[in] When value of vAlign1 is greater than 2, vertical position of QR code 1 can be defined by user(cannot over current height, otherwise QR code cannot be printed).

Alignment Method	Value	Description
ALIGNMENT_TOP	0	Top alignment
ALIGNMENT_CENTER	1	Center alignment
ALIGNMENT_BOTTOM	2	Bottom alignment

Return Value:

Error Code	Value	Description
E_SUCCESS	0	Normal
E_INVALID_PARAMETER	-1	Invalid parameter
E_INVALID_MODEL_TYPE	-3	The model type does not support this function
E_BAD_HANDLE	-6	Invalid handle
E_IO_PORT_NOT_OPEN	-309	Communication port not open
E_IO_WRITE_FAILED	-321	Write failed
E_IO_WRITE_TIMEOUT	-322	Write timeout

PrintImage

Print specified image (support bmp, jpg, gif, etc.).

```
int PrintImage(  
    string filePath,  
    int scaleMode  
);
```

Parameter:

string filePath

[in] The complete path of image.

int scaleMode

[in] The scale mode for printing image.

Mode	Value	Description
PRINT_IMAGE_NORMAL	0	Normal mode
PRINT_IMAGE_DOUBLE_WIDTH	1	Double-width mode
PRINT_IMAGE_DOUBLE_HEIGHT	2	Double-height mode
PRINT_IMAGE_QUADRUPLE	3	Quadruple mode

Return Value:

Error Code	Value	Description
E_SUCCESS	0	Normal
E_INVALID_PARAMETER	-1	Invalid parameter
E_INVALID_MODEL_TYPE	-3	The model type does not support this function
E_BAD_HANDLE	-6	Invalid handle
E_NOT_ENOUGH_MEMORY	-9	Not enough memory
E_IMAGE_BAD_SIZE	-25	Image size error
E_IO_PORT_NOT_OPEN	-309	Communication port not open
E_IO_WRITE_FAILED	-321	Write failed
E_IO_WRITE_TIMEOUT	-322	Write timeout

DefineNVImageCompatible

Define specified NV bit image in NV image buffer area.

```
int DefineNVImageCompatible(  
    String [] fileList,  
    int imageQty  
);
```

Parameter:

string [] fileList

[in] The specified image path list.

int imageQty

[in] The specified image quantity.

Return Value:

Error Code	Value	Description
E_SUCCESS	0	Normal
E_INVALID_PARAMETER	-1	Invalid parameter
E_INVALID_MODEL_TYPE	-3	The model type does not support this function
E_BAD_HANDLE	-6	Invalid handle
E_NOT_ENOUGH_MEMORY	-9	Not enough memory
E_IMAGE_BAD_SIZE	-25	Image size error
E_IO_PORT_NOT_OPEN	-309	Communication port not open
E_IO_WRITE_FAILED	-321	Write failed
E_IO_WRITE_TIMEOUT	-322	Write timeout

PrintNVImageCompatible

Print the specified NV image in NV image buffer area.

```
int PrintNVImageCompatible(  
    int imgNo,  
    int scaleMode  
);
```

Parameter:

int imgNo,

[in] Print specified nth image (the image serial number which undefined in NV buffer area will not be printed). $1 \leq n \leq 255$

int scaleMode,

[in] The scale mode for printing image.

Mode	Value	Description
PRINT_IMAGE_NORMAL	0	Normal mode
PRINT_IMAGE_DOUBLE_WIDTH	1	Double-width mode
PRINT_IMAGE_DOUBLE_HEIGHT	2	Double-height mode
PRINT_IMAGE_QUADRUPLE	3	Quadruple mode

Return Value:

Error Code	Value	Description
E_SUCCESS	0	Normal
E_INVALID_PARAMETER	-1	Invalid parameter
E_INVALID_MODEL_TYPE	-3	The model type does not support this function
E_BAD_HANDLE	-6	Invalid handle
E_IO_PORT_NOT_OPEN	-309	Communication port not open
E_IO_WRITE_FAILED	-321	Write failed
E_IO_WRITE_TIMEOUT	-322	Write timeout

DefineDownloadedImageCompatible

Define downloaded image in image buffer area.

```
int DefineDownloadedImageCompatible(  
    string filePath  
);
```

Parameter:

string filePath

[in] The complete path of image.

