

[Printer SDK for Windows]

[打印机开发手册 v2.1.5] – ESC/POS指令集

目录

[Printer SDK for Windows]	1
1. 手册信息	3
2. 操作系统	3
3. 备注	3
4. NuGet仓库配置	3
5. 接口	3
5.1. PrinterCreator	3
5.2. ReleasePrinter	4
5.3. OpenPort	4
5.4. ClosePort	5
5.5. WriteData	5
5.6. ReadData	6
5.7. PrinterInitialize	6
5.8. SetTextLineSpace	7
5.9. CancelPrintDataInPageMode	7
5.10. GetPrinterState	8
5.11. SetCodePage	9
5.12. SetInternationalCharacter	10
5.13. CutPaper/CutPaperWithDistance	11
5.14. FeedLine	11
5.15. PrintAndFeedLine	12
5.16. OpenCashDrawer	12
5.17. PrintText	13
5.18. SetRelativeHorizontal	14
5.19. PrintTextS	14
5.20. PrintBarCode	15
5.21. PrintSymbol	16
5.22. PrintImage	17
5.23. DefineNVImageCompatible	18
5.24. PrintNVImageCompatible	19
5.25. SelectPageMode	19
5.26. SelectStandardMode	20
5.27. SelectPrintDirectionInPageMode	20
5.28. SetAbsoluteVerticalPrintPositionInPageMode	21
5.29. PrintAndReturnStandardMode	21
5.30. SetPrintAreaInPageMode	21
5.31. PrintDataInPageMode	22
5.32. SetAbsolutePrintPosition	23
5.33. PositionNextLabel	23
5.34. SetAlign	24
5.35. SetTextBold	24
5.36. SetTextFont	25
5.37. SetHorizontalAndVerticalMotionUnits	25
5.38. EnableBlackMark	26
5.39. SetBlackMarkDistance	26
5.40. SetBlackMarkHeight	27
5.41. DefineUserDefinedCharacters	27
5.42. DeleteUserDefinedCharacter	28
5.43. GoHomeWithBlackMark	28
5.44. SetBlackMarkAdjust	29
5.45. FirmwareUpgrade	29
5.46. SearchEscNetDevice	30
5.47. SetEscNetInfo	31

1. 手册信息

本手册描述了我公司支持ESC/POS指令的打印机开发接口文档。
由于我们在持续提升产品的功能和质量，本手册描述的产品规格和接口内容可能会更改，将不再另行通知。

2. 操作系统

Windows 10以上

3. 备注

错误代码返回值大于0时，属于 Windows 系统内部错误，请查阅相关帮助文档。

4. NuGet仓库配置

Visual Studio可以使用nuget仓库
仓库地址:<http://47.245.124.98:8081/repository/nuget-hosted/>
用户和密码:您可以通过咨询客服免费获取账号和密码

5. 接口

5.1. PrinterCreator

此函数功能为创建指定机型的打印机对象(在进行任何打印机操作之前必须先创建打印机对象)。

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

参数：
*void** handle*
[in,out] 创建目标打印机对象。
const TCHAR model*
[in] 指定目标打印机型号。

返回值：

错误代码	值	描述
ERROR_CM_SUCCESS	0	成功
ERROR_CM_INVALID_PARAMETER	-1	参数无效

5.2. ReleasePrinter

此函数功能为释放已创建指定机型打印机对象的资源(在操作结束后且不再进行 打印机操作时必须释放创建的打印机对象)。

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

参数：
void* hPrinter
[in] 需要释放的目标打印机对象的句柄。

返回值：

错误代码	值	描述
ERROR_CM_SUCCESS	0	成功
ERROR_CM_INVALID_HANDLE	-2	失败，句柄无效
ERROR_CM_INVALID_PARAMETER	-1	参数无效
ERROR_CM_INSUFFICIENT_MEMORY	-4	申请内存失败

5.3. OpenPort

此函数功能为打开通讯端口，与打印建立连接。连接成功后才能正常使用其它功能。连接失败时，请查看函数返回的错误信息。目前支持 USB。

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

参数：
void* hPrinter
[in] 创建的目标打印机对象。

ioSettings
[in] 设置连接目标打印机的通讯端口参数。具体内容查看下表：
配置列表：

类别	配置	描述	示例
USB	USB,Model/PortNum	USB，打印机型号 USB，端口号 如果同时连接本公司多台不同型号的打印机,建议采用“USB,机型”的方式连接	USB,4B-2054A USB,USB031
NET	NET, IP 地址 (IPV4)[,端口]	指定网络打印机的IP地址和端口。如果不指定端口，默认端口是9100。	NET,192.168.1.10 NET,192.168.1.10,9100
COM	COMn,rate	指定连接的串口端口号和波特率。	COM10,19200
LPT	LPTn	指定连接的并口端口号	LPT3

返回值：

错误代码	值	描述
ERROR_CM_SUCCESS	0	成功
ERROR_CM_INVALID_HANDLE	-2	失败，句柄无效
ERROR_CM_INSUFFICIENT_MEMORY	-4	申请内存失败
ERROR_IO_USB_DEVICE_NOT_FOUND	-17	找不到USB设备
ERROR_IO_OPEN_FAILED	-8	打开IO失败
ERROR_CM_INVALID_PARAMETER	-1	参数无效

5.4. ClosePort

此函数功能为关闭通讯。当不使用端口通讯时，请关闭端口。

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

参数：

void* hPrinter
[in] 创建的目标打印机对象。

返回值：

错误代码	值	描述
ERROR_CM_SUCCESS	0	成功
ERROR_CM_INVALID_HANDLE	-2	失败，句柄无效
ERROR_CM_INSUFFICIENT_MEMORY	-4	申请内存失败

5.5. WriteData

此函数功能为向打印机发送数据。

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

参数：

void* handle
[in] 打印机对象句柄。

unsigned char* buffer
[in] 发给打印机的数据，数据是十六进制字符串。

unsigned int size
[in] 发送数据的长度。

返回值：

错误代码	值	描述
ERROR_CM_SUCCESS	0	成功
ERROR_CM_INVALID_HANDLE	-2	失败，句柄无效
ERROR_CM_INVALID_PARAMETER	-1	参数无效
ERROR_CM_INSUFFICIENT_MEMORY	-4	申请内存失败
ERROR_IO_WRITE_FAILED	-9	写入数据失败
ERROR_IO_WRITE_TIMEOUT	-10	写入数据超时

