

# **TSPL Windows SDK Manual**

## **2.3.1**

## 1. UpdateRecords

Version	Text content	Editor
2.1.0	Add wide character type to image path	dan
2.1.3	Text supports no alignment parameters	dan
2.1.5	Join the Nuget repository	dan
2.1.8	Add GPIO settings	liang
2.1.9	Add black mark sensor switch interface	dan

## 2. Information of the Manual

This SDK manual provides the dll file information for Windows application development. We are constantly striving to improve and upgrade the functionality and quality of all our products. Afterwards, the product specifications and user manual may be changed. Please contact our customer service to confirm the latest version.

## 3. Operation System

Windows 10 or above

## 4. Remark

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

The printer resolution is 200 dpi,1 mm=8 dot;The printer resolution is 300 dpi,1 mm=12 dot.

## 5. Method

### 5.1. PrinterCreator

Set up the target printer of specified model (the printer object must be created before any printer operation).

```
int PrinterCreator(void** handle, const char* model);
```

**Parameter:**

*void\*\* handle*

[in,out] The created target printer object.

*const char\* model*

[in] Specify the model of target printer.

**Return Value:**

Code	Value	Description
ERROR_CM_SUCCESS	0	Success
ERROR_CM_INVALID_PARAMETER	-1	Invalid argument

## 5.2. ReleasePrinter

The method is to release the resources of the printer object (the created printer object must be released after the operation is completed ).

```
int ReleasePrinter(void* hPrinter);
```

**Parameter:**

*void\* hPrinter*

[in] Handle to the target printer object that needs to be released

**Return Value:**

Code	Value	Description
ERROR_CM_SUCCESS	0	Success
ERROR_CM_INVALID_PARAMETER	-1	Invalid argument
ERROR_CM_INVALID_HANDLE	-2	Invalid Handle
ERROR_CM_INSUFFICIENT_MEMORY	-4	Failed, out of memory

## 5.3. OpenPort

Open the communication port and connect with the printer. After successfully connected, other functions can be used. If failed connecting, please check the error information. Currently it supports USB, internet, serial interface.

```
int OpenPort(void* handle,const TCHAR* setting);
```

```
int OpenPortA(void* handle, const char* setting);
```

**Parameter:**

*void\* handle*

[in] Printer object handle

*setting*

[in] Set the communication port parameters to connect to the target printer. See the table below for details:

Configuration List:

Category	Configuration	Description	Example
USB	USB,Model/PortNum	USB,printer model USB,The port number If you connect multiple printers of different models of our company at the same time, it is recommended to use "USB, model" to connect	USB,4B-2054A USB,USB031
NET	NET, IP address(IPV4)[,port]	Specify the IPAddress and port.If no port is specified,The default port is 9100.	NET,192.168.1.10 NET,192.168.1.10,9100
COM	COMn,rate	Specify the number and baud rate of connected serial port	COM10,19200
LPT	LPTn	Specify the number of connected parallel port.	LPT3

**Return Value:**

Code	Value	Description
ERROR_CM_SUCCESS	0	Success
ERROR_CM_INVALID_PARAMETER	-1	Invalid argument
ERROR_CM_INVALID_HANDLE	-2	Invalid Handle
ERROR_CM_INSUFFICIENT_MEMORY	-4	Failed, out of memory
ERROR_IO_OPEN_FAILED	-8	Failed to open port
ERROR_IO_USB_DEVICE_NOT_FOUND	-17	Unable to find USB device

## 5.4. ClosePort

This function is to close the communication port and disconnect with the printer.

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

**Parameter:**

*void\* handle*

[in] Printer object handle

**Return Value:**

Code	Value	Description
ERROR_CM_SUCCESS	0	Success
ERROR_CM_INVALID_HANDLE	-2	Invalid Handle

## 5.5. WriteData

This function is to send data to the printer.

```
int WriteData(void* handle,unsigned char* buffer,unsigned int size);
```

**Parameter:**

*void\* handle*

[in] Printer object handle

*unsigned char\* buffer*

[in] The data sent to the printer (hex string).

*unsigned int size*

[in] The length of the sent data.

**Return Value:**

Code	Value	Description
ERROR_CM_SUCCESS	0	Success
ERROR_CM_INVALID_PARAMETER	-1	Invalid argument
ERROR_CM_INVALID_HANDLE	-2	Invalid Handle
ERROR_CM_INSUFFICIENT_MEMORY	-4	Failed, out of memory
ERROR_IO_WRITE_FAILED	-9	Failed to send data
ERROR_IO_WRITE_TIMEOUT	-10	Write data timed out

## 5.6. ReadData

This function is to read the printer data.

```
int ReadData(void* handle,unsigned char* buffer,unsigned int size);
```

**Parameter:**

*void\* handle*

[in] Printer object handle

*unsigned char\* buffer*

[in] Printer data to be read.

*unsigned int size*

[in] The length of the data to be read.

