

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 suspicious points, errors, omissions, etc., please contact us.
- Descriptions are subject to change without prior notice.
Please ask for up-to-date information.
- Reprinting, copying, duplicating, or falsifying part or all of the contents of this document without permission is strictly prohibited.
- Please note that we do not take any responsibility for the effects caused by the results of use.
- We assume no responsibility whatsoever for any damages resulting from improper use, the use without understanding of descriptions in this document or the repair and the change by the third party.

Trademark

MS-DOS®, Microsoft®, Win32®, Windows®, Windows Vista®, Visual Studio®, Visual Basic®, Visual C++®, Visual C#® are trademark or registered trademark of Microsoft Corporation in the U.S.A. and other countries.

Signs

Please understand the meaning of each symbol before handling the product.

 Caution	It indicates the precautions to be observed fully when using the product. Disregard of the signs and wrong usage may cause product failure and malfunction.
 Reference	It indicates supplementary descriptions 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, etc. directly on conveyance including aircraft, train, ship or vehicle; etc., disaster prevention and security devices, etc., users should use products after considering the safety design of the whole system by applying fail-safe or redundancy design in order to maintain reliability and safety.

This product is not designed to use with devices that require extremely high reliability and safety including aerospace mechanism, devices for trunk communication, nuclear control device or medical services. Users should ascertain and evaluate suitability of these products for those applications.

Table of contents

- Overview
 - [System configuration with using SDK guide](#) 6
- Various settings
 - [Install](#) 7
 - [Regarding firewall](#) 8
 - [How to fix the IP address](#) 10
 - [How to use this SDK guide, Regarding driver connection](#)
[Regarding Bluetooth connection, Regarding serial connection](#)... 11
 - [Notes for direct communication](#) 11
 - [Storage location for temporary files](#) 11
 - [Regarding log output](#) 12
 - [Regarding barcode setting file](#) 15
- Win32 API Reference
 - [Category of API](#) 17
 - Function
 - [NCallback \(Interface\)](#) 19
 - [NSetCallback](#) 21
 - [NEnumPrinters / NEnumPrintersA](#) 22
 - [NGetPrinterFromID / NGetPrinterFromIDA](#)..... 24
 - [NDeletePrinter / NDeletePrinterA](#) 25
 - [NRenamePrinter / NRenamePrinterA](#) 26
 - [NGetPrinterInf / NGetPrinterInfA](#) 27
 - [NAutoOpen](#) 28
 - [NOpenPrinter / NOpenPrinterA](#) 29
 - [NClosePrinter / NClosePrinterA](#) 30
 - [NClosePrinters](#) 31
 - [Nprint / NPrintA](#) 32
 - [NDPrint / NDPrintA](#) 33
 - [NImagePrint / NImagePrintA](#)..... 34
 - [NImagePrintF / NImagePrintFA](#) 35
 - [NGetStatus / NGetStatusA](#) 36
 - [NGetInformation](#) 37
 - [Extended Information NGetInformationA](#) 38
 - [NResetPrinter NResetPrinterA](#) 40
 - [NStartDoc / NStartDocA](#) 41

– NEndDoc / NEndDocA	42
– NCancelDoc / NCancelDocA	43
– NEnumDoc / NEnumDocA	44
– NDeleteDoc / NEnumDocA	45
– NBarcodeSettings	46
– Barcode setting items list	47
– Options list	48
– NBarcode / NBarcodeA	53
– NBarcode2	55
– NFirmwareDL / NFirmwareDLA	57
– NInitializeNetwork	58
– NScanPrinters	59
– NTCPPortLock / NTCPPortLockA	61
– NBufferClear / NBufferClearA	62
– NBlockSendSetting / NBlockSendSettingA	63
– Error codes table	64
– List of error codes by functions	66

▪ .net API Reference

– Class description	72
Method	
– NEnumPrinters	74
– NGetPrinterFromID	74
– NDeletePrinter	75
– NRenamePrinter	75
– NGetPrinterInf	76
– NAutoOpen	76
– NOpenPrinter	77
– NClosePrinter	77
– NClosePrinters	78
– NPrint	78
– NDPrint	79
– NImagePrint	80
– NImagePrintF	81
– NGetStatus	81
– NGetInformation	82
– NResetPrinter	82

– NStartDoc	83
– NEndDoc	83
– NCancelDoc	84
– NEnumDoc	84
– NDeleteDoc	85
– NBarcodeSettings	85
– NBarcode	86
– NBarcode2	86
– NFirmwareDL	87
– NScanPrinters	87
– NTCPPortLock	88
– NBufferClear	88
– NBlockSendSetting	89
– NSetCallback	89
– NInitializeNetwork	90