5.6. ReadData

此函数功能为读取打印机的数据。

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

参数：

void handle*

[in] 打印机对象句柄。

unsigned char buffer*

[in] 需要读取的打印机数据。

unsigned int size

[in] 所需读取的数据长度。

返回值：

错误代码	值	描述
	>0	成功，读取的数据长度
ERROR_CM_INVALID_HANDLE	-2	失败，句柄无效
ERROR_CM_INVALID_PARAMETER	-1	参数无效
ERROR_CM_INSUFFICIENT_MEMORY	-4	申请内存失败
ERROR_IO_OPEN_FAILED	-8	打开IO失败

5.7. PrinterInitialize

此函数功能为清除打印缓冲数据并重置打印机模式为开机有效模式。不清除任何宏函数。

不清除脱机回应选择。

不清除用户 NV 存储内容。

不清除 NV 图像 (NV 位图) 和 NV 用户内存。

本指令不影响维护对应值。

不清除指定的脱机回应。

```
int PrinterInitialize(
```

```
void* hPrinter
```

```
);
```

参数：

void hPrinter*

[in] 创建的目标打印机对象。

返回值：

错误代码	值	描述
ERROR_CM_SUCCESS	0	成功
ERROR_CM_INVALID_HANDLE	-2	失败，句柄无效
ERROR_CM_INVALID_PARAMETER	-1	参数无效
ERROR_CM_INSUFFICIENT_MEMORY	-4	申请内存失败
ERROR_IO_WRITE_FAILED	-9	写入数据失败
ERROR_IO_WRITE_TIMEOUT	-10	写入数据超时

5.8. SetTextLineSpace

此函数功能为设置行间距为行间距 × (垂直或水平移动单元)。
选择标准模式时，行间距使用的是垂直移动单元。
选择页模式时，根据打印方向确认使用垂直或水平移动单元。当打印方向为从左至右时，行间距使用的是垂直移动单元；当打印方向为从上至下时，行间距使用的是水平移动单元。

```
int SetTextLineSpace(  
    void* hPrinter,  
    int lineSpace  
);
```

参数：
void* hPrinter
[in] 创建的目标打印机对象。
int lineSpace
[in] 设置字符行间距 0≤行间距≤255。

返回值：

错误代码	值	描述
ERROR_CM_SUCCESS	0	成功
ERROR_CM_INVALID_HANDLE	-2	失败，句柄无效
ERROR_CM_INVALID_PARAMETER	-1	参数无效
ERROR_CM_INSUFFICIENT_MEMORY	-4	申请内存失败
ERROR_IO_WRITE_FAILED	-9	写入数据失败
ERROR_IO_WRITE_TIMEOUT	-10	写入数据超时

5.9. CancelPrintDataInPageMode

此函数功能为在页模式下，清除当前打印区域的所有打印数据。

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

参数：
void* hPrinter
[in,out] 创建的目标打印机对象。

返回值：

错误代码	值	描述
ERROR_CM_SUCCESS	0	成功
ERROR_CM_INVALID_HANDLE	-2	失败，句柄无效
ERROR_CM_INVALID_PARAMETER	-1	参数无效
ERROR_CM_INSUFFICIENT_MEMORY	-4	申请内存失败
ERROR_IO_WRITE_FAILED	-9	写入数据失败
ERROR_IO_WRITE_TIMEOUT	-10	写入数据超时

5.10. GetPrinterState

此函数功能为获取打印机实时状态操作。

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

参数：

void* hPrinter

[in] 创建的目标打印机对象。

unsigned int * printerStatus

[in,out] 输入值范围：1-4,返回状态参照以下表格。

printerStatus	位	功能	值	十进制值
1	0	固定为0	0	0
	1	固定为1	1	2
	2	一个或两个钱箱打开	0	0
		两个钱箱都关闭	1	4
	3	联机	0	0
		脱机	1	8
	4	脱机	1	16
	5,6	固定为0	0	0
2	7	固定为0	0	0
	0	固定为0	0	0
	1	固定为1	1	2
	2	上盖关	0	0
		上盖开	1	4
	3	未按走纸键	0	0
		按下走纸键	1	8
	4	固定为1	1	16
	5	打印机不缺纸	0	0
		打印机缺纸	1	32
	6	无错误情况	0	0
		有错误情况	1	64
3	7	固定为0	0	0
	0	固定为0	0	0
	1	固定为1	1	2
	2	未定义	0	0
	3	切刀无错误	0	0
		切刀有错误	1	8
	4	固定为1	1	16
	5	无不可恢复错误	0	0
		有不可恢复错误	1	32
	6	打印头温度和电压正常	0	0
		打印头温度或电压超出范围	1	64
	7	固定为0	0	0
4	0	固定为0	0	0
	1	固定为1	1	2
	2,3	有纸	0	0
		纸将尽	1	12
	4	固定为1	1	16
	5,6	有纸	0	0

		纸尽	1	96
	7	固定0	0	0

返回值：

错误代码	值	描述
ERROR_CM_SUCCESS	0	成功
ERROR_CM_INVALID_HANDLE	-2	失败，句柄无效
ERROR_CM_INVALID_PARAMETER	-1	参数无效
ERROR_CM_INSUFFICIENT_MEMORY	-4	申请内存失败
ERROR_IO_WRITE_FAILED	-9	写入数据失败
ERROR_IO_WRITE_TIMEOUT	-10	写入数据超时
ERROR_IO_READ_FAILED	-11	读取数据失败

5.11. SetCodePage

此函数功能为设置字符集。

```
int SetCodePage(  
    void* hPrinter,  
    int characterSet  
);
```

参数：

void* hPrinter

[in] 创建的目标打印机对象。

int characterSet

[in] 选择字符集设置。

值	描述	值	描述
0	PC437(Std.Europe)	56	PC861(Icelandic)
1	Katakana	57	PC863(Canadian)
2	PC850(Multilingual)	58	PC865(Nordic)
3	PC860(Portugal)	59	PC866(Russian)
4	PC863(Canadian)	60	PC855(Bulgarian)
5	PC865(Nordic)	61	PC857(Turkey)
6	West Europe	62	PC862(Hebrew)
7	Greek	63	PC864(Arabic)
8	Hebrew	64	PC737(Greek)
9	East Europe	65	PC851(Greek)
10	Iran	66	PC869(Greek)
16	WPC1252	67	PC928(Greek)
17	PC866(Cyrillic#2)	68	PC772(Lithuanian)
18	PC852(Latin2)	69	PC774(Lithuanian)
19	PC858	70	PC874(Thai)
20	IranII	71	WPC1252(Latin-1)
21	Latvian	72	WPC1250(Latin-2)
22	Arabic	73	WPC1251(Cyrillic)
23	PT1511251	74	PC3840(IBM-Russian)
24	PC747	75	PC3841(Gost)
25	WPC1257	76	PC3843(Polish)
27	Vietnam	77	PC3844(CS2)
28	PC864	78	PC3845(Hungarian)
29	PC1001	79	PC3846(Turkish)
30	Uigur	80	PC3847(Brazil-ABNT)
31	Hebrew	81	PC3848(Brazil)