Return Value:

Error Code	Value	Description
E_SUCCESS	0	Normal
E_INVALID_PARAMETER	-1	Invalid parameter
E_INVALID_MODEL_TYPE	-3	The model type does not support this function
E_BAD_HANDLE	-6	Invalid handle
E_NOT_ENOUGH_MEMORY	-9	Not enough memory
E_IMAGE_BAD_SIZE	-25	Image size error
E_IO_PORT_NOT_OPEN	-309	Communication port not open
E_IO_WRITE_FAILED	-321	Write failed
E_IO_WRITE_TIMEOUT	-322	Write timeout

PrintDownloadedImageCompatible

Print downloaded image in image buffer area.

```
int PrintDownloadedImageCompatible(  
    int scaleMode  
);
```

Parameter:

int scalemode

[in] The scale mode for printing image.

Mode	Value	Description
PRINT_IMAGE_NORMAL	0	Normal mode
PRINT_IMAGE_DOUBLE_WIDTH	1	Double-width mode
PRINT_IMAGE_DOUBLE_HEIGHT	2	Double-height mode
PRINT_IMAGE_QUADRUPLE	3	Quadruple mode

Return Value:

Error Code	Value	Description
E_SUCCESS	0	Normal
E_INVALID_PARAMETER	-1	Invalid parameter
E_INVALID_MODEL_TYPE	-3	The model type does not support this function
E_BAD_HANDLE	-6	Invalid handle
E_IO_PORT_NOT_OPEN	-309	Communication port not open
E_IO_WRITE_FAILED	-321	Write failed
E_IO_WRITE_TIMEOUT	-322	Write timeout

GetPrinterVersion

Get the current printer version.

```
int GetPrinterVersion(  
    Int [] version,  
    );
```

Parameter:

int [] version

[in,out] Printer firmware version number, e.g. 1.3.12.

Return Value:

Error Code	Value	Description
E_SUCCESS	0	Normal
E_INVALID_MODEL_TYPE	-3	The model type does not support this function
E_BAD_HANDLE	-6	Invalid handle
E_IO_PORT_NOT_OPEN	-309	Communication port not open
E_IO_WRITE_FAILED	-321	Write failed
E_IO_WRITE_TIMEOUT	-322	Write timeout
E_IO_READ_FAILED	-331	Read failed
E_IO_READ_TIMEOUT	-332	Read timeout

SelectPageMode

Printer select page mode (only printer supports page mode and effective in standard mode).

```
int SelectPageMode(  
    );
```

Parameter:

N/A.

Return Value:

Error Code	Value	Description
E_SUCCESS	0	Normal
E_INVALID_MODEL_TYPE	-3	The model type does not support this function
E_BAD_HANDLE	-6	Invalid handle
E_IO_PORT_NOT_OPEN	-309	Communication port not open
E_IO_WRITE_FAILED	-321	Write failed
E_IO_WRITE_TIMEOUT	-322	Write timeout

SelectStandardMode

Printer select standard mode and clear the data in printing buffer (only effective in page mode).

```
int SelectStandardMode(  
    );
```

Parameter:

N/A.

Return Value:

Error Code	Value	Description
E_SUCCESS	0	Normal
E_INVALID_MODEL_TYPE	-3	The model type does not support this function
E_BAD_HANDLE	-6	Invalid handle
E_IO_PORT_NOT_OPEN	-309	Communication port not open
E_IO_WRITE_FAILED	-321	Write failed
E_IO_WRITE_TIMEOUT	-322	Write timeout

SetPageModePrintDirection

In page mode, select printer print direction.

```
int SetPrintDirectionInPageMode(  
    int direction  
);
```

Parameter:

int direction

[in] Select print direction.

Print Direction	Value	Description	Start Position
PRINT_DIRECTION_LEFT_TO_RIGHT	0	Left->Right	Top left corner
PRINT_DIRECTION_BOTTOM_TO_TOP	1	Bottom->Top	Bottom left corner
PRINT_DIRECTION_RIGHT_TO_LEFT	2	Right->Left	Bottom right corner
PRINT_DIRECTION_TOP_TO_BOTTOM	3	Top->Bottom	Top right corner

Return Value:

Error Code	Value	Description
E_SUCCESS	0	Normal
E_INVALID_PARAMETER	-1	Invalid parameter
E_INVALID_MODEL_TYPE	-3	The model type does not support this function
E_BAD_HANDLE	-6	Invalid handle
E_IO_PORT_NOT_OPEN	-309	Communication port not open
E_IO_WRITE_FAILED	-321	Write failed
E_IO_WRITE_TIMEOUT	-322	Write timeout

SetPageModeHorizontalPosition

Set the print start position (In standard mode, the command set print horizontal position. In page mode, when print start position is top left corner or bottom right corner, is horizontal position setting; when start position is bottom left corner or top right corner, is vertical position setting).

```
int SetAbsoluteVerticalPrintPositionInPageMode(  
    int position  
);
```

Parameter:

int position
[in] Print start position.