**Return Value:**

Code	Value	Description
	>0	Success, read data length
ERROR_CM_INVALID_PARAMETER	-1	Invalid argument
ERROR_CM_INVALID_HANDLE	-2	Invalid Handle
ERROR_CM_INSUFFICIENT_MEMORY	-4	Failed, out of memory
ERROR_IO_OPEN_FAILED	-8	Failed to open port

## 5.7. TSPL\_SelfTest

This method is to print a self-test page for the printer, which contains basic configuration information for the printer.

int TSPL\_SelfTest(void\* hPrinter);

**Parameter:**

*void\* hPrinter*

[in] Printer object handle

**Return Value:**

Code	Value	Description
ERROR_CM_SUCCESS	0	Success
ERROR_CM_INVALID_PARAMETER	-1	Invalid argument
ERROR_CM_INVALID_HANDLE	-2	Invalid Handle
ERROR_CM_INSUFFICIENT_MEMORY	-4	Failed, out of memory
ERROR_IO_WRITE_FAILED	-9	Failed to send data
ERROR_IO_WRITE_TIMEOUT	-10	Write data timed out
Other values	Other values	The error code returned by the system

## 5.8. TSPL\_Bar

The function is to draw a bar.

```
int TSPL_Bar(void* hPrinter, int x, int y,int width, int height);
```

**Parameter:**

*void\* hPrinter*

[in] Printer object handle

*int x*

[in] Horizontal starting position, unit:dot.

*int y*

[in] Vertical starting position,unit:dot.

*int width*

[in] width, unit:dot.

*int height*

[in] height, unit:dot.

**Return Value:**

Code	Value	Description
ERROR_CM_SUCCESS	0	Success
ERROR_CM_INVALID_PARAMETER	-1	Invalid argument
ERROR_CM_INVALID_HANDLE	-2	Invalid Handle
ERROR_CM_INSUFFICIENT_MEMORY	-4	Failed, out of memory
ERROR_IO_WRITE_FAILED	-9	Failed to send data
ERROR_IO_WRITE_TIMEOUT	-10	Write data timed out
Other values	Other values	The error code returned by the system

## 5.9. TSPL\_BarCode

The method is to print a one-dimensional barcode.

```
int TSPL_BarCode(void* hPrinter,int x,int y,int type,const char* content,int height,int showText  
= 0,int rotation = 0,int narrow = 2,int wide = 2);
```

**Parameter:**

*void\* hPrinter*

[in] Printer object handle

*int x*

[in] Horizontal starting position, unit:dot.

*int y*

[in] Vertical starting position,unit:dot.

*int type*

[in] barcode type.

Type	Value
Code 128	0
Code 128M	1
EAN 128	2
Interleaved 2 of 5	3
Interleaved 2 of 5 with check digits	4
Code 39	5
Code 39C	6
Code 93	7
EAN 13	8
EAN 13 with 2 digits add-on	9
EAN 13 with 5 digits add-on	10
EAN 8	11
EAN 8 with 2 digits add-on	12
EAN 8 with 5 digits add-on	13
Codabar	14
Postnet	15
UPC-A	16
UPC-A with 2 digits add-on	17
UPC-A with 5 digits add-on	18
UPC- E	19
UPC- E with 2 digits add-on	20
UPC- E with 5 digits add-on	21
China post code	22
MSI code	23
MSI with check digit	24
PLESSEY code	25
ITF 14 code	26
EAN 14 code	27
Code 11	28
TELEPEN	29
TELEPENN	30
PLANET	31
CODE_49	32
DPI	33
DPL	34

*const char\* content*

[in] barcode data.

*int height*

[in] height, unit:dot.

*int showText*

[in] text display type.

Type	Value
do not display text	0



text is displayed on the left	1
text centered	2
text is displayed on the right	3

*int rotation*

[in] Rotation direction

Direction	Value
No rotation	0
Rotate 90 degrees	1
Rotate 180 degrees	2
Rotate 270 degrees	3

*int narrow*

[in] Narrow Bar Width, unit:dot.

*int rotation*

[in] Wide Bar Width, unit:dot.

**Return Value:**

Code	Value	Description
ERROR_CM_SUCCESS	0	Success
ERROR_CM_INVALID_PARAMETER	-1	Invalid argument
ERROR_CM_INVALID_HANDLE	-2	Invalid Handle
ERROR_CM_INSUFFICIENT_MEMORY	-4	Failed, out of memory
ERROR_IO_WRITE_FAILED	-9	Failed to send data
ERROR_IO_WRITE_TIMEOUT	-10	Write data timed out
Other values	Other values	The error code returned by the system

## 5.10. TSPL\_BitMap

This method is used to draw the binarized bitmap.

```
int TSPL_BitMap(void* hPrinter,int x,int y,int width,int height,int mode,unsigned char* data);
```

**Parameter:**

*void\* hPrinter*

[in] Printer object handle

*int x*

[in] Horizontal starting position, unit:dot.

*int y*

[in] Vertical starting position,unit:dot.

*int width*

[in] width, unit:dot.

*int height*

[in] height, unit:dot.

*int mode*

[in] Mode for printing bitmap.

Mode	Value
OVERWRITE	0
OR	1
XOR	2

*unsigned char\* data*

[in] Binarized bitmap data

#### Return Value:

Code	Value	Description
ERROR_CM_SUCCESS	0	Success
ERROR_CM_INVALID_PARAMETER	-1	Invalid argument
ERROR_CM_INVALID_HANDLE	-2	Invalid Handle
ERROR_CM_INSUFFICIENT_MEMORY	-4	Failed, out of memory
ERROR_IO_WRITE_FAILED	-9	Failed to send data
ERROR_IO_WRITE_TIMEOUT	-10	Write data timed out
Other values	Other values	The error code returned by the system

## 5.11. TSPL\_Image

This method is for printing the specified image (supports bmp, jpg, gif, etc), this function is suitable for printing LOGO.

