NEmuPrinter API

SDK guide

● Overview

Describes overview of SDK

● Win32 API reference

Describes syntax of API

● .net API reference

Describes syntax of API used in .net environment.

Document disclaimer

- The description in this document is taken all possible measures to ensure the correctness, however if you find any, please contact us.
- Descriptions are subject to change without prior notice.
  Please ask for up-to-date information.

- All rights of this description reserved.

- We are not responsible for any influence by the results from the use.

- We assume no responsibility whatsoever for any damages resulting from improper use, the use without understanding of this description, repair and change by the third party.

Trademark

MS-DOS®,Microsoft®,Win32®,Windows®,Windows Vista®,Visual Studio®,Visual Basic®,Visual C++®,Visual C#® are trademarks or registered trademarks of Microsoft Corporation in USA and their respective countries.

Signs

Following signs are used in this document. Please use the products on the base of understanding of the signs.

Caution	It indicates the precautions to be observed fully. Disregard of the signs and wrong usage may casue malfunction and inoperative.
Reference	It indicates supplement description and relevant matters.

Usage restrictions

In the event that this product is used with devices that require high reliability and safety for function and accuracy on conveyance including aircraft, train, vehicle; disaster prevention and security device, users shall use products after considering the safety design of whole system by applying fail-safe and redundancy design to maintain reliability and safety.
This product is not designed to use with devices that require extremely high reliability and safety including aerospace mechanism, signal axis mechanism, nuclear control device, medical device. Users shall ascertain and evaluate suitability of this product for those application.

Table of contents

·Overview

·Win32 API Reference

·.net API Reference

- Property

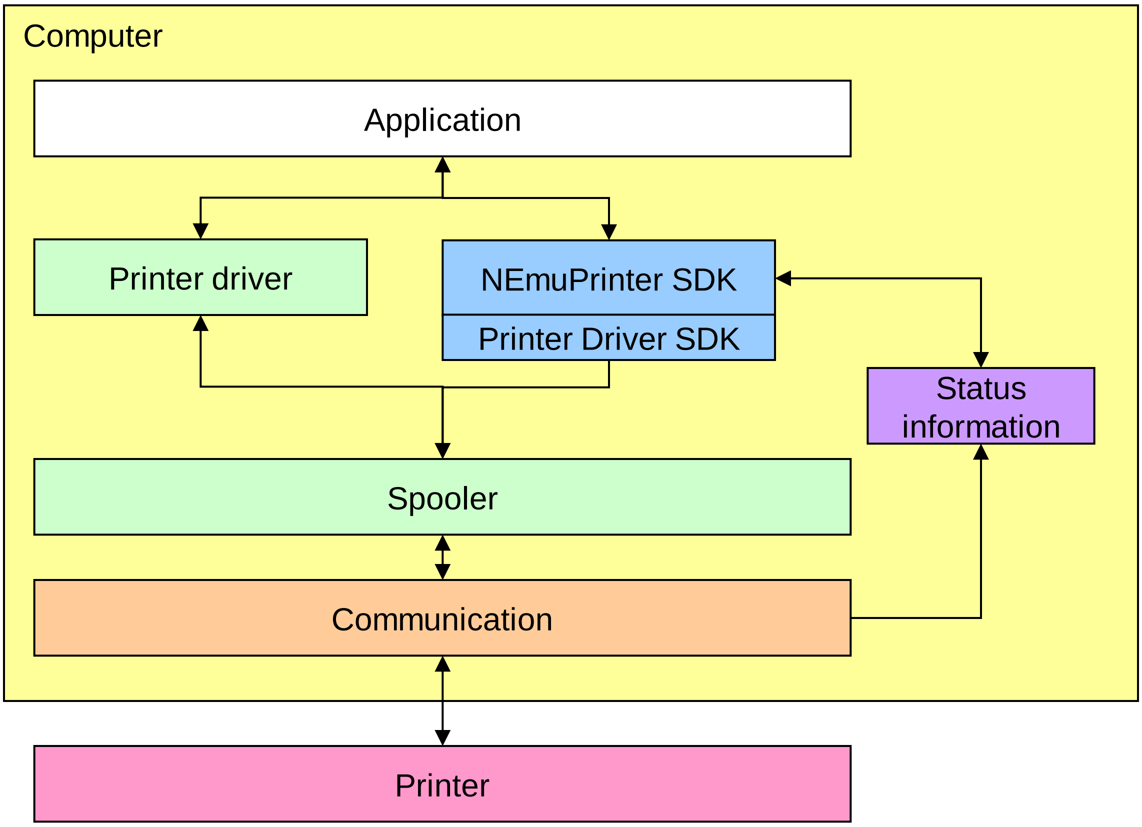
ー Method

Overview

The function of printing and printer monitoring can be installed in targeted application by using SDK.