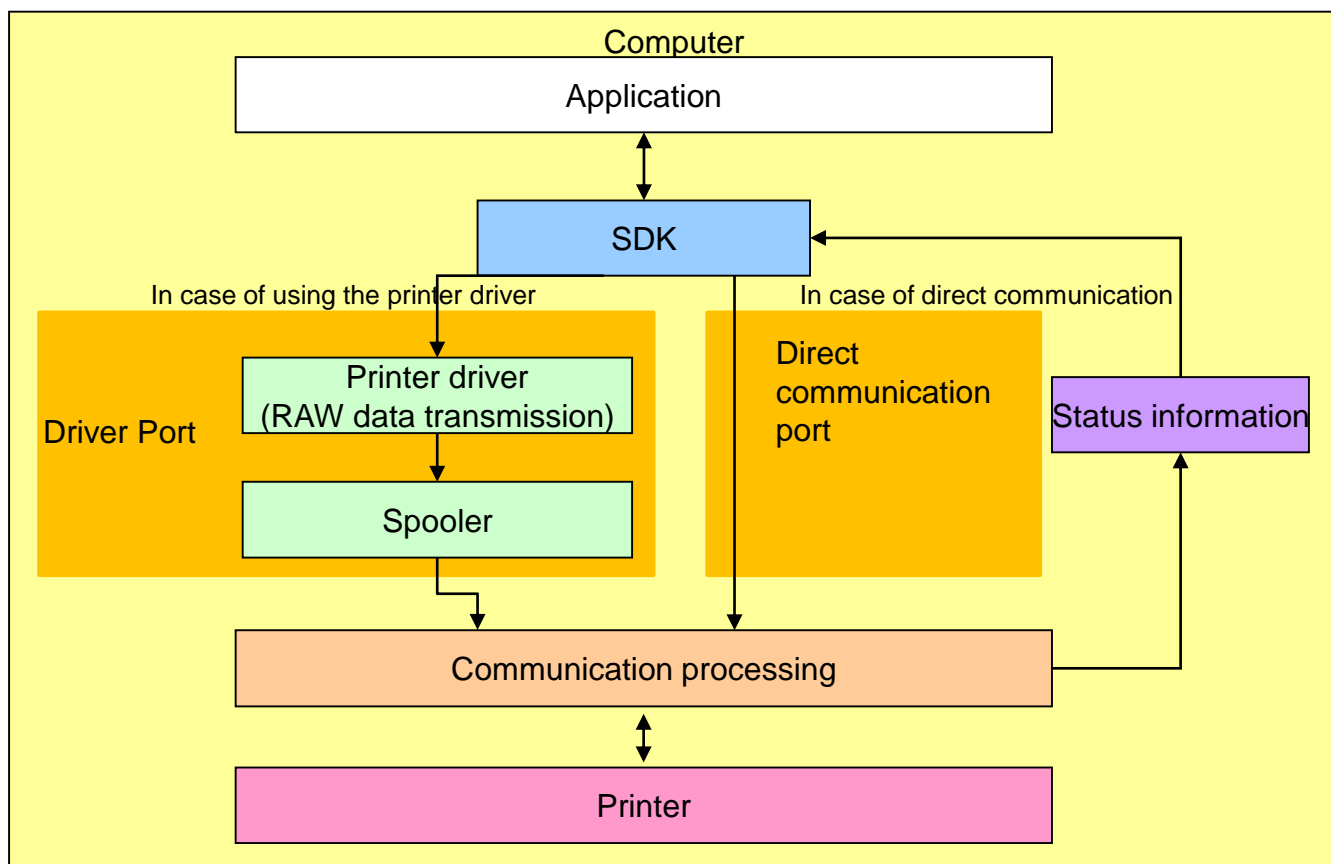
Overview

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

Files provided as SDK are distributed as Dynamic Link Library (DLL) and various executable files (EXE).

File list : NPrinterLib.dll, BarCodeLib.dll, NBarcodeLib.dll, NPrinterCLib.dll, NServiceDrv.exe

System configuration when using SDK



Development language

Win32 : Visual C++

.NET : Visual C#

Interface

USB, RS232C, Bluetooth (2.1+EDR), TCP/UDP

Install

When using only this library

1. Please save "NPrinterLib.dll" and "NPrinterCLib.dll" in a folder that can be referenced by the application which will be used.

Please save "NBarcodeLib.dll" in System32 folder.