Convert the image of the specified path to bitmap data and send it to the printer and print.

```
int TSPL_Image(void* hPrinter, int x, int y, int mode, const char* imgPath);
```

```
int TSPL_ImageW(void* hPrinter, int x, int y, int mode, const wchar_t* imgPathW);
```

#### Parameter:

*void\* hPrinter*

[in] Printer object handle

*int x*

[in] Horizontal starting position, unit:dot.

*int y*

[in] Vertical starting position,unit:dot.

*int mode*

[in] Mode for printing bitmap.

Mode	Value
OVERWRITE	0
OR	1
XOR	2

*const char\* imgPath*

*const wchar\_t\* imgPathW*

[in] The correct path to the image.

#### Return Value:

Code	Value	Description
ERROR_CM_SUCCESS	0	Success
ERROR_CM_INVALID_PARAMETER	-1	Invalid argument
ERROR_CM_INVALID_HANDLE	-2	Invalid Handle
ERROR_CM_INSUFFICIENT_MEMORY	-4	Failed, out of memory
ERROR_IO_WRITE_FAILED	-9	Failed to send data
ERROR_IO_WRITE_TIMEOUT	-10	Write data timed out
Other values	Other values	The error code returned by the system

## 5.12. TSPL\_Setup

This method is to set for the printer's basic parameters.

*int* TSPL\_Setup(*void\** hPrinter, *int* printSpeed, *int* printDensity, *int* labelWidth, *int* labelHeight, *int* labelType, *int* gapHeight, *int* offset);

#### Parameter:

*void\* hPrinter*

[in] Printer object handle

*int printSpeed*

[in] printing speed.

*int printDensity*

[in] print density (range: 0- 15).

*int labelWidth*

[in] Label width, unit:mm.

*int labelHeight*

[in] Label length, unit:mm.

*int labelType*

[in] label paper type (0: black mark/continuous paper 1: seam label/continuous paper).

*int gapHeight*

[in] The space between two labels, if it is set to 0, it means that it is continuous paper,unit:mm.  
*int offset*  
[in] offset position,unit:mm.

**Return Value:**

Code	Value	Description
ERROR_CM_SUCCESS	0	Success
ERROR_CM_INVALID_PARAMETER	-1	Invalid argument
ERROR_CM_INVALID_HANDLE	-2	Invalid Handle
ERROR_CM_INSUFFICIENT_MEMORY	-4	Failed, out of memory
ERROR_IO_WRITE_FAILED	-9	Failed to send data
ERROR_IO_WRITE_TIMEOUT	-10	Write data timed out
Other values	Other values	The error code returned by the system

## 5.13. TSPL\_ClearBuffer

This function is to clear the printer memory cache. Clear the printer cache before executing print data.

```
int TSPL_ClearBuffer(void* hPrinter);
```

**Parameter:**

*void\* hPrinter*  
[in] Printer object handle

**Return Value:**

Code	Value	Description
ERROR_CM_SUCCESS	0	Success
ERROR_CM_INVALID_PARAMETER	-1	Invalid argument
ERROR_CM_INVALID_HANDLE	-2	Invalid Handle
ERROR_CM_INSUFFICIENT_MEMORY	-4	Failed, out of memory
ERROR_IO_WRITE_FAILED	-9	Failed to send data
ERROR_IO_WRITE_TIMEOUT	-10	Write data timed out
Other values	Other values	The error code returned by the system

## 5.14. TSPL\_Box

The method is used to draw box.

```
int TSPL_Box(void* hPrinter, int x, int y, int x_end, int y_end, int thickness = 1, int radius = 0);
```

### Parameter:

*void\* hPrinter*

[in] Printer object handle

*int x*

[in] Horizontal starting position, unit:dot.

*int y*

[in] Vertical starting position,unit:dot.

*int x\_end*

[in] Horizontal end position, unit:dot.

*int y\_end*

[in] Vertical end position, unit:dot.

*int thickness*

[in] Line thickness., unit:dot.

*int radius*

[in] Specifies whether to have rounded corners, unit:dot.,default is 0.

### Return Value:

Code	Value	Description
ERROR_CM_SUCCESS	0	Success
ERROR_CM_INVALID_PARAMETER	-1	Invalid argument
ERROR_CM_INVALID_HANDLE	-2	Invalid Handle
ERROR_CM_INSUFFICIENT_MEMORY	-4	Failed, out of memory
ERROR_IO_WRITE_FAILED	-9	Failed to send data
ERROR_IO_WRITE_TIMEOUT	-10	Write data timed out
Other values	Other values	The error code returned by the system

## 5.15. TSPL\_QrCode

This method is used to print a QR code.

```
int TSPL_QrCode(void* hPrinter, int x, int y, int width, int eccLevel, int mode, int rotation, int model, int mask, const char* data);
```

### Parameter:

*void\* hPrinter*

[in] Printer object handle

*int x*

[in] Horizontal starting position, unit:dot.

*int y*

[in] Vertical starting position,unit:dot.

*int width*

[in] QR code print width (range: 1- 10).

*int eccLevel*

[in] Error correction level.

Error correction level.	Value
7%	0
15%	1
25%	2
30%	3

*int mode*

[in] Automatic / manual coding (0 : automatic, 1 : manual).

*int rotation*

[in] Rotation direction

Direction	Value
No rotation	0
Rotate 90 degrees	1
Rotate 180 degrees	2
Rotate 270 degrees	3

*int model*

[in] QR code version (0 : Basic, 1 : Enhanced).