SDK is distributed as Dynamic Link Library (DLL).
File name: NEmuPrinterAPI.dll
NEmuPrinterAPIC.dll

System configuration with SDK



Development language

Win32
- Visual Basic 6.0
- Visual C++

.NET
- Visual Basic .NET
- Visual C#

Install

NemuPrinter SDK is installed automatically when NExD (NPI EX Driver) is installed.

Reference

You can find required files for environment construction in each sample in this manual.

Construction of development environment

Visual Basic 6.0

Following is an example of development environment construction for Visual Basic.

1. Copy API definition file (NEmuPrinterAPI.bas) from VBSample folder to working folder where you make a project for an application.
2. Run the Microsoft Visual Basic and open project window.
3. Select [Add standard module] from [Project] menu on menu bar.
4. A window to add module will open. Select [Existing file] tab, specify the definition file copied in above step #1 and click [Open] button.
5. [Module1.bas] will be added to project explorer.
6. Select [Reference] from [Project] menu on menu bar.
7. Development environment for Visual Basic 6.0 will be ready.

Visual C++

Following is an example of development environment construction for Visual C++.

1. Run the Microsoft Visual C++ and open project window.
2. Copy API definition file (ImportApiEx.h) to working folder where you make a project for an application.
3. Open Source File. Define (ImportApiEx.h) by using #include directory.
   Definition method: #include "ImportApiEx.h"
4. Development environment for Visual C++ will be ready.

* Use NEmuPrinterAPI.ba, ImportApiEx.h after the modification of those files depending on your usage since those files are just reference.

Category of API

Following APIs are available.

Application	API	Description
Start/Finish NEmuPrinterAPI	EiOpenMonPrinter	Start to use NEmuPrinterAPI with the specified printer
	EiCloseMonPrinter	Finish to use NEmuPrinterAPI
Occupy printer	EiLockPrinter	Lock printer
	EiUnlockPrinter	Unlock printer
Sending command	EiDirectIO	Send/Receive binary data
	EiDirectIOEx	Send/Receive binary data (Other items than status can be specified)
Reset printer	EiResetPrinter	Reset printer
Obtaining status	EiGetStatus	Get printer status
	EiSetStatusBackFunction	Register call back function which will be called when auto status notice.
	EiSetStatusBackFunctionEx	Register call back function which will be called when auto status notice. (It is possible to get port name)
	EiSetStatusBackWnd	Register window handle of button to send click event when auto status notice and memory address to set status information.
	EiCancelStatusBack	Cancel auto status notice request process
Power OFF preprocessing	EiPowerOff	Power off printer or put it in standby mode
Get and reset maintenance counter	EiGetCounter	Get maintenance counter
	EiResetCounter	Reset maintenance counter
Get printer information	EiGetPrnCapability	Get printer information
Drawer contorl	EiOpenDrawer	Open drawer
Edit DevMode	EmuDM_GetProperty	Get the target DevMode information
	EmuDM_SetProperty	Set the target DevMode information
Draw barcode	EmuBarcode	Draw barcode

Function name	EiOpenMonPrinter / EiOpenMonPrinterW		
Argument name	**IN/OUT**	**Type**	**Description**
i_type	I	int	2 (fixed)
i_prt	I	PCHAR / PWCHAR	Output printer driver name
Return value	INT		

·Error (negative value), Normal end (returns handle which specifies printer)

Processing Description

·Starts to use NEmuPrinterAPI with the specified printer and returns a handle.