32	WPC1255(Israel)	82	PC1001(Arabic)
255	Thai	83	PC2001(Lithuan)
33	WPC1256	84	PC3001(Estonian-1)
50	PC437(Std.Europe)	85	PC3002(Eston-2)
51	Katakana	86	PC3011(Latvian-1)
52	PC437(Std.Europe)	87	PC3012(Tatv-2)
53	PC858(Multilingual)	88	PC3021(Bulgarian)
54	PC852(Latin-2)	89	PC3041(Maltese)
55	PC860(Portuguese)	255	[Thai]

返回值:

错误代码	值	描述
ERROR_CM_SUCCESS	0	成功
ERROR_CM_INVALID_HANDLE	-2	失败, 句柄无效
ERROR_CM_INVALID_PARAMETER	-1	参数无效
ERROR_CM_INSUFFICIENT_MEMORY	-4	申请内存失败
ERROR_IO_WRITE_FAILED	-9	写入数据失败
ERROR_IO_WRITE_TIMEOUT	-10	写入数据超时

5.12. SetInternationalCharacter

此函数功能为选择国际字符集。

```
int SetInternationalCharacter(
    void* hPrinter,
    int characterSet
);
```

参数:

void* hPrinter

[in] 创建的目标打印机对象。

int characterSet

[in] 国际字符集。

默认: U.S.A

值	描述
0	U.S.A
1	France
2	Germany
3	U K
4	Denmark I
5	Sweden
6	Italy
7	Spain
8	Japan
9	Norway
10	Denmark II
11	Spain II
12	Latin America
13	Korean
14	Slovenia / Croatia
15	Chinese

返回值:

错误代码	值	描述
ERROR_CM_SUCCESS	0	成功
ERROR_CM_INVALID_HANDLE	-2	失败，句柄无效
ERROR_CM_INVALID_PARAMETER	-1	参数无效
ERROR_CM_INSUFFICIENT_MEMORY	-4	申请内存失败
ERROR_IO_WRITE_FAILED	-9	写入数据失败
ERROR_IO_WRITE_TIMEOUT	-10	写入数据超时

5.13. CutPaper/CutPaperWithDistance

此函数功能为走纸至(裁切位置+ 间距 × 垂直移动单元)，执行全切(完全切纸)或者半切(留一点不切)，然后走纸至打印起始位置。

```
int CutPaper(  
    void* hPrinter,  
    int cutMode  
);  
int CutPaperWithDistance(  
    void* hPrinter,  
    int distance  
);
```

参数：

void* hPrinter

[in] 创建的目标打印机对象。

int cutMode

[in] 切纸模式，全切或者半切。

切纸模式	值	描述
FULL_CUT	0/48	全切
PARTIAL_CUT	1/49	半切

int distance

[in] 指定切纸范围 0≤间距≤255。

返回值：

错误代码	值	描述
ERROR_CM_SUCCESS	0	成功
ERROR_CM_INVALID_HANDLE	-2	失败，句柄无效
ERROR_CM_INVALID_PARAMETER	-1	参数无效
ERROR_CM_INSUFFICIENT_MEMORY	-4	申请内存失败
ERROR_IO_WRITE_FAILED	-9	写入数据失败
ERROR_IO_WRITE_TIMEOUT	-10	写入数据超时

5.14. FeedLine

此函数功能为打印打印缓冲区的数据并走纸，页模式下，只有打印位置移动，打印机实际上不执行打印。

```
int FeedLine(  
    void* hPrinter,  
    int lines  
);
```

参数：

void* hPrinter

[in] 创建的目标打印机对象。

int lines

[in] 设置走纸行数 $0 \leq \text{lines} \leq 255$ 。

返回值：

错误代码	值	描述
ERROR_CM_SUCCESS	0	成功
ERROR_CM_INVALID_HANDLE	-2	失败，句柄无效
ERROR_CM_INVALID_PARAMETER	-1	参数无效
ERROR_CM_INSUFFICIENT_MEMORY	-4	申请内存失败
ERROR_IO_WRITE_FAILED	-9	写入数据失败
ERROR_IO_WRITE_TIMEOUT	-10	写入数据超时

5.15. PrintAndFeedLine

此函数功能为打印打印缓冲区的数据并走纸一行。

```
int PrintAndFeedLine(  
    void* hPrinter  
)
```

参数：

void* hPrinter

[in] 创建的目标打印机对象。

返回值：

错误代码	值	描述
ERROR_CM_SUCCESS	0	成功
ERROR_CM_INVALID_HANDLE	-2	失败，句柄无效
ERROR_CM_INVALID_PARAMETER	-1	参数无效
ERROR_CM_INSUFFICIENT_MEMORY	-4	申请内存失败
ERROR_IO_WRITE_FAILED	-9	写入数据失败
ERROR_IO_WRITE_TIMEOUT	-10	写入数据超时

5.16. OpenCashDrawer

此函数功能为打开钱箱 (打印机必须连接钱箱)。

```
int OpenCashDrawer(  
    void* hPrinter,  
    int pinMode,  
    int onTime,  
    int ofTime  
);
```

参数：

void* hPrinter

[in] 创建的目标打印机对象。

int pinMode

[in] 选择钱箱连接的引脚。

引脚	值	描述
CASDRAWER_1	0	引脚 2
CASDRAWER_2	1	引脚 5

int onTime
[in] 设置脉冲开始时间，onTime*2毫秒。

int ofTime
[in] 设置脉冲结束时间，offTime*2毫秒。
备注：当结束时间设置值小于开始时间时，结束时间等于开始时间。

返回值：

错误代码	值	描述
ERROR_CM_SUCCESS	0	成功
ERROR_CM_INVALID_HANDLE	-2	失败，句柄无效
ERROR_CM_INVALID_PARAMETER	-1	参数无效
ERROR_CM_INSUFFICIENT_MEMORY	-4	申请内存失败
ERROR_IO_WRITE_FAILED	-9	写入数据失败
ERROR_IO_WRITE_TIMEOUT	-10	写入数据超时