*int mask*

[in] mask(range: 0-8).

*const char\* data*

[in] DataContent

**Return Value:**

Code	Value	Description
ERROR_CM_SUCCESS	0	Success
ERROR_CM_INVALID_PARAMETER	-1	Invalid argument
ERROR_CM_INVALID_HANDLE	-2	Invalid Handle
ERROR_CM_INSUFFICIENT_MEMORY	-4	Failed, out of memory
ERROR_IO_WRITE_FAILED	-9	Failed to send data
ERROR_IO_WRITE_TIMEOUT	-10	Write data timed out
Other values	Other values	The error code returned by the system

## 5.16. TSPL\_Text

This method is to print text for the printer.

```
int TSPL_Text(void* hPrinter, int x, int y, const char* fontName, const char* content, int rotation  
= 0, int x_multiplication = 1, int y_multiplication = 1, int alignment = 0);
```

### Parameter:

*void\* hPrinter*

[in] Printer object handle

*int x*

[in] Horizontal starting position, unit:dot.

*int y*

[in] Vertical starting position,unit:dot.

*const char\* fontName*

[in] Font name.

Description	Value
normal	"0"
8x12	"1"
12x20	"2"
16x24	"3"
24x32	"4"
32x48	"5"
14x19	"6"
21x27	"7"
14x25	"8"
Simplified Chinese	"TSS24.BF2"

*const char\* content*

[in] DataContent

*int rotation*

[in] Rotation direction

Direction	Value
No rotation	0
Rotate 90 degrees	1
Rotate 180 degrees	2
Rotate 270 degrees	3

*int x\_multiplication*

[in] Horizontal magnification ratio, effective parameters: 1~10.

*int y\_multiplication*

[in] Vertical magnification ratio, effective parameters: 1~10.

*int alignment*

[in] the alignment of the text.

Description	Value
Do not use this parameter	-1
Default (Left)	0
Align Left	1
Align Center	2
Align Right	3

**Return Value:**

Code	Value	Description
ERROR_CM_SUCCESS	0	Success
ERROR_CM_INVALID_PARAMETER	-1	Invalid argument
ERROR_CM_INVALID_HANDLE	-2	Invalid Handle
ERROR_CM_INSUFFICIENT_MEMORY	-4	Failed, out of memory
ERROR_IO_WRITE_FAILED	-9	Failed to send data
ERROR_IO_WRITE_TIMEOUT	-10	Write data timed out
Other values	Other values	The error code returned by the system

## 5.17. TSPL\_Print

This method is to perform printing operations.

```
int TSPL_Print(void* hPrinter, int num, int copies);
```

**Parameter:**

*void\* hPrinter*

[in] Printer object handle

*int num*

[in] The number of labels.

*int copies*

[in] Print the number of copies of each label.

**Return Value:**

Code	Value	Description
ERROR_CM_SUCCESS	0	Success
ERROR_CM_INVALID_PARAMETER	-1	Invalid argument
ERROR_CM_INVALID_HANDLE	-2	Invalid Handle
ERROR_CM_INSUFFICIENT_MEMORY	-4	Failed, out of memory
ERROR_IO_WRITE_FAILED	-9	Failed to send data
ERROR_IO_WRITE_TIMEOUT	-10	Write data timed out



Other values	Other values	The error code returned by the system
--------------	--------------	---------------------------------------

## 5.18. TSPL\_FormFeed

The method is to push the paper to the starting position of the next label.

```
int TSPL_FormFeed(void* hPrinter);
```

**Parameter:**

*void\* hPrinter*

[in] Printer object handle

**Return Value:**

Code	Value	Description
ERROR_CM_SUCCESS	0	Success
ERROR_CM_INVALID_PARAMETER	-1	Invalid argument
ERROR_CM_INVALID_HANDLE	-2	Invalid Handle
ERROR_CM_INSUFFICIENT_MEMORY	-4	Failed, out of memory
ERROR_IO_WRITE_FAILED	-9	Failed to send data
ERROR_IO_WRITE_TIMEOUT	-10	Write data timed out
Other values	Other values	The error code returned by the system

## 5.19. TSPL\_SetTear

This function is to enable/disable the tear-off function for setting the printer.

```
int TSPL_SetTear(void* hPrinter, int mode);
```

**Parameter:**

*void\* hPrinter*

[in] Printer object handle

*int mode*

[in] Enable/disable the tear-off function (0 : OFF, 1 : ON).

**Return Value:**

Code	Value	Description
ERROR_CM_SUCCESS	0	Success
ERROR_CM_INVALID_PARAMETER	-1	Invalid argument
ERROR_CM_INVALID_HANDLE	-2	Invalid Handle

ERROR_CM_INSUFFICIENT_MEMORY	-4	Failed, out of memory
ERROR_IO_WRITE_FAILED	-9	Failed to send data
ERROR_IO_WRITE_TIMEOUT	-10	Write data timed out
Other values	Other values	The error code returned by the system

## 5.20. TSPL\_SetRibbon

The method is used to set the ribbon working mode.

```
int TSPL_SetRibbon(void* hPrinter, int mode);
```

**Parameter:**

*void\* hPrinter*

[in] Printer object handle

*int mode*

[in] Ribbon mode; 0 = turn off the ribbon state; 1 = turn on the ribbon state.