Error description

-10 : Parameter error for i_type
-20 : The specified printer is already opened
-30 : The specified printer does not exist
-50 : Memory shortage
-80 : Unable to R/W to printer
(Printer is not powered on, bad cable connection etc)
-340 : LM does not run or status information acquisition error from LM
-350 : Spooler does not run

Function name	EiCloseMonPrinter		
Argument name	IN/OUT	Type	Description
i_hdl	I	int	Handle
Return value	INT		
・Error (negative value), Normal end (0)			
Processing Description			
·Finish to use NEmuPrinterAPI with the specified printer Error Description -60 : Handle value is invalid			

Function name	EiLockPrinter		
Argument name	IN/OUT	Type	Description
i_hdl i_timeout	I I	int DWORD	Handle Timeout period(msec)
Return value	INT		
・Error (negative value), Normal end (0)			
Processing Description			
·Lock printer Error Description -60 : Handle value is invalid -340 : Spooler does not run -1000 : Unable to execute this function because printer is locked			

Reference Printer is locked by using Mutex in computer of host printer.

Function name	EiUnlockPrinter		
Argument name	IN/OUT	Type	Description
i_hdl	I	int	Handle
Return value	INT		
･Error (negative value), Normal end (0)			
Processing Description			

･Unlock printer

Error Description

-60 : Handle value is invalid

-340 : Spooler does not run

-1000 : Unable to execute this function because printer is locked

Win32 API Reference

Function name: EiDirectIO

Argument name	IN/OUT	Type	Description
i_hdl	I	int	Handle
i_wlen	I	BYTE	Output size
i_wcmd	I	PBYTE	Output command
io_rlen	IO	PBYTE	Read size
o_rbuf	O	PBYTE	received data
i_timeout	I	DWORD	Timeout period
i_nt	I	BOOL	receiving flag

Return value: INT

- Error (negative value), Normal end (0)

Processing Description

- Send/Receive binary data

- Argument detailed description

Read size: Specify the data length acquired from printer
Specify this argument if the command execution result from printer is necessary. The acquired buffer size is returned.
Specify “0” if it is not necessary to acquire the data.
Timeout period: Specify by msec (millisecond)
receiving flag: TRUE = until when something is received
FALSE = until either read size is received or timeout

Error Description
-60 : Handle value is invalid
-70 : Timeout error
-80 : Unable to R/W to printer
-340 : Spooler does not run
-1000 : Unable to execute this function because printer is locked

Function name	EiDirectIOEx		
Argument name	IN/OUT	Type	Description
i_hdl	I	int	Handle
i_wlen	I	DWORD	Output size
i_wcmd	I	PBYTE	Output command
io_rlen	IO	PDWORD	Read size
o_rbuf	O	PBYTE	received data
i_timeout	I	DWORD	Timeout period
i_nt	I	BOOL	receiving flag
i_op	I	BYTE	Receiving target
Return value	INT		
･Error (negative value), Normal end (0)			

Processing Description

·Send/Receive binary data

·Argument detailed description

Read size: Specify the data length acquired from printer
Specify this argument if the command execution result from printer is necessary. The acquired buffer size is returned.
Specify “0” if it is not necessary to acquire the data.

Timeout period: Specify by msec (millisecond)

receiving flag: TRUE = until when something is received
FALSE = until either read size is received or timeout

Receiving target: 0 = All received data is targeted.
1 = Exempt Status from target

Error Description

-60 : Handle value is invalid
-70 : Timeout error
-80 : Unable to R/W to printer
-340 : Spooler does not run
-1000 : Unable to execute this function because printer is locked

Function name	EiResetPrinter

Argument name	IN/OUT	Type	Description
i_hdl	I	int	Handle

Return value	INT

・Error (negative value), Normal end (0)

Processing Description

·Reset printer

Error Description

-60 : Handle value is invalid

-340 : Spooler does not run

-1000 : Unable to execute this function because printer is locked

*This function is not available on Windows 2000.

Function name	EiGetStatus

Argument name	IN/OUT	Type	Description
i_hdl	I	int	Handle
o_status	O	PDWORD	Status

Return value	INT

・Error (negative value), Normal end (0)

Processing Description

·Get printer status

Error Description

-60 : Handle value is invalid

*Please refer to “Status list" for status description.