When using the SDK for 32bit under 64bit environment, please save "NBarcodeLib.dll" for 32bit in SysWOW64 folder.

When using C# applications, please use "NPrinterCLib.dll" after referring to it and setting it.

2. Please make NServiceDrv.exe service registration.

<32bit environment>

- Please save the file in System32 folder.
- Please start the command prompt by administrator rights and execute the following command.

```
sc create NServiceDrv binpath= "C:¥Windows¥system32¥NServiceDrv.exe"
```

<64bit environment>

- Please save the file in SysWOW64 folder.
- Please start the command prompt by administrator rights and execute the following command.

```
sc create NServiceDrv binpath= "C:¥Windows¥SysWOW64¥NServiceDrv.exe"
```

3. Please start the registered service.

- Starting Windows [Administrative Tools] – [Services]
- Changing NserviceDrv property [Type of Start-up] into [Automatic]
- Executing [Start of the service]

When using this library with driver

No special settings for this library are required because this library is automatically installed when you set up the driver. For driver setup, please refer to "Installation Guide" of the driver.

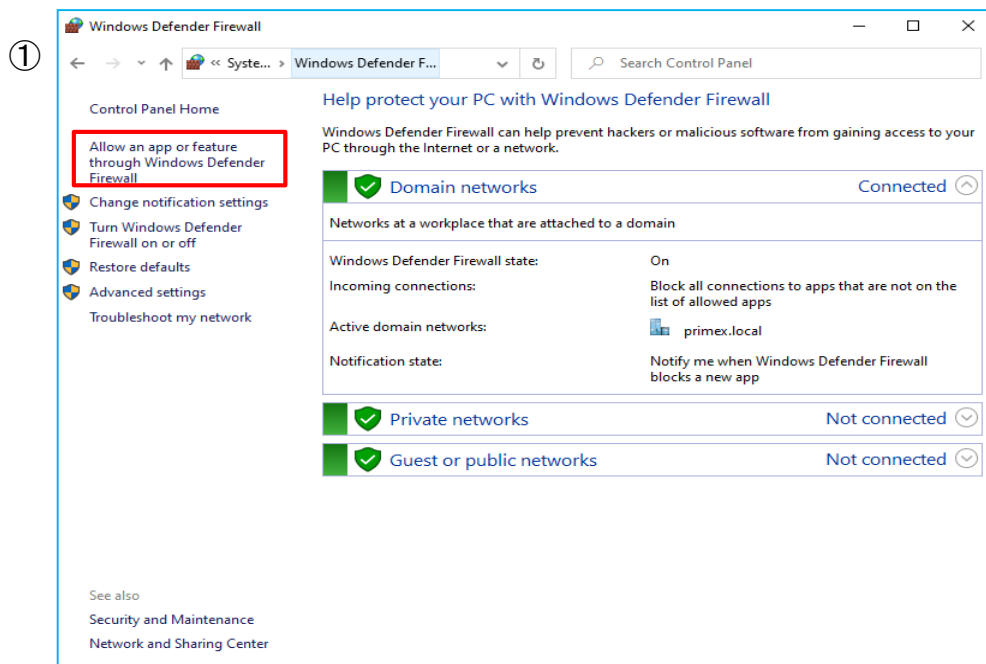
Reference

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

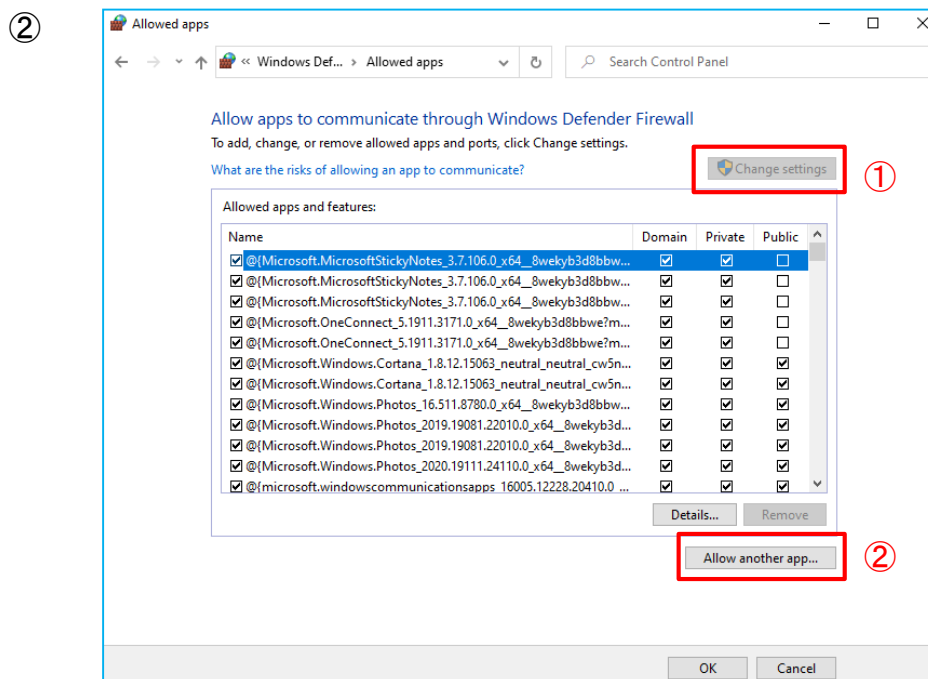
Regarding firewall (1/2)