5.17. PrintText

此函数功能为打印文本数据。

```
int PrintText(
    void* hPrinter,
    const char* data,
    int alignment,
    int textSize
);
```

参数：

void* hPrinter
[in] 创建的目标打印机对象。

const char* data,
[in] 所需打印的文本数据。

int alignment
[in] 文本对齐方式。

对齐方式	值	描述
ALIGNMENT_LEFT	0	左对齐
ALIGNMENT_CENTER	1	居中
ALIGNMENT_RIGHT	2	右对齐

int textSize
[in] 设置字体大小 (文本长度超过打印纸范围将不被打印)。

<div> <div>横向放大倍数</div> <div>纵向放大倍数</div> </div>	1		2		3		4		5		6		7		8	
	十六进制	十进制	十六进制	十进制	十六进制	十进制	十六进制	十进制	十六进制	十进制	十六进制	十进制	十六进制	十进制	十六进制	十进制
1	00	0	10	16	20	32	30	48	40	64	50	80	60	96	70	112
2	01	1	11	17	21	33	31	49	41	65	51	81	61	97	71	113
3	02	2	12	18	22	34	32	50	42	66	52	82	62	98	72	114
4	03	3	13	19	23	35	33	51	43	67	53	83	63	99	73	115
5	04	4	14	20	24	36	34	52	44	68	54	84	64	10	74	116

														0		
6	05	5	15	21	25	37	35	53	45	69	55	85	65	101	75	117
7	06	6	16	22	26	38	36	54	46	70	56	86	66	102	76	118
8	07	7	17	23	27	39	37	55	47	71	57	87	67	103	77	119

返回值：

错误代码	值	描述
ERROR_CM_SUCCESS	0	成功
ERROR_CM_INVALID_HANDLE	-2	失败，句柄无效
ERROR_CM_INVALID_PARAMETER	-1	参数无效
ERROR_CM_INSUFFICIENT_MEMORY	-4	申请内存失败
ERROR_IO_WRITE_FAILED	-9	写入数据失败
ERROR_IO_WRITE_TIMEOUT	-10	写入数据超时

5.18. SetRelativeHorizontal

设置相对横向打印位置

```
int SetRelativeHorizontal(
    void* hPrinter,
    int position
)
```

参数：

void* hPrinter

[in] 创建的目标打印机对象。

int position

[in] 相对位置

返回值：

错误代码	值	描述
ERROR_CM_SUCCESS	0	成功
ERROR_CM_INVALID_HANDLE	-2	失败，句柄无效
ERROR_CM_INVALID_PARAMETER	-1	参数无效
ERROR_CM_INSUFFICIENT_MEMORY	-4	申请内存失败
ERROR_IO_WRITE_FAILED	-9	写入数据失败
ERROR_IO_WRITE_TIMEOUT	-10	写入数据超时

5.19. PrintTextS

此函数功能为打印文本。

```
int PrintTextS(
    void* hPrinter,
    const char* data
);
```

参数：

void* hPrinter

[in] 创建的目标打印机对象。

const char* data
[in] 需要打印的文件数据。

返回值：

错误代码	值	描述
ERROR_CM_SUCCESS	0	成功
ERROR_CM_INVALID_HANDLE	-2	失败，句柄无效
ERROR_CM_INVALID_PARAMETER	-1	参数无效
ERROR_CM_INSUFFICIENT_MEMORY	-4	申请内存失败
ERROR_IO_WRITE_FAILED	-9	写入数据失败
ERROR_IO_WRITE_TIMEOUT	-10	写入数据超时

5.20. PrintBarcode

此函数功能为打印条码。在标准模式下，条码打印位置在新行或者没有数据存在于缓存时才能正常打印。在页模式下，未接受打印条码命令时，条码数据保存于缓存中，不打印条码。

```
int PrintBarcode(  
    void* hPrinter,  
    int bcType,  
    const char* bcData,  
    int width,  
    int height,  
    int alignment,  
    int hriPosition  
);
```

参数：

void* hPrinter
[in] 创建的目标打印机对象。

int bcType
[in] 设置条码类型。

const char* bcData,
[in] 条码数据。

条码类型	值	条码数据长度	数据有效值范围
BARCODE_UPC_A	65	11≤n≤12	48≤data≤57
BARCODE_UPC_E	66	n=6	48≤data≤57
BARCODE_EAN13 BARCODE_JAN13	67	12≤n≤13	48≤data≤57
BARCODE_EAN8 BARCODE_JAN8	68	7≤n≤8	48≤data≤57
BARCODE_CODE39	69	1≤n≤255	48≤data≤57,65≤data≤90, data=32,36,37,43,45,46,47
BARCODE_ITF	70	1≤n≤255 (even number)	48≤data≤57
BARCODE_CODABAR	71	1≤n≤255	48≤data≤57,65≤data≤68, data =36,43,45,46,47,58
BARCODE_CODE93	72	1≤n≤255	0≤data≤127
BARCODE_CODE128	73	2≤n≤255	0≤data≤127

int width
[in] 条码宽度有效值范围：2-7，当条码打印宽度超过打印纸可打印范围，条码不打印。此参数对二维码无效。

int height

[in] 设置条码打印高度。有效范围：1-255。

int alignment

[in] 设置条码对齐方式。

对齐方式	值	描述
ALIGNMENT_LEFT	0	左对齐
ALIGNMENT_CENTER	1	居中
ALIGNMENT_RIGHT	2	右对齐

int hriPosition

[in] 设置条码可见字符位置。

位置	值	描述
BRACODE_HRI_NONE	0	不打印可见字符
BRACODE_HRI_ABOVE	1	在条码上方打印可见字符
BRACODE_HRI_BELOW	2	在条码下方打印可见字符
BRACODE_HRI_BOTH	3	在条码上、下方打印可见字符

返回值：

错误代码	值	描述
ERROR_CM_SUCCESS	0	成功
ERROR_CM_INVALID_HANDLE	-2	失败，句柄无效
ERROR_CM_INVALID_PARAMETER	-1	参数无效
ERROR_CM_INSUFFICIENT_MEMORY	-4	申请内存失败
ERROR_IO_WRITE_FAILED	-9	写入数据失败
ERROR_IO_WRITE_TIMEOUT	-10	写入数据超时

5.21. PrintSymbol

此函数功能为打印二维码。

```
int PrintSymbol(  
    void* hPrinter,  
    int type,  
    const char* data,  
    int errLevel,  
    int width,  
    int height,  
    int alignment  
);
```