Function name	EiSetStatusBackFunction		
Argument name	IN/OUT	Type	Description
i_hdl i_stcb	I I	int Please see as follows	Handle Callback function address
Return value	INT		
･Error (negative value), Normal end (0)			
Processing Description			

·Register callback function called for auto-status notice

·Argument detailed description

Callback function address: Application defined callback function to receive status

int CALLBACK StatusCB(DWORD st)

Error Description

-60 : Handle value is invalid

Function name	EiSetStatusBackFunctionEx

Argument name	IN/OUT	Type	Description
i_hdl	I	int	Handle
i_stcb	I	Please see the following	Callback function address

Return value	INT

・Error (negative value), Normal end (0)

Processing Description

·Register callback function called for auto-status notice
(Enables to get a port name by callback function)

·Argument detailed description
Callback function address: Application defined callback function to receive status
`int CALLBACK StatusCBEx(DWORD st,LPSTR pname)`

Error Description
-60 : Handle value is invalid

Function name	EiSetStatusBackWnd		
Argument name	IN/OUT	Type	Description
i_hdl	I	int	Handle
i_Wnd	I	HANDLE	Windows handle
i_status	I	PDWORD	Status saving address
Return value	INT		
・Error (negative value), Normal end (0)			
Processing Description			

·Register window handle of button to send click event when auto status notice and memory address to set status information

·Argument detailed description
Status saving address : Save status at the address specified
when issueing click event.

Error Description
-60 : Handle value is invalid

Function name	EiCancelStatusBack		
Argument name	IN/OUT	Type	Description
i_hdl	I	Int	Handle
Return value	INT		
・Error (negative value), Normal end (0)			
Processing Description			

·Cancel auto status notice request process
Error Description
-60 : Handle value is invalid

Function name	EiPowerOff		
Argument name	IN/OUT	Type	Description
i_hdl	I	int	Handle
Return value	INT		
･Error (negative value), Normal end (0)			

Processing Description

·Power off printer

Error Description

-60 : Handle value is invalid
-70 ： Timeout error
-80 : Unable to R/W to printer
-1000 : Unable to execute this function because printer is locked

Function name	EiGetCounter		
Argument name	IN/OUT	Type	Description
i_hdl	I	int	Handle
i_readno	I	WORD	Acquisition counter ID
o_readcounter	O	PDWORD	Counter stored area
Return value	INT		
･Error (negative value), Normal end (0)			

Processing Description

·Get maintenance counter

·Argument detailed description

Acquisition counter ID :

14H:	Number of travelling dot line	(User maintenance counter)
15H:	Number of head energizing dot line	(User maintenance counter)
32H:	Number of cut	(User maintenance counter)
94H:	Number of travelling dot line	
95H:	Number of head energizing dot line	
B2H:	Number of cut	

Error Description

-60 : Handle value is invalid
-70 : Timeout error
-80 : Unable to R/W to printer
-100 : Unsupported model
-1000 : Unable to execute this function because printer is locked

Function name	EiResetCounter		
Argument name	IN/OUT	Type	Description
i_hdl i_readno	I I	int WORD	Handle Acquisition counter ID
Return value	INT		
・Error (negative value), Normal end (0)			
Processing Description			

·Reset maintenance counter

·Argument detailed description

Acquisition counter ID:

14H: Number of travelling dot line

15H: Number of head energizing dot line

32H: Number of cut

Error Description

-60 : Handle value is invalid

-70 : Timeout error

-80 : Unable to R/W to printer

-100 : Unsupported model

-1000 : Unable to execute this function because printer is locked

Function name	EiGetPrnCapability		
Argument name	IN/OUT	Type	Description
i_hdl	I	int	Handle
i_id	I	BYTE	Type ID (Refer to Extended Information)
io_datsize	IO	PBYTE	Data acquired size
o_dat	O	PBYTE	Data acquired area (NULL can be specified)
Return value	INT		
･Error (negative value), Normal end (0)			
Processing Description			

·Get printer information

·Argument detailed description

Data acquired area : End after setting data size to data acquired size when NULL is specified.

Error Description

-60 : Handle value is invalid

-70 ： Timeout error

-80 : Unable to R/W to printer

-1000 : Unable to execute this function because printer is locked

Extended Information

Please refer to the item of **“Extended Information”** of Setting guide regarding information that you can get by EiGetPrnCapabilit.
Except, you may get different information depending on printer version.