**Return Value:**

Code	Value	Description
ERROR_CM_SUCCESS	0	Success
ERROR_CM_INVALID_PARAMETER	-1	Invalid argument
ERROR_CM_INVALID_HANDLE	-2	Invalid Handle
ERROR_CM_INSUFFICIENT_MEMORY	-4	Failed, out of memory
ERROR_IO_WRITE_FAILED	-9	Failed to send data
ERROR_IO_WRITE_TIMEOUT	-10	Write data timed out
Other values	Other values	The error code returned by the system

## 5.21. TSPL\_GetRibbonState

This function is to obtain the current printer ribbon status

```
int TSPL_GetRibbonState(void* hPrinter, int* mode);
```

**Parameter:**

*void\* hPrinter*

[in] Printer object handle

*int\* mode*

[in,out] Ribbon mode; 0 = turn off the ribbon state; 1 = turn on the ribbon state.

**Return Value:**

Code	Value	Description
ERROR_CM_SUCCESS	0	Success
ERROR_CM_INVALID_PARAMETER	-1	Invalid argument
ERROR_CM_INVALID_HANDLE	-2	Invalid Handle
ERROR_CM_INSUFFICIENT_MEMORY	-4	Failed, out of memory
ERROR_IO_WRITE_FAILED	-9	Failed to send data
ERROR_IO_WRITE_TIMEOUT	-10	Write data timed out
Other values	Other values	The error code returned by the system

## 5.22. TSPL\_Offset

This method is used to define the extra feed length of each label of the printer.

```
int TSPL_Offset(void* hPrinter, int distance);
```

**Parameter:**

*void\* hPrinter*

[in] Printer object handle

*int distance*

[in] Extra feed length, unit:dot.

**Return Value:**

Code	Value	Description
ERROR_CM_SUCCESS	0	Success
ERROR_CM_INVALID_PARAMETER	-1	Invalid argument
ERROR_CM_INVALID_HANDLE	-2	Invalid Handle
ERROR_CM_INSUFFICIENT_MEMORY	-4	Failed, out of memory
ERROR_IO_WRITE_FAILED	-9	Failed to send data
ERROR_IO_WRITE_TIMEOUT	-10	Write data timed out
Other values	Other values	The error code returned by the system

## 5.23. TSPL\_Direction

This method is used to set the printer printing direction.

```
int TSPL_Direction(void* hPrinter, int direction, int mirror);
```

**Parameter:**

*void\* hPrinter*

[in] Printer object handle

*int direction*

[in] Define the print orientation of the printer(0:Forward printing,1:reverse print).

*int mirror*

[in] Define the mirror of the printed content, 0:normal; 1:mirror

**Return Value:**

Code	Value	Description
ERROR_CM_SUCCESS	0	Success
ERROR_CM_INVALID_PARAMETER	-1	Invalid argument
ERROR_CM_INVALID_HANDLE	-2	Invalid Handle
ERROR_CM_INSUFFICIENT_MEMORY	-4	Failed, out of memory
ERROR_IO_WRITE_FAILED	-9	Failed to send data
ERROR_IO_WRITE_TIMEOUT	-10	Write data timed out
Other values	Other values	The error code returned by the system

## 5.24. TSPL\_Feed

This method is used to feed a specified length of paper to the printer.

`int TSPL_Feed(void* hPrinter, int n);`

**Parameter:**

*void\* hPrinter*

[in] Printer object handle

*int n*

[in] Feed length Valid parameter:  $\pm 1 \sim 9999$ , When the number is negative, the paper will be ejected backward, unit:dot.

**Return Value:**

Code	Value	Description
ERROR_CM_SUCCESS	0	Success
ERROR_CM_INVALID_PARAMETER	-1	Invalid argument
ERROR_CM_INVALID_HANDLE	-2	Invalid Handle
ERROR_CM_INSUFFICIENT_MEMORY	-4	Failed, out of memory
ERROR_IO_WRITE_FAILED	-9	Failed to send data
ERROR_IO_WRITE_TIMEOUT	-10	Write data timed out
Other values	Other values	The error code returned by

		the system
--	--	------------

## 5.25. TSPL\_Home

This method is used to feed the paper to the starting position for the printer.

```
int TSPL_Home(void* hPrinter);
```

**Parameter:**

*void\* hPrinter*

[in] Printer object handle

**Return Value:**

Code	Value	Description
ERROR_CM_SUCCESS	0	Success
ERROR_CM_INVALID_PARAMETER	-1	Invalid argument
ERROR_CM_INVALID_HANDLE	-2	Invalid Handle
ERROR_CM_INSUFFICIENT_MEMORY	-4	Failed, out of memory
ERROR_IO_WRITE_FAILED	-9	Failed to send data
ERROR_IO_WRITE_TIMEOUT	-10	Write data timed out
Other values	Other values	The error code returned by the system

## 5.26. TSPL\_Learn

This method is used to learn labels for the printer.

```
int TSPL_Learn(void* hPrinter);
```

**Parameter:**

*void\* hPrinter*

[in] Printer object handle

**Return Value:**

Code	Value	Description
ERROR_CM_SUCCESS	0	Success
ERROR_CM_INVALID_PARAMETER	-1	Invalid argument
ERROR_CM_INVALID_HANDLE	-2	Invalid Handle
ERROR_CM_INSUFFICIENT_MEMORY	-4	Failed, out of memory
ERROR_IO_WRITE_FAILED	-9	Failed to send data
ERROR_IO_WRITE_TIMEOUT	-10	Write data timed out

Other values	Other values	The error code returned by the system
--------------	--------------	---------------------------------------

## 5.27. TSPL\_GetSN

This method is used to obtain the serial number of the printer SN.

```
int TSPL_GetSN(void* hPrinter, char* snBuffer, int bufSize);
```

**Parameter:**

*void\* hPrinter*

[in] Printer object handle

*char\* snBuffer*

[in,out] The buffer address used to get the serial number.