参数：

void* hPrinter

[in] 创建的目标打印机对象。

int type

[in] 二维码类型

类型	值	描述
BARCODE_STANDARD_PDF417	48	标准样式 PDF417码
BARCODE_QRCODE	49	QR Code

const char* data

[in] 二维码内容。

int errLevel

[in] 二维码设置错误校正等级。

错误校正等级	值	代码或容错率
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] 二维码宽度 0≤n≤255。

int height

[in] 二维码高度0≤n≤255 (此参数对 QRCode 无效)。

int alignment

[in] 二维码对齐方式。

对齐方式	值	描述
ALIGNMENT_LEFT	0	左对齐
ALIGNMENT_CENTER	1	居中
ALIGNMENT_RIGHT	2	右对齐

返回值:

错误代码	值	描述
ERROR_CM_SUCCESS	0	成功
ERROR_CM_INVALID_HANDLE	-2	失败，句柄无效
ERROR_CM_INVALID_PARAMETER	-1	参数无效
ERROR_CM_INSUFFICIENT_MEMORY	-4	申请内存失败
ERROR_IO_WRITE_FAILED	-9	写入数据失败
ERROR_IO_WRITE_TIMEOUT	-10	写入数据超时

5.22. PrintImage

此函数功能为打印指定的图片。在页模式下，位图只储存在打印缓冲区且不打印。
注意：图片宽度不要超过打印宽度。

```
int PrintImage(  
    void* hPrinter,  
    const char* imagePath,  
    int scaleMode  
);  
int PrintImageW(  
    void* hPrinter,  
    const wchar_t* imagePathW,  
    int scaleMode  
);
```

参数：

void* hPrinter

[in] 创建的目标打印机对象。

const char* imagePath

const wchar_t* imagePathW

[in] 图片的完整路径。

int scaleMode

[in] 打印图片的缩放模式。

模式	值	描述
PRINT_IMAGE_NORMAL	0	正常模式
PRINT_IMAGE_DOUBLE_WIDTH	1	倍宽模式
PRINT_IMAGE_DOUBLE_HEIGHT	2	倍高模式
PRINT_IMAGE_QUADRUPLE	3	四倍模式

返回值：

错误代码	值	描述
ERROR_CM_SUCCESS	0	成功
ERROR_CM_INVALID_HANDLE	-2	失败，句柄无效
ERROR_CM_INVALID_PARAMETER	-1	参数无效
ERROR_CM_INSUFFICIENT_MEMORY	-4	申请内存失败
ERROR_IO_WRITE_FAILED	-9	写入数据失败
ERROR_IO_WRITE_TIMEOUT	-10	写入数据超时

5.23. DefineNVImageCompatible

此函数功能为在 NV 图形缓存区域定义指定的 NV 位图。可以同时下载多张图片，下载到打印机的图片编号从 1 开始累加。只有部分机型支持此功能，未来机型可能不支持此功能。

注意：图片宽度不要超过打印宽度。

```
int DefineNVImageCompatible(  
    void* hPrinter,  
    const char** fileNameList,  
    int imageQty  
);  
int DefineNVImageCompatibleW(  
    void* hPrinter,  
    const wchar_t** fileNameListW,  
    int imageQty  
);
```

参数：

void* hPrinter

[in] 创建的目标打印机对象。

const char** fileNameList

const wchar_t** fileNameListW

[in] 指定的图片路径列表。

int imageQty

[in] 指定图片的数量。

返回值：

错误代码	值	描述
ERROR_CM_SUCCESS	0	成功
ERROR_CM_INVALID_HANDLE	-2	失败，句柄无效
ERROR_CM_INVALID_PARAMETER	-1	参数无效
ERROR_CM_INSUFFICIENT_MEMORY	-4	申请内存失败
ERROR_IO_WRITE_FAILED	-9	写入数据失败

ERROR_IO_WRITE_TIMEOUT	-10	写入数据超时
------------------------	-----	--------

5.24. PrintNVImageCompatible

此函数功能为打印由<DefineNVImageCompatible>下载的 NV 位图。只有部分机型 支持此功能，未来机型可能不支持此功能。

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

参数：

void* hPrinter

[in] 创建的目标打印机对象。

int imgNo

[in] 打印指定的第 n 张图片 (不打印在 NV 缓冲区未定义的图片系列号)。1≤n≤255

int scaleMode

[in] 打印图片的缩放模式。

返回值：

错误代码	值	描述
ERROR_CM_SUCCESS	0	成功
ERROR_CM_INVALID_HANDLE	-2	失败，句柄无效
ERROR_CM_INVALID_PARAMETER	-1	参数无效
ERROR_CM_INSUFFICIENT_MEMORY	-4	申请内存失败
ERROR_IO_WRITE_FAILED	-9	写入数据失败
ERROR_IO_WRITE_TIMEOUT	-10	写入数据超时

5.25. SelectPageMode

此函数功能为切换标准模式到页模式(仅打印机支持页模式并且在标准模式下时 有效)。

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

参数：

void* hPrinter

[in] 创建的目标打印机对象。

返回值：

错误代码	值	描述
ERROR_CM_SUCCESS	0	成功
ERROR_CM_INVALID_HANDLE	-2	失败，句柄无效
ERROR_CM_INVALID_PARAMETER	-1	参数无效
ERROR_CM_INSUFFICIENT_MEMORY	-4	申请内存失败
ERROR_IO_WRITE_FAILED	-9	写入数据失败
ERROR_IO_WRITE_TIMEOUT	-10	写入数据超时

5.26. SelectStandardMode

此函数功能为切换页模式到标准模式 (仅在页模式下有效)。

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

参数：

void* hPrinter
[in] 创建的目标打印机对象。

返回值：

错误代码	值	描述
ERROR_CM_SUCCESS	0	成功
ERROR_CM_INVALID_HANDLE	-2	失败，句柄无效
ERROR_CM_INVALID_PARAMETER	-1	参数无效
ERROR_CM_INSUFFICIENT_MEMORY	-4	申请内存失败
ERROR_IO_WRITE_FAILED	-9	写入数据失败
ERROR_IO_WRITE_TIMEOUT	-10	写入数据超时

5.27. SelectPrintDirectionInPageMode

此函数功能为在页模式下，选择打印机的打印方向。此函数只有在页模式下有效。

```
int SelectPrintDirectionInPageMode(  
    void* hPrinter,  
    int direction  
);
```