- Since NServiceDrv.exe is a program for UDP communication, firewall authorization is required before using it.

* This is not needed when USB, serial or Bluetooth is used.

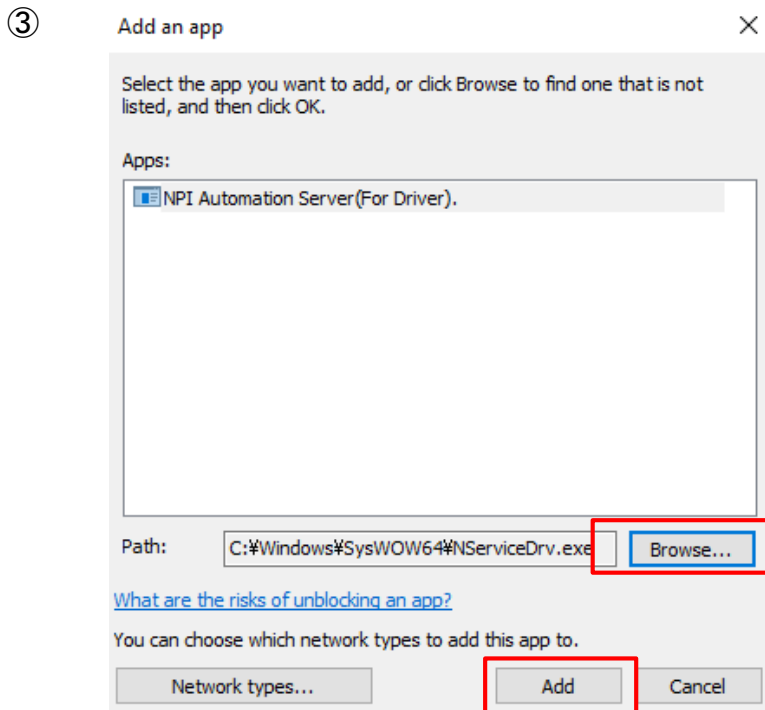


Please select "Control Panel"->"Firewall"->"Allow apps or features".

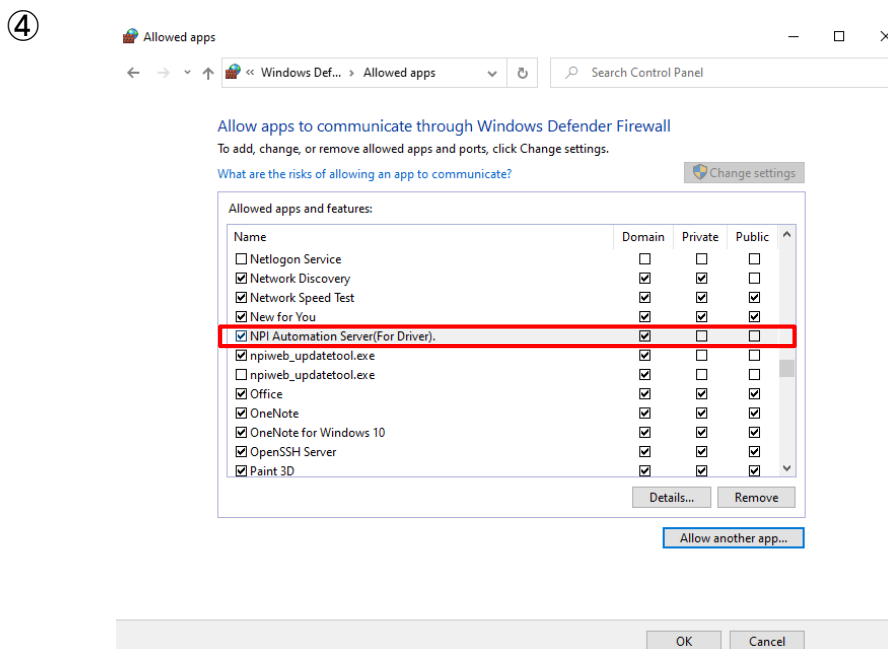


Press the button in the order of "Change settings" - "Allow another app..."