Function name	EiOpenDrawer		
Argument name	**IN/OUT**	**Type**	**Description**
i_hdl	I	int	Handle
i_drawer	I	BYTE	Target drawer
i_pulse	I	BYTE	Drawer kick period
Return value	INT		
･Error (negative value), Normal end (0)			

Processing Description

·Open drawer

·Argument detailed description

Target drawer: 1 = Open Drawer 1
2 = Open Drawer 2

Draw kick period: 1 = Activate drawer after 100 msec
2 = Activate drawer after 200 msec
3 = Activate drawer after 300 msec
4 = Activate drawer after 400 msec
5 = Activate drawer after 500 msec

Error Description
-60 : Handle value is invalid
-70 : Timeout error
-80 : Unable to R/W to printer
-90 : Parameter error
-100 : Unsupported model
-1000 : Unable to execute this function because printer is locked

Function name	EmuDM_GetProperty		
Argument name	IN/OUT	Type	Description
i_devmode i_pid o_dat io_size	I I O IO	PDEVMODEA BYTE PBYTE PDWORD	DevMode address Property ID Data save address Data size
Return value	INT		

・Error (negative value), Normal end (0)

Processing Description

·Returns property status set by the specified DevMode.

·Argument detailed description

Property ID : 0 = All Image
Property ID : 1 = Printer Initialization
Property ID : 2 = Print speed
Property ID : 3 = Margin
Property ID : 4 = Density
Property ID : 5 = Watermark
Property ID : 6 = Scale
Property ID : 7 = Rotation
Property ID : 8 = Disable setting
Property ID : 9 = Color print mode

Property ID : 10 = Print notice setting at printing start
Property ID : 11 = Cutter operation at printing start
Property ID : 12 = Drawer at printing start
Property ID : 13 = Drawer kick ON TIME at printing start
Property ID : 14 = Drawer kick OFF TIME at printing start
Property ID : 15 = Custom command at printing start
Property ID : 16 = Back feed at printing start
Property ID : 17 = Feed at printing start
Property ID : 19 = Move to the cutting position at printing start

Please refer to "property value" for each property value.

Function name	EmuDM_GetProperty
Processing Description	

·Argument detailed description

Property ID：21 = Cutter operation at page start
Property ID：22 = Drawer at page start
Property ID：23 = Drawer kick ON TIME at page start
Property ID：24 = Drawer kick OFF TIME at page start
Property ID：25 = Custom command at page start
Property ID：26 = Back feed at page start
Property ID：27 = Feed at page start
Property ID：29 = Move to cutting position at page start

Property ID：31 = Cutting operation at page end
Property ID：32 = Drawer at page end
Property ID：32 = Drawer kick ON TIME at page end
Property ID：34 = Drawer kick OFF TIME at page end
Property ID：35 = Custom command at page end
Property ID：36 = Back feed at page end
Property ID：37 = Feed at page end
Property ID：39 = Move to cutting position at page end

Property ID：40 = Print notice setting at print end
Property ID：41 = Cutter operation at print end
Property ID：42 = Drawer at print end
Property ID：43 = Drawer kick ON TIME at print end
Property ID：44 = Drawer kick OFF TIME at print end
Property ID：45 = Custom command at print end
Property ID：46 = Back feed at print end
Property ID：47 = Feed at print end
Property ID：49 = Move to cutting position at print end

Error Description

-90 ：Parameter error

* This function is not available for Windows 2000.

Function name	EmuDM_SetProperty		
Argument name	IN/OUT	Type	Description
i_devmode	I	PDEVMODEA	DevMode address
i_pid	I	BYTE	Property ID
i_dat	I	PBYTE	Setting data address
i_size	I	DWORD	Data size
Return value	INT		
･Error (negative value), Normal end (0)			

Processing Description

･Set data specified at specified DevMode

･Argument detailed description

Property ID： 0 = All Image
Property ID： 1 = Printer Initialization
Property ID： 2 = Print speed
Property ID： 3 =Margin
Property ID： 4 =Density
Property ID： 5 =Watermark
Property ID： 6 = Scale
Property ID： 7 = Rotation
Property ID： 8 = Disable setting
Property ID： 9 = Color print mode

Property ID：10 = Print notice setting at printing start
Property ID：11 = Cutter operation at printing start
Property ID：12 = Drawer at printing start
Property ID：13 = Drawer kick ON TIME at printing start
Property ID：14 = Drawer kick OFF TIME at printing start
Property ID：15 = Custom command at printing start
Property ID：16 = Back feed at printing start
Property ID：17 = Feed at printing start
Property ID：19 = Move to the cutting position at printing start

Property ID：21 = Cutter operation at page start
Property ID：22 = Drawer at page start
Property ID：23 = Drawer kick ON TIME at page start
Property ID：24 = Drawer kick OFF TIME at page start
Property ID：25 = Custom command at page start

Please refer to "property value" for each property value.