参数：

void* hPrinter
[in] 创建的目标打印机对象。

int direction
[in] 选择打印方向。

打印方向	值	描述	起始位置
PRINT_DIRECTION_LEFT_TO_RIGHT	0	左->右	左上角
PRINT_DIRECTION_BOTTOM_TO_TOP	1	下->上	左下角
PRINT_DIRECTION_RIGHT_TO_LEFT	2	右->左	右下角
PRINT_DIRECTION_TOP_TO_BOTTOM	3	上->下	右上角

返回值：

错误代码	值	描述
ERROR_CM_SUCCESS	0	成功
ERROR_CM_INVALID_HANDLE	-2	失败，句柄无效
ERROR_CM_INVALID_PARAMETER	-1	参数无效
ERROR_CM_INSUFFICIENT_MEMORY	-4	申请内存失败
ERROR_IO_WRITE_FAILED	-9	写入数据失败
ERROR_IO_WRITE_TIMEOUT	-10	写入数据超时

5.28. SetAbsoluteVerticalPrintPositionInPageMode

此函数功能为在页模式下，设置垂直打印位置(当起始打印位置为左上角或右下角时，为设置垂直位置；当起始打印位置为左下角或右上角时，为设置水平位置)。

```
int SetAbsoluteVerticalPrintPositionInPageMode(  
    void* hPrinter,  
    int position  
);
```

参数：

void* hPrinter

[in] 创建的目标打印机对象。

int position

[in] 设置垂直位置。

返回值：

错误代码	值	描述
ERROR_CM_SUCCESS	0	成功
ERROR_CM_INVALID_HANDLE	-2	失败，句柄无效
ERROR_CM_INVALID_PARAMETER	-1	参数无效
ERROR_CM_INSUFFICIENT_MEMORY	-4	申请内存失败
ERROR_IO_WRITE_FAILED	-9	写入数据失败
ERROR_IO_WRITE_TIMEOUT	-10	写入数据超时

5.29. PrintAndReturnStandardMode

此函数功能为打印并返回到标准模式(仅在页模式下有效)。

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

参数：

void* hPrinter

[in] 创建的目标打印机对象。

返回值：

错误代码	值	描述
ERROR_CM_SUCCESS	0	成功
ERROR_CM_INVALID_HANDLE	-2	失败，句柄无效
ERROR_CM_INVALID_PARAMETER	-1	参数无效
ERROR_CM_INSUFFICIENT_MEMORY	-4	申请内存失败
ERROR_IO_WRITE_FAILED	-9	写入数据失败
ERROR_IO_WRITE_TIMEOUT	-10	写入数据超时

5.30. SetPrintAreaInPageMode

此函数功能为在页模式下，设置打印区域的大小和逻辑起点。打印区域的宽度和高度都不能设为零。

```
int SetPrintAreaInPageMode(
    void* hPrinter,
    int horizontal,
    int vertical,
    int width,
    int height
);
```

参数：

void* hPrinter

[in] 创建的目标打印机对象。

int horizontal

[in] 设置起始打印的水平位置(范围：0-32000，单位：dot)。

int vertical

[in] 设置起始打印的垂直位置(范围：0-32000，单位：dot)。

int width

[in] 设置可打印区域的水平宽度。

int height

[in] 设置可打印区域的垂直高度。

当打印纸为 80mm 宽度时：水平起点 = 0, 垂直起点 = 0, 宽度 = 576, 高度 = 840。

返回值：

错误代码	值	描述
ERROR_CM_SUCCESS	0	成功
ERROR_CM_INVALID_HANDLE	-2	失败，句柄无效
ERROR_CM_INVALID_PARAMETER	-1	参数无效
ERROR_CM_INSUFFICIENT_MEMORY	-4	申请内存失败
ERROR_IO_WRITE_FAILED	-9	写入数据失败
ERROR_IO_WRITE_TIMEOUT	-10	写入数据超时

5.31. PrintDataInPageMode

此函数功能为在页模式下打印数据，打印后不返回标准模式 (仅在页模式下有效)。

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

参数：

void* hPrinter

[in] 创建的目标打印机对象。

返回值：

错误代码	值	描述
ERROR_CM_SUCCESS	0	成功
ERROR_CM_INVALID_HANDLE	-2	失败，句柄无效
ERROR_CM_INVALID_PARAMETER	-1	参数无效
ERROR_CM_INSUFFICIENT_MEMORY	-4	申请内存失败
ERROR_IO_WRITE_FAILED	-9	写入数据失败
ERROR_IO_WRITE_TIMEOUT	-10	写入数据超时

5.32. SetAbsolutePrintPosition

此函数功能为从打印区域左边沿移动打印位置至 $n \times$ (水平或垂直移动单元)。打印机忽略超过打印区域的任何设置。

选择标准模式时，使用水平移动单元。

选择页模式时，使用水平或垂直移动单元作为打印方向。

```
int SetAbsolutePrintPosition(  
    void* hPrinter,  
    int position  
);
```

参数：

void* hPrinter

[in] 创建的目标打印机对象。

int position

[in] 水平起始打印位置。

返回值：

错误代码	值	描述
ERROR_CM_SUCCESS	0	成功
ERROR_CM_INVALID_HANDLE	-2	失败，句柄无效
ERROR_CM_INVALID_PARAMETER	-1	参数无效
ERROR_CM_INSUFFICIENT_MEMORY	-4	申请内存失败
ERROR_IO_WRITE_FAILED	-9	写入数据失败
ERROR_IO_WRITE_TIMEOUT	-10	写入数据超时

5.33. PositionNextLabel

此函数功能为打印标签内容并定位下一个标签起始位置。

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

参数：

void* hPrinter

[in] 创建的目标打印机对象。

返回值：

错误代码	值	描述
ERROR_CM_SUCCESS	0	成功
ERROR_CM_INVALID_HANDLE	-2	失败，句柄无效
ERROR_CM_INVALID_PARAMETER	-1	参数无效
ERROR_CM_INSUFFICIENT_MEMORY	-4	申请内存失败
ERROR_IO_WRITE_FAILED	-9	写入数据失败
ERROR_IO_WRITE_TIMEOUT	-10	写入数据超时

5.34. SetAlign

此函数功能为设置打印对齐方式。在页模式下对齐方式无效。

```
int SetAlign(  
    void* hPrinter,  
    int align  
);
```

参数：

void* hPrinter

[in] 创建的目标打印机对象。

int align

[in] 设置对齐方式。

值	对齐方式
0,48	左对齐
1,49	居中
2,50	右对齐

返回值：

错误代码	值	描述
ERROR_CM_SUCCESS	0	成功
ERROR_CM_INVALID_HANDLE	-2	失败，句柄无效
ERROR_CM_INVALID_PARAMETER	-1	参数无效
ERROR_CM_INSUFFICIENT_MEMORY	-4	申请内存失败
ERROR_IO_WRITE_FAILED	-9	写入数据失败
ERROR_IO_WRITE_TIMEOUT	-10	写入数据超时