Regarding firewall (2/2)



Please specify the path to “NServiceDrv” in “Browse...” and execute application permission by “Add” button.



Please check whether “NPI Automation Server(For Driver)” is added and input the checkmark in the checkbox

How to fix the IP address

This SDK automatically acquire IP address set in the PC in order to perform broadcasting. However, if there are multiple network adapters or multiple network settings depending on the virtual environment, there is a risk that an IP address other than the network adapter used for communication with the printer may be acquired .

To specify the IP address to be acquired by SDK, it is necessary to create the IPAddress.inf file directly under the NPI folder and describe the IP address set in the PC.



192.168.0.1

IPAddress.inf

* Please describe only IP address.

Reference

Broadcasting is used when printers detection is performed by NScanPrinters function.

How to use this SDK guide

After installation is completed, please include ImportApi.h in the development environment.

Please load the NPrinterLib.dll by the LoadLibrary function after that.

If the function succeeds, the DLL's handle is returned. The address of the DLL function can be obtained by passing that handle to the GetProcAddress function.

For details, please refer to the LoadLibrary execution part of the sample program (Document/API_Sample) in the release media. ImportApi.h is included in the sample.

Regarding driver connection

It is not necessary to call NEnumPrinters or NOpenPrinter when using the driver port.

Regarding Bluetooth connection

In this SDK, it is necessary to pair the PC and the printer before making a Bluetooth connection.

Regarding serial connection

In this SDK, it is necessary to complete setting port to be used before making a serial connection. Data cannot be sent and received properly when the port setting of the printer does not match the port setting of the PC.

Notes for direct communication

Please DO NOT set up the driver regarding the port to be used for direct communication.

Storage location for temporary files

By default, an NPI folder is created directly under the OS setup drive.

To change the storage location, create the following key in the registry, create DirPath as a character string value under it, and specify the folder to be used.

HKEY_LOCAL_MACHINE¥SOFTWARE¥NPI

Example:

[HKEY_LOCAL_MACHINE¥SOFTWARE¥NPI]

"DirPath"="d:¥¥Printer"

Since the above registry may not be accessed when using SDK for 32bit is used under 64bit environment, please describe "DirPath" in the

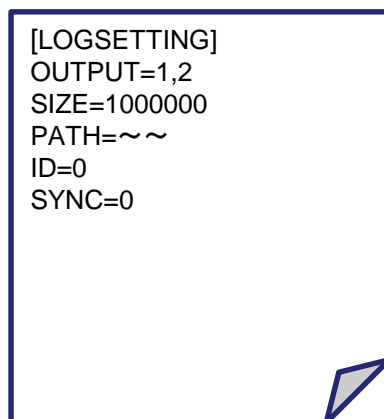
[HKEY_LOCAL_MACHINE¥SOFTWARE¥WOW6432Node¥NPI] folder.

Regarding log output (output setting)

This SDK outputs the file named SDKLog.txt under the specified folder for log files.

For log output, it is necessary to create a log setting file (OS drive ¥ NPI ¥ NLogInf.inf).

(No log is output if the file does not exist)



Please Please write the configuration file in the following format.

[LOGSETTING] ← fixed

OUTPUT=1,2 ← Specifying the log types to be output separated by commas.
(In this case 1,2 : only ERROR and WARN are output.)

SIZE=1000000 ← Specifying the maximum file size of log writing.

PATH=C:¥NPI¥log ← Specifying the log writing folder

ID=0 ← Setting the output settings for the process ID and thread ID.
(0 : without output, 1 : with output)

SYNC=0 ← Specifying the timing for writing logs.
(0 : Storing log and asynchronous writing at regular intervals.)
(1 : Synchronous writing.)

Regarding log types (OUTPUT).

1 ERROR : Error

※ “N****Port ~~~~lastError” log returns the error value (Value of GetLastError) by the system call.

2 WARN : Warning

3 FUNC : Function call (except checking status and extended information)

4 IN : Port receiving data

5 OUT : Port transmitting data

6 CHK : When checking status and extended information

7 PST : When printer status value is changing, when receiving extended information

* Log will be output according to specified types.