Return Value:

Error Code	Value	Description
E_SUCCESS	0	Normal
E_INVALID_PARAMETER	-1	Invalid parameter
E_INVALID_MODEL_TYPE	-3	The model type does not support this function
E_BAD_HANDLE	-6	Invalid handle
E_IO_PORT_NOT_OPEN	-309	Communication port not open
E_IO_WRITE_FAILED	-321	Write failed
E_IO_WRITE_TIMEOUT	-322	Write timeout

SetPageModeVerticalPosition

In page mode, set the vertical print position (when print start position is top left corner or bottom right corner, is vertical position setting. When print start position is bottom left corner or top right corner, is horizontal setting).

```
int SetPageModeVerticalPosition(  
    int position  
);
```

Parameter:

int position
[in] Set vertical position.

Return Value:

Error Code	Value	Description
E_SUCCESS	0	Normal
E_INVALID_PARAMETER	-1	Invalid parameter
E_INVALID_MODEL_TYPE	-3	The model type does not support this function
E_BAD_HANDLE	-6	Invalid handle
E_IO_PORT_NOT_OPEN	-309	Communication port not open
E_IO_WRITE_FAILED	-321	Write failed
E_IO_WRITE_TIMEOUT	-322	Write timeout

SetPrintAndReturnStandardMode

Print and return standard mode (only effective in page mode).

```
int SetPrintAndReturnStandardMode(  
    );
```

Parameter:

N/A.

Return Value:

Error Code	Value	Description
E_SUCCESS	0	Normal
E_INVALID_MODEL_TYPE	-3	The model type does not support this function
E_BAD_HANDLE	-6	Invalid handle
E_IO_PORT_NOT_OPEN	-309	Communication port not open
E_IO_WRITE_FAILED	-321	Write failed
E_IO_WRITE_TIMEOUT	-322	Write timeout

SetPageModePrintArea

Set print area in page mode.

```
int PrtSetPrintAreaInPageMode(  
    int horizontal,  
    int vertical,  
    int width,  
    int height  
);
```

Parameter:

int horizontal

[in] Set horizontal position of start print.

int vertical

[in] Set vertical position of start print.

int width

[in] Set horizontal width of printable area.

int height

[in] Set vertical height of printable area.

When print width is 80mm: horizontal start point = 0, vertical start point = 0,
width = 576, height = 840.

Return Value:

Error Code	Value	Description
E_SUCCESS	0	Normal
E_INVALID_PARAMETER	-1	Invalid parameter
E_INVALID_MODEL_TYPE	-3	The model type does not support this function
E_BAD_HANDLE	-6	Invalid handle
E_IO_PORT_NOT_OPEN	-309	Communication port not open
E_IO_WRITE_FAILED	-321	Write failed
E_IO_WRITE_TIMEOUT	-322	Write timeout

PrintDataInPageMode

Print data in page mode, and not return standard mode after printing (only effective in page mode).

```
int PrtPrintDataInPageMode(  
    );
```

Parameter:

N/A.

Return Value:

Error Code	Value	Description
E_SUCCESS	0	Normal
E_INVALID_MODEL_TYPE	-3	The model type does not support this function
E_BAD_HANDLE	-6	Invalid handle
E_IO_PORT_NOT_OPEN	-309	Communication port not open
E_IO_WRITE_FAILED	-321	Write failed
E_IO_WRITE_TIMEOUT	-322	Write timeout

DirectIO

User-defined send and read printer data.

```
int DirectIO(  
    byte [] data,  
    byte [] readData,  
    int readNum,  
    ref int readedNum  
);
```

Parameter:

byte [] data,
[in] The data sent to the printer, which is Hex character string.

byte [] readData,
[in,out] The actual number of read data.

int readNum,
[in] Set the number of read data for read operation.

ref int readedNum*
[in,out] The number of actual read data by calling read operation.

Return Value:

Error Code	Value	Description
E_SUCCESS	0	Normal
E_BAD_HANDLE	-6	Invalid handle
E_IO_PORT_NOT_OPEN	-309	Communication port not open
E_IO_WRITE_FAILED	-321	Write failed
E_IO_WRITE_TIMEOUT	-322	Write timeout
E_IO_READ_FAILED	-331	Read failed
E_IO_READ_TIMEOUT	-332	Read timeout