5.35. SetTextBold

此函数功能为打开或关闭突出模式。

```
int SetTextBold(  
    void* hPrinter,  
    int bold  
);
```

参数：

void* hPrinter

[in] 创建的目标打印机对象。

int bold

[in] 设置文本突出模式。

0：突出模式关闭。

1：突出模式打开。

返回值：

错误代码	值	描述
ERROR_CM_SUCCESS	0	成功
ERROR_CM_INVALID_HANDLE	-2	失败，句柄无效
ERROR_CM_INVALID_PARAMETER	-1	参数无效
ERROR_CM_INSUFFICIENT_MEMORY	-4	申请内存失败

ERROR_IO_WRITE_FAILED	-9	写入数据失败
ERROR_IO_WRITE_TIMEOUT	-10	写入数据超时

5.36. SetTextFont

此函数功能为设置文本字体。

```
int SetTextFont(  
    void* hPrinter,  
    int font  
);
```

参数：

void* hPrinter

[in] 创建的目标打印机对象。

int font

[in] 设置字体类型

值	字体
0,48	Font A
1,49	Font B

返回值：

错误代码	值	描述
ERROR_CM_SUCCESS	0	成功
ERROR_CM_INVALID_HANDLE	-2	失败，句柄无效
ERROR_CM_INVALID_PARAMETER	-1	参数无效
ERROR_CM_INSUFFICIENT_MEMORY	-4	申请内存失败
ERROR_IO_WRITE_FAILED	-9	写入数据失败
ERROR_IO_WRITE_TIMEOUT	-10	写入数据超时

5.37. SetHorizontalAndVerticalMotionUnits

此函数功能为设置水平方向、垂直方向的移动单元。

```
int SetHorizontalAndVerticalMotionUnits(  
    void* hPrinter,  
    int horizontal,  
    int vertical  
);
```

参数：

void* hPrinter

[in] 创建的目标打印机对象。

int horizontal

[in] 水平运动单位 $0 \leq \text{horizontal} \leq 255$ 。

int vertical

[in] 垂直运动单位 $0 \leq \text{vertical} \leq 255$ 。

返回值：

错误代码	值	描述
ERROR_CM_SUCCESS	0	成功

ERROR_CM_INVALID_HANDLE	-2	失败，句柄无效
ERROR_CM_INVALID_PARAMETER	-1	参数无效
ERROR_CM_INSUFFICIENT_MEMORY	-4	申请内存失败
ERROR_IO_WRITE_FAILED	-9	写入数据失败
ERROR_IO_WRITE_TIMEOUT	-10	写入数据超时

5.38. EnableBlackMark

此函数功能为启用/关闭黑标功能。

```
int EnableBlackMark (
    void* hPrinter,
    int enable
);
```

参数：

void* hPrinter

[in] 创建的目标打印机对象。

int enable

[in] 0 表示关闭黑标功能，非0 表示启用黑标功能

返回值：

错误代码	值	描述
ERROR_CM_SUCCESS	0	成功
ERROR_CM_INVALID_HANDLE	-2	失败，句柄无效
ERROR_CM_INVALID_PARAMETER	-1	参数无效
ERROR_CM_INSUFFICIENT_MEMORY	-4	申请内存失败
ERROR_IO_WRITE_FAILED	-9	写入数据失败
ERROR_IO_WRITE_TIMEOUT	-10	写入数据超时

5.39. SetBlackMarkDistance

此函数功能为设置黑标间隔距离。

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

参数：

void* hPrinter

[in] 创建的目标打印机对象。

int distance

[in] 设置黑标间隔长度，以点为单位，取值范围400 <= distance <= 4000

返回值：

错误代码	值	描述
ERROR_CM_SUCCESS	0	成功
ERROR_CM_INVALID_HANDLE	-2	失败，句柄无效
ERROR_CM_INVALID_PARAMETER	-1	参数无效
ERROR_CM_INSUFFICIENT_MEMORY	-4	申请内存失败
ERROR_IO_WRITE_FAILED	-9	写入数据失败

ERROR_IO_WRITE_TIMEOUT	-10	写入数据超时
------------------------	-----	--------

5.40. SetBlackMarkHeight

此函数功能为设置黑标间隔距离。

```
int SetBlackMarkHeight(  
    void* hPrinter,  
    int height  
);
```

参数：

void* hPrinter

[in] 创建的目标打印机对象。

int height

[in] 设置黑标高度，以点为单位，取值范围24 <= height <= 240

返回值：

错误代码	值	描述
ERROR_CM_SUCCESS	0	成功
ERROR_CM_INVALID_HANDLE	-2	失败，句柄无效
ERROR_CM_INVALID_PARAMETER	-1	参数无效
ERROR_CM_INSUFFICIENT_MEMORY	-4	申请内存失败
ERROR_IO_WRITE_FAILED	-9	写入数据失败
ERROR_IO_WRITE_TIMEOUT	-10	写入数据超时

5.41. DefineUserDefinedCharacters

此函数功能为自定义下载字符集。

```
int DefineUserDefinedCharacters(  
    void* hPrinter,  
    unsigned char* data  
    int size  
);
```

参数：

void* hPrinter

[in] 创建的目标打印机对象。

unsigned char* data

[in] 自定义的字符集数据，包括头部信息。具体结构如下：

fonttype c1 c2 name[32] fontdata[d1...dk]

当fonttype=0时，字体为fontA；

当fonttype=1时，字体为fontB；

c1 c2表示自定义字符区间（32<=c1<=c2<=255）

name：可以用32个字符表示用户自定义字符集的名字

fontdata：字体数据

int size

[in] 传入的数据长度

返回值：

错误代码	值	描述
------	---	----

ERROR_CM_SUCCESS	0	成功
ERROR_CM_INVALID_HANDLE	-2	失败，句柄无效
ERROR_CM_INVALID_PARAMETER	-1	参数无效
ERROR_CM_INSUFFICIENT_MEMORY	-4	申请内存失败
ERROR_IO_WRITE_FAILED	-9	写入数据失败
ERROR_IO_WRITE_TIMEOUT	-10	写入数据超时