*int bufSize*

[in] Buffer size.

**Return Value:**

Code	Value	Description
ERROR_CM_SUCCESS	0	Success
ERROR_CM_INVALID_PARAMETER	-1	Invalid argument
ERROR_CM_INVALID_HANDLE	-2	Invalid Handle
ERROR_CM_INSUFFICIENT_MEMORY	-4	Failed, out of memory
ERROR_IO_WRITE_FAILED	-9	Failed to send data
ERROR_IO_WRITE_TIMEOUT	-10	Write data timed out
Other values	Other values	The error code returned by the system

## 5.28. TSPL\_GetPrinterStatus

This method is used to get the current state of the printer.

```
int TSPL_GetPrinterStatus(void* hPrinter, unsigned int* printerStatus);
```

**Parameter:**

*void\* hPrinter*

[in] Printer object handle

*unsigned int\* printerStatus*

[in,out] Current status of the printer.

Status	Value
--------	-------

Normal	0
The print head is opened	1
Paper jam	2
Out of paper	4
Out of ribbon	8
Print pause	16
Printing	32
Cover opened	64
Other error	128

#### Return Value:

Code	Value	Description
ERROR_CM_SUCCESS	0	Success
ERROR_CM_INVALID_PARAMETER	-1	Invalid argument
ERROR_CM_INVALID_HANDLE	-2	Invalid Handle
ERROR_CM_INSUFFICIENT_MEMORY	-4	Failed, out of memory
ERROR_IO_WRITE_FAILED	-9	Failed to send data
ERROR_IO_WRITE_TIMEOUT	-10	Write data timed out
Other values	Other values	The error code returned by the system

## 5.29. TSPL\_SetCodePage

This method is used to set the code page for the printer.

```
int TSPL_SetCodePage(void* hPrinter, const char* codepage);
```

#### Parameter:

*void\* hPrinter*

[in] Printer object handle

*const char\* codepage*

[in] Code page.

Code	Name	Code	Name	Code	Name	Code	Name
USA	USA	437	United States	1250	Central Europe	8859-1	Latin 1
BRI	British	737	Greek	1251	Cyrillic	8859-2	8859-2
GER	German	850	Multilingual	1252	Latin I	8859-3	Latin 3
FRE	French	851	Greek 1	1253	Greek	8859-4	Baltic
DAN	Danish	852	Slavic	1254	Turkish	8859-5	Cyrillic
ITA	Italian	855	Cyrillic	1255	Hebrew	8859-6	Arabic
SPA	Spanish	857	Turkish	1256	Arabic	8859-7	Greek
SWE	Swedish	860	Portuguese	1257	Baltic	8859-8	Hebrew

SWI	Swiss	861	Icelandic	1258	Vietnam	8859-9	Turkish
		862	Hebrew	932	Japanese Shift-JIS	8859-10	Latin 6
		863	Canadian/French	936	Simplified Chinese GBK	8859-15	Latin 9
		864	Arabic	949	Korean		
		865	Nordic	950	Traditional Chinese Big5		
		866	Russian	UTF-8	UTF 8		
		869	Greek 2				

#### Return Value:

Code	Value	Description
ERROR_CM_SUCCESS	0	Success
ERROR_CM_INVALID_PARAMETER	-1	Invalid argument
ERROR_CM_INVALID_HANDLE	-2	Invalid Handle
ERROR_CM_INSUFFICIENT_MEMORY	-4	Failed, out of memory
ERROR_IO_WRITE_FAILED	-9	Failed to send data
ERROR_IO_WRITE_TIMEOUT	-10	Write data timed out
Other values	Other values	The error code returned by the system

## 5.30. TSPL\_PDF417

This method is used to print PDF417 QR code.

```
int TSPL_PDF417(void* hPrinter, int x, int y, int width, int height, int rotation, const char* option, const char* data);
```

#### Parameter:

*void\* hPrinter*

[in] Printer object handle

*int x*

[in] Horizontal starting position, unit:dot.

*int y*

[in] Vertical starting position,unit:dot.

*int width*

[in] width, unit:dot.

*int height*

[in] height, unit:dot.



*int rotation*

[in] Rotation direction

Direction	Value
No rotation	0
Rotate 90 degrees	1
Rotate 180 degrees	2
Rotate 270 degrees	3

*const char\* option*

[in] Options (Example: E3,W2,H8).