Function name	EmuDM_SetProperty
Processing Description	

·Argument detailed description

Property ID：26 = Back feed at page start
Property ID：27 = Feed at page start
Property ID：29 = Move to cutting position at page start

Property ID：31 = Cutting operation at page end
Property ID：32 = Drawer at page end
Property ID：32 = Drawer kick ON TIME at page end
Property ID：34 = Drawer kick OFF TIME at page end
Property ID：35 = Custom command at page end
Property ID：36 = Back feed at page end
Property ID：37 = Feed at page end
Property ID：39 = Move to cutting position at page end

Property ID：40 = Print notice setting at print end
Property ID：41 = Cutter operation at print end
Property ID：42 = Drawer at print end
Property ID：43 = Drawer kick ON TIME at print end
Property ID：44 = Drawer kick OFF TIME at print end
Property ID：45 = Custom command at print end
Property ID：46 = Back feed at print end
Property ID：47 = Feed at print end
Property ID：49 = Move to cutting position at print end

Property ID：50 = Paper size

Property ID：60 = 58mm POS setting
Property ID：61 = 80mm POS setting
Property ID：62 = 58mm Ticket setting
Property ID：63 = 80mm Ticket setting
Property ID：66 = A4-58mm reduced printing
Property ID：67 = A4-80mm reduced printing
Property ID：68 = Return to default

・Error Description

-90 ：Parameter error

* This function is not available for Windows 2000/XP up to and including Service Pack2.

Win32 API Reference

Property value

Property ID： 0 = All Image 0：False 1 ：True

Property ID： 1 = Printer Initialization 0:Printer Initialization
1：None(each font setting initialization)

Property ID： 2 = Print speed 0ー4

Property ID： 3 = Margin 0：Standard 1: Saving

Property ID： 4 = Density 0：Standard 1： Dark 2： Light

Property ID： 5 = Watermark 0: None 1: Upper left 2: Top center 3: Upper right
4: Center left 5: Center 6: Center right
7: Lower left 8: Center bottom 9: Lower right

Property ID： 6 = Scale(%) 25-100

Property ID： 7 = Rotation 0: 0° 1: 90° 2: 180° 3: 270°

Property ID： 8 = Setting is disabled 0: False 1：TRUE

Property ID： 9 = Color print mode 0: Black and white 1: Gray Scale(halftone)

Property ID：10/40 = Print notice setting 0: None 1: Flag 2: Data hold
3：Flag + Notice
4：Flag + Notice + Data hold

Property ID：11/21/31/41 = Cutter operation 0: None 1: Partial cut
2： Full cut

Property ID：12/22/32/42 = Drawer 0：None 1: 2PIN 2: 5PIN

Property ID：13/23/33/43 = Drawer kick ON TIME 1-255

Property ID：14/24/34/44 = Drawer kick OFF TIME 1-255

Property ID：15/25/35/45 = Custom command (100BYTE) Specifying Hex character

Property ID：16/26/36/46 = Back feed 0-255

Property ID：17/27/37/47 = Feed 0-255

Property ID：19/29/39/49 = Feed to cut position 0: Does not feed to cut position
1: Feed to cut position

Property dedicated for setting

Property ID：50 = Paper size 0：Letter 1：A4 2：A4 Extra
3：80mm 4：58mm 5：PostCard
Others: Paper number created by user

Property ID：60 = 58mm POS setting, 61 = 80mm POS setting,
62 = 58mm Ticket setting, 63 = 80mm Ticket setting,
66 = A4-58mm reduced printing, 67 = A4-80mm reduced printing,
68 = Return to default

The setting data is ignored when specifying the above values.

Function name	EmuBarcode		
Argument name	IN/OUT	Type	Description
i_hdl	I	int	Handle
i_fname	I	PCHAR	Font name
i_bmp	IO	HDC	Device context handle
i_x	I	DWORD	X
i_y	I	DWORD	Y
i_width	I	DWORD	Width
i_height	I	DWORD	Height
i_dat	I	PBYTE	Barcode data
i_size	I	DWORD	Data size
Return value	INT		
･Error (negative value), Normal end (0)			

Processing Description

･Draw the barcode/2D barcode specified at the barcode font/2D barcode font setting of printer on the device context.