5.42. DeleteUserDefinedCharacter

此函数功能为取消用户自定义字符集。

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

参数：

void* hPrinter

[in] 创建的目标打印机对象。

int n

[in] 取消用户自定义字符中代码为n的字符。

返回值：

错误代码	值	描述
ERROR_CM_SUCCESS	0	成功
ERROR_CM_INVALID_HANDLE	-2	失败，句柄无效
ERROR_CM_INVALID_PARAMETER	-1	参数无效
ERROR_CM_INSUFFICIENT_MEMORY	-4	申请内存失败
ERROR_IO_WRITE_FAILED	-9	写入数据失败
ERROR_IO_WRITE_TIMEOUT	-10	写入数据超时

5.43. GoHomeWithBlackMark

将黑标打印纸进纸到打印起始位置。

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

参数：

void* hPrinter

[in] 创建的目标打印机对象。

返回值：

错误代码	值	描述
ERROR_CM_SUCCESS	0	成功
ERROR_CM_INVALID_HANDLE	-2	失败，句柄无效
ERROR_CM_INVALID_PARAMETER	-1	参数无效
ERROR_CM_INSUFFICIENT_MEMORY	-4	申请内存失败
ERROR_IO_WRITE_FAILED	-9	写入数据失败

ERROR_IO_WRITE_TIMEOUT	-10	写入数据超时
------------------------	-----	--------

[注释] • 仅当开启黑标功能时，该命令才被激活。

- 该命令将下一个打印位置设定在一行的开始。
- 即使该命令在标记打印纸的打印起始位置执行，打印机并不将打印进纸到下一个打印起始位置。
- 使用黑标功能时需要使用该指令

5.44. SetBlackMarkAdjust

将黑标打印纸进纸到打印起始位置。

```
int SetBlackMarkAdjust(
    void* hPrinter,
    int func,
    int direction,
    int offset
);
```

参数：

void* hPrinter
 [in] 创建的目标打印机对象。

Int func
 [in] 取值1用于设置起始打印位置，取值2用于设置开始裁纸位置。

Int direction
 [in] 指定调整的方向，0/48指定为进纸的方向，1/49指定为与进制方向相反。

Int offset
 [in] 位置距离，单位点

返回值：

错误代码	值	描述
ERROR_CM_SUCCESS	0	成功
ERROR_CM_INVALID_HANDLE	-2	失败，句柄无效
ERROR_CM_INVALID_PARAMETER	-1	参数无效
ERROR_CM_INSUFFICIENT_MEMORY	-4	申请内存失败
ERROR_IO_WRITE_FAILED	-9	写入数据失败
ERROR_IO_WRITE_TIMEOUT	-10	写入数据超时

5.45. FirmwareUpgrade

此函数功能为升级打印机固件

```
int FirmwareUpgrade(
    void* handle,
    const char* cFileName,
    void (*progressCallback)(float)
);
```

参数：

void* handle
 [in] 创建的目标打印机对象。

```
const char* cFileName
[in] 固件文件路径
void (*progressCallback)(float)
更新进度回调
```

状态	值
更新进度	0~1
更新成功	0
内存不足	-4
读取文件失败	-11
发送数据失败	-9
更新失败	-18

返回值:

错误代码	值	描述
ERROR_CM_SUCCESS	0	成功
ERROR_CM_INVALID_HANDLE	-2	失败，句柄无效
ERROR_CM_INVALID_PARAMETER	-1	参数无效
ERROR_CM_INSUFFICIENT_MEMORY	-4	申请内存失败
ERROR_IO_WRITE_FAILED	-9	写入数据失败
ERROR_IO_WRITE_TIMEOUT	-10	写入数据超时

5.46. SearchEscNetDevice

此函数功能为搜索同一个局域网的打印机设备信息。

```
int SearchEscNetDevice(
    int revTimeout,
    void (callback)(unsigned char* data)
)
```

参数:

```
int revTimeout
[in] 超时时间
void (callback)(unsigned char* data)
[out]搜到打印设备时的回调。
data的组成结构:
0~5字节表示mac地址
8~11字节表示IP地址
12~15字节表示子网掩码
16~19字节表示默认网关
22字节表示DHCP， 1为开启， 0为关闭。
```

返回值:

错误代码	值	描述
ERROR_CM_SUCCESS	0	成功
ERROR_CM_INVALID_HANDLE	-2	失败，句柄无效
ERROR_CM_INVALID_PARAMETER	-1	参数无效
ERROR_CM_INSUFFICIENT_MEMORY	-4	申请内存失败
ERROR_IO_WRITE_FAILED	-9	写入数据失败
ERROR_IO_WRITE_TIMEOUT	-10	写入数据超时
ERROR_IO_READ_FAILED	-11	读失败
ERROR_IO_OPEN_FAILED	-8	打开IO失败

举例:

```

SearchEscNetDevice(5, [](BYTE* data)
{
    printf("mac:%02hX:%02hX:%02hX:%02hX:%02hX\n", data[0], data[1],
data[2], data[3], data[4], data[5]);
    printf("IP:%d.%d.%d.%d\n", data[8], data[9], data[10], data[11]);
    printf("mask:%d.%d.%d.%d\n", data[12], data[13], data[14], data[15]);
    printf("gateway:%d.%d.%d.%d\n", data[16], data[17], data[18], data[19]);
    printf("DHCP:%d\n", data[22]);
}
);

```

打印结果:

```

mac:00:ED:09:21:53:3D
IP:192.168.91.28
mask:255.255.255.0
gateway:192.168.91.1
DHCP:1

```

5.47. SetEscNetInfo

此函数功能为根据网口打印机的mac地址，设置对应的网络信息。

```

int SetEscNetInfo(
    unsigned char* mac,
    unsigned char* ipAddress,
    unsigned char* mask,
    unsigned char* gateway,
    unsigned char dhcp
);

```

参数:

```

unsigned char* mac
    [in] mac地址
unsigned char* ipAddress
    [in] ip地址
unsigned char* mask
    [in] 子网掩码
unsigned char* gateway
    [in] 默认网关
unsigned char dhcp
    [in] dhcp 1为开 0为关

```

返回值:

错误代码	值	描述
ERROR_CM_SUCCESS	0	成功
ERROR_CM_INVALID_HANDLE	-2	失败，句柄无效
ERROR_CM_INVALID_PARAMETER	-1	参数无效
ERROR_CM_INSUFFICIENT_MEMORY	-4	申请内存失败
ERROR_IO_WRITE_FAILED	-9	写入数据失败
ERROR_IO_WRITE_TIMEOUT	-10	写入数据超时
ERROR_IO_READ_FAILED	-11	读失败
ERROR_IO_OPEN_FAILED	-8	打开IO失败