Value	Description
P	Data compression method: 0 : automatic 1 : binary mode
E	Error check level (0~8)
M	Bar code center print mode 0: This mode will print in the upper left corner alignment area 1: will print in the middle area
Ux,y,c	Code readable x: the x coordinate specified by the readable character y: the y coordinate specified by the readable character c: maximum number of readable characters per line
W	Module width (2~9: dot)
H	Height of small bar code (4~99: dot)
R	Maximum number of rows
C	Maximum number of columns
T	Whether to cut off 0: No 1: Yes
Lm	Indicates the length (1~2048)

*const char\* data*

[in] DataContent

**Return Value:**

Code	Value	Description
ERROR_CM_SUCCESS	0	Success
ERROR_CM_INVALID_PARAMETER	-1	Invalid argument
ERROR_CM_INVALID_HANDLE	-2	Invalid Handle
ERROR_CM_INSUFFICIENT_MEMORY	-4	Failed, out of memory
ERROR_IO_WRITE_FAILED	-9	Failed to send data

ERROR_IO_WRITE_TIMEOUT	-10	Write data timed out
Other values	Other values	The error code returned by the system

## 5.31. TSPL\_Block

This method is used to prints paragraph on label.

```
int TSPL_Block(void* hPrinter, int x, int y, int width, int height,const char* fontName,const char* content,int rotation = 0,int x_multiplication = 1,int y_multiplication = 1,int alignment = 0);
```

### Parameter:

*void\* hPrinter*

[in] Printer object handle

*int x*

[in] Horizontal starting position, unit:dot.

*int y*

[in] Vertical starting position,unit:dot.

*int width*

[in] The width of block for the paragraph in dots.

*int height*

[in] The height of block for the paragraph in dots.

*const char\* fontName*

[in] Font name.

Description	Value
normal	"0"
8x12	"1"
12x20	"2"
16x24	"3"
24x32	"4"
32x48	"5"
14x19	"6"
21x27	"7"
14x25	"8"
Simplified Chinese	"TSS24.BF2"

*const char\* content*

[in] DataContent

*int rotation*

[in] Rotation direction

Direction	Value
No rotation	0

Rotate 90 degrees	1
Rotate 180 degrees	2
Rotate 270 degrees	3

*int x\_multiplication*

[in] Horizontal magnification ratio, effective parameters: 1~10.

*int y\_multiplication*

[in] Vertical magnification ratio, effective parameters: 1~10.

*int alignment*

[in] the alignment of the text.

the alignment of the text.	Value
Default (Left)	0
Align Left	1
Align Center	2
Align Right	3

**Return Value:**

Code	Value	Description
ERROR_CM_SUCCESS	0	Success
ERROR_CM_INVALID_PARAMETER	-1	Invalid argument
ERROR_CM_INVALID_HANDLE	-2	Invalid Handle
ERROR_CM_INSUFFICIENT_MEMORY	-4	Failed, out of memory
ERROR_IO_WRITE_FAILED	-9	Failed to send data
ERROR_IO_WRITE_TIMEOUT	-10	Write data timed out
Other values	Other values	The error code returned by the system

## 5.32. TSPL\_Reverse

This method is used to reverses a region in image buffer.

`int TSPL_Reverse(void* hPrinter,int x,int y,int width,int height);`

**Parameter:**

*void\* hPrinter*

[in] Printer object handle

*int x*

[in] Horizontal starting position, unit:dot.

*int y*

[in] Vertical starting position,unit:dot.

*int width*

[in] width, unit:dot.

*int height*

[in] height, unit:dot.

**Return Value:**

Code	Value	Description
ERROR_CM_SUCCESS	0	Success
ERROR_CM_INVALID_PARAMETER	-1	Invalid argument
ERROR_CM_INVALID_HANDLE	-2	Invalid Handle
ERROR_CM_INSUFFICIENT_MEMORY	-4	Failed, out of memory
ERROR_IO_WRITE_FAILED	-9	Failed to send data
ERROR_IO_WRITE_TIMEOUT	-10	Write data timed out
Other values	Other values	The error code returned by the system

### 5.33. TSPL\_GapDetect

This method is used to detect paper size and gap size.

```
int TSPL_GapDetect(void* hPrinter, int x = 0, int y = 0);
```

**Parameter:**

*void\* hPrinter*

[in] Printer object handle

*int x*

[in] Paper length (in dots).

*int y*

[in] Gap length (in dots).

**Return Value:**

Code	Value	Description
ERROR_CM_SUCCESS	0	Success
ERROR_CM_INVALID_PARAMETER	-1	Invalid argument
ERROR_CM_INVALID_HANDLE	-2	Invalid Handle
ERROR_CM_INSUFFICIENT_MEMORY	-4	Failed, out of memory
ERROR_IO_WRITE_FAILED	-9	Failed to send data
ERROR_IO_WRITE_TIMEOUT	-10	Write data timed out
Other values	Other values	The error code returned by the system

## 5.34. TSPL\_Dmatrix

This method is used to print Data Matrix 2D barcodes.

```
int TSPL_Dmatrix(void* hPrinter, int x, int y, int width, int height, const char* content, int  
blockSize = 0, int row = 10, int col = 10);
```

### Parameter:

*void\* hPrinter*

[in] Printer object handle

*int x*

[in] Horizontal starting position, unit:dot.

*int y*

[in] Vertical starting position,unit:dot.

*int width*

[in] width, unit:dot.

*int height*

[in] height, unit:dot.

*const char\* content*

[in] DataContent

*int blockSize*

[in] optional,Module size (in dots).

*int row*

[in] optional,Symbol size of row: 10 to 144.

*int col*

[in] optional,Symbol size of col: 10 to 144.