Error Description

-50 : Memory shortage

-60 : Handle value is invalid

-90 : Parameter error

.net API Reference

Class description

Name space: NEmuPrinterAPIC
Class name: NEmuPrinterAPIClass

Type	Name	Remarks
Property	Status	Final status
Property	LastError	The last executed API error code
Type	**Name**	**Remarks**
Method	OpenMonPrinter	Win32 API wrapper
Method	CloseMonPrinter	Win32 API wrapper
Method	LockPrinter	Win32 API wrapper
Method	UnlockPrinter	Win32 API wrapper
Method	DirectIOEx	Win32 API wrapper
Method	ResetPrinter	Win32 API wrapper
Method	SetStatusBack	Win32 API wrapper
Method	CancelStatusBack	Win32 API wrapper
Method	PowerOff	Win32 API wrapper
Method	GetCounter	Win32 API wrapper
Method	ResetCounter	Win32 API wrapper
Method	GetPrnCapability	Win32 API wrapper
Method	OpenDrawer	Win32 API wrapper
Method	DM_GetProperty	Win32 API wrapper
Method	DM_SetProperty	Win32 API wrapper
Method	Barcode	Win32 API wrapper
Event	StatusCallback	Event to process the notified status
Event	StatusCallbackEx	Event to process the notified status

Property

Property Name	LastError
Type	uint
Processing Description	Get ·The return value of the last executed API Set ·-
Remark	
·Default value is 0	

Property Name	Status
Type	long
Processing Description	Get ·Get the latest printer status Set ·-
Remark	
·Default value is 0	

.net API Reference

Method

Function name	OpenMonPrinter		
Argument name	IN/OUT	Type	Description
i_type i_prt	I I	int [MarshalAs(UnmanagedType.LPStr)] string	2（Fixed） Output printer driver name
Return value	int		
Processing Description			
Remark			

Please refer to EiOpenMonPrinter “Win32 API reference” for Return value, Processing Description

Function name	CloseMonPrinter		
Argument name	IN/OUT	Type	Description
Return value	int		
Processing Description			
Remark			

Please refer to EiCloseMonPrinter “Win32 API reference” for Return value, Processing Description

Function name	LockPrinter		
Argument name	IN/OUT	Type	Description
i_timeout	I	long	Timeout period
Return value	int		
Processing Description			
Remark			

Please refer to EiLockPrinter "Win32 API reference" for Return value, Processing Description

Function name	UnlockPrinter		
Argument name	IN/OUT	Type	Description
Return value	int		
Processing Description			
Remark			
·Use this function after setting PrinterName propetry			

Please refer to EiUnlockPrinter "Win32 API reference" for Return value, Processing Description

.net API Reference

Function name	DirectIOEx		
Argument name	IN/OUT	Type	Description
i_wlen i_wcmd io_rlen o_rbuf i_timeout i_nt i_op	I I IO O I I I	long byte[] out long out byte[] long byte byte	Output size Output command Read size received data Timeout receiving flag Receiving target
Return value	int		
Processing Description			
Remark			

Please refer to EiDirectIOEx "Win32 API reference" for Return value, Processing Description.

Function name	DirectIOEx		
Argument name	IN/OUT	Type	Description
i_wlen i_wcmd i_timeout	I I I	long byte[] long	Output size Output command Timeout
Return value	int		
Processing Description			
Remark			

Please refer to EiDirectIOEx "Win32 API reference" for Return value, Processing Description.

Function name	ResetPrinter		
Argument name	IN/OUT	Type	Description
Return value	int		
Processing Description			
Remark			

Please refer to EiResetPrinter "Win32 API reference" for Return value, Processing Description.

Function name	SetStatusBack		
Argument name	IN/OUT	Type	Description
Return value	int		
Processing Description			
Remark			

Please refer to EiSetStatusBackFunctionEx "Win32 API reference" for Return value, Processing Description.

Function name	CancelStatusBack		
Argument name	IN/OUT	Type	Description
Return value	int		
Processing Description			
Remark			

Please refer to EiCancelStatusBack "Win32 API reference" for Return value, Processing Description.

Function name	PowerOff		
Argument name	IN/OUT	Type	Description
Return value	int		
Processing Description			
Remark			

Please refer to EiPowerOff "Win32 API reference" for Return value, Processing Description.

Function name	GetCounter		
Argument name	IN/OUT	Type	Description
i_readno	I	short	Acquisition counter ID
o_readcounter	O	out long	Counter stored area
Return value	int		
Processing Description			
Remark			

Please refer to EiGetCounter "Win32 API reference" for Return value, Processing Description.

Function name	ResetCounter		
Argument name	IN/OUT	Type	Description
i_readno	I	short	Acquisition counter ID
Return value	int		
Processing Description			
Remark			

Please refer to EiResetCounter "Win32 API reference" for Return value, Processing Description.

Function name	GetPrnCapability		
Argument name	IN/OUT	Type	Description
i_id	I	byte	Type ID (Refer to Extended Information)
io_datsize	IO	ref byte[]	Data acquisition size
o_dat	O	byte[]	Data stored area (Enables to acquire NULL)
Return value	int		
Processing Description			
Remark			

Please refer to EiGetPrnCapability "Win32 API reference" for Return value, Processing Description.

Function name	OpenDrawer		
Argument name	IN/OUT	Type	Description
i_drawer	I	byte	Target drawer
i_pulse	I	byte	Drawer kick period
Return value	int		
Processing Description			
Remark			

Please refer to EiOpenDrawer "Win32 API reference" for Return value, Processing Description.

Function name	DM_GetProperty		
Argument name	IN/OUT	Type	Description
i_devmode	I	IntPtr	DevMode address
i_pid	I	byte	Property ID
o_dat	O	byte[]	Data save address
io_size	IO	ref long	Data size
Return value	int		
Processing Description			
Remark			

Please refer to EmuDM_GetProperty "Win32 API reference" for Return value, Processing Description.

Function name	DM_SetProperty		
Argument name	IN/OUT	Type	Description
i_devmode	I	IntPtr	DevMode address
i_pid	I	byte	Property ID
i_dat	I	byte[]	Setting data address
i_size	I	long	Data size
Return value	int		
Processing Description			
Remark			

Please refer to EmuDM_SetProperty "Win32 API reference" for Return value, Processing Description.

.net API Reference

Function name	Barcode		
Argument name	IN/OUT	Type	Description
i_fname	I	[MarshalAs(UnmanagedType.LPStr)] string	Font name
i_bmp	IO	IntPtr	Device context handle
i_x	I	long	X
i_y	I	long	Y
i_width	I	long	Width
i_height	I	long	Height
i_dat	I	byte[]	Barcode data
i_size	I	long	Data size
Return value	int		
Processing Description			
Remark			

Please refer to EmuBarcode "Win32 API reference" for Return value, Processing Description.

Event

Function name	StatusCallback		
Argument name	IN/OUT	Type	Description
i_status	I	long	Status
Return value	int		
Processing Description			
Remark			

Please refer to EiSetStatusBackFunction "Win32 API reference" for Return value, Processing Description.

Function name	StatusCallbackEx		
Argument name	IN/OUT	Type	Description
i_status	I	long	Status
i_port	I	string	Port Name
Return value	int		
Processing Description			
Remark			

Please refer to EiSetStatusBackFunctionEx "Win32 API reference" for Return value, Processing Description.

Status List

Emu Status	Value
ESB_NO_RESPONSE	0x00000001 Printer no response
	0x00000000 Printer response
ESB_DRAWER_KICK	0x00000004 Drawer kick connector status of 3rd pin =”H”
	0x00000000 Drawer kick connector status of 3rd pin =“L”
ESB_OFF_LINE	0x00000008 Offline
	0x00000000 Online
ESB_COVER_OPEN	0x00000020 Cover open
	0x00000000 Cover close
ESB_AUTOCUTTER_ERR	0x00000800 Auto cutter error occurred
	0x00000000 Auto cutter error not occurred
ESB_AUTORECOVER_ERR	0x00004000 Auto recovery error occurred
	0x00000000 Auto recovery error not occurred
ESB_RECEIPT_NEAR_END	0x00020000 No paper at receipt near end detector
	0x00000000 Paper at receipt near end detector
ESB_RECEIPT_END	0x00080000 No paper at receipt end detector
	0x00000000 Paper at receipt end detector
ESB_SPOOLER_IS_STOPPED	0x80000000 Stop spooler
	0x00000000 -
ESB_NOW_PRINTING	0x00000080 Printing
	0x00000000 -