### Return Value:

Code	Value	Description
ERROR_CM_SUCCESS	0	Success
ERROR_CM_INVALID_PARAMETER	-1	Invalid argument
ERROR_CM_INVALID_HANDLE	-2	Invalid Handle
ERROR_CM_INSUFFICIENT_MEMORY	-4	Failed, out of memory
ERROR_IO_WRITE_FAILED	-9	Failed to send data
ERROR_IO_WRITE_TIMEOUT	-10	Write data timed out
Other values	Other values	The error code returned by the system

## 5.35. TSPL\_SetCutter

This method is used to set the cutter working mode.

```
int TSPL_SetCutter(void* hPrinter, int mode);
```

### Parameter:

*void\* hPrinter*

[in] Printer object handle

*int mode*

[in] Set number of printing labels per cut.0=Turn off the cutter function,-1=

Cut paper after printing job, 1-65535=Number of labels for cut paper

### Return Value:

Code	Value	Description
ERROR_CM_SUCCESS	0	Success
ERROR_CM_INVALID_PARAMETER	-1	Invalid argument
ERROR_CM_INVALID_HANDLE	-2	Invalid Handle
ERROR_CM_INSUFFICIENT_MEMORY	-4	Failed, out of memory
ERROR_IO_WRITE_FAILED	-9	Failed to send data
ERROR_IO_WRITE_TIMEOUT	-10	Write data timed out
Other values	Other values	The error code returned by the system

## 5.36. TSPL\_SetGPO

This method is used to set the GPIO signals by the printer.

```
int TSPL_SetGPO(void* hPrinter, int output_number, int state, int Delay0_ms, int Pulse0_ms, int Delay1_ms, int Pulse1_ms, int gpo_fc );
```

### Parameter:

*void\* hPrinter*

[in] Printer object handle

*int output\_number*

[in] Output number: 1 to 7.

*int state*

[in] Signal state

Signal state	Value
LOW	0
HIGH	1

NEG	2
POS	3

*int Delay0\_ms*

[in] Delay time: 0 - 32000 millisecond

*int Pulse0\_ms*

[in] Pulse width: 0 - 32000 millisecond

*int Delay1\_ms*

[in] Delay time: 0 - 32000 millisecond

*int Pulse1\_ms*

[in] Pulse width: 0 - 32000 millisecond

*int gpo\_fc*

[in] GPO Function condition

GPO Function condition	Value
Printer fault.	0
Ribbon error is occurred.	1
Paper empty or paper jam is occurred.	2
Carriage is open.	3
Out of memory is occurred.	4
Cutter error is occurred.	5
Stepping motor or print head is over heat.	6
Pause status is occurred.	7
Take label is occurred.	8
Printer is idle.	9
Printer is printing.	10

**Return Value:**

Code	Value	Description
ERROR_CM_SUCCESS	0	Success
ERROR_CM_INVALID_PARAMETER	-1	Invalid argument
ERROR_CM_INVALID_HANDLE	-2	Invalid Handle
ERROR_CM_INSUFFICIENT_MEMORY	-4	Failed, out of memory
ERROR_IO_WRITE_FAILED	-9	Failed to send data
ERROR_IO_WRITE_TIMEOUT	-10	Write data timed out
Other values	Other values	The error code returned by the system

## 5.37. TSPL\_SetGPI

This method is used receive the GPIO signals from external controlling devices.

int TSPL\_SetGPI(void\* hPrinter, int input\_number, int state, int Pulse\_ms, int gpi\_fc, int fc\_number );

**Parameter:**

*void\* hPrinter*

[in] Printer object handle

*int gpi\_number*

[in] Input Number: 1 - 4

*int state*

[in] SetGPIStateParam

Signal state	Value
LOW	0
HIGH	1
NEG	2
POS	3

*int Pulse\_ms*

[in] Pulse width: 0 - 32000 millisecond

*int gpi\_fc*

[in] GPI Function Control

GPI Function Control	Value
Toggle pause status.	0
Enter pause status.	1
Cancel pause status.	2
fc_number is numerical. Specify how many labels to print.	3
Cut immediately.	4
fc_number is numerical and the unit is dot. Specify the feeding length.	5
fc_number is numerical and the unit is dot. Specify the backfeeding length.	6
Feeding an empty label.	7
Back label to the printing position.	8

*int fc\_number*

[in] GPI Function Control params: 1 - 32000 pages/dot

**Return Value:**

Code	Value	Description
ERROR_CM_SUCCESS	0	Success
ERROR_CM_INVALID_PARAMETER	-1	Invalid argument
ERROR_CM_INVALID_HANDLE	-2	Invalid Handle
ERROR_CM_INSUFFICIENT_MEMORY	-4	Failed, out of memory
ERROR_IO_WRITE_FAILED	-9	Failed to send data



ERROR_IO_WRITE_TIMEOUT	-10	Write data timed out
Other values	Other values	The error code returned by the system

## 5.38. TSPL\_SetBlineSensor

This function is used to set the black label sensor and is only applicable to non HQ printers

```
int TSPL_SetBlineSensor(void* hPrinter, int isOpen)
```

### Parameter:

*void\* hPrinter*

[in] Printer object handle

*int isOpen*

[in] Whether to turn on the black mark sensor

Description	Value
Open	1
Close	0

### Return Value:

Code	Value	Description
ERROR_CM_SUCCESS	0	Success
ERROR_CM_INVALID_PARAMETER	-1	Invalid argument
ERROR_CM_INVALID_HANDLE	-2	Invalid Handle
ERROR_CM_INSUFFICIENT_MEMORY	-4	Failed, out of memory
ERROR_IO_WRITE_FAILED	-9	Failed to send data
ERROR_IO_WRITE_TIMEOUT	-10	Write data timed out
Other values	Other values	The error code returned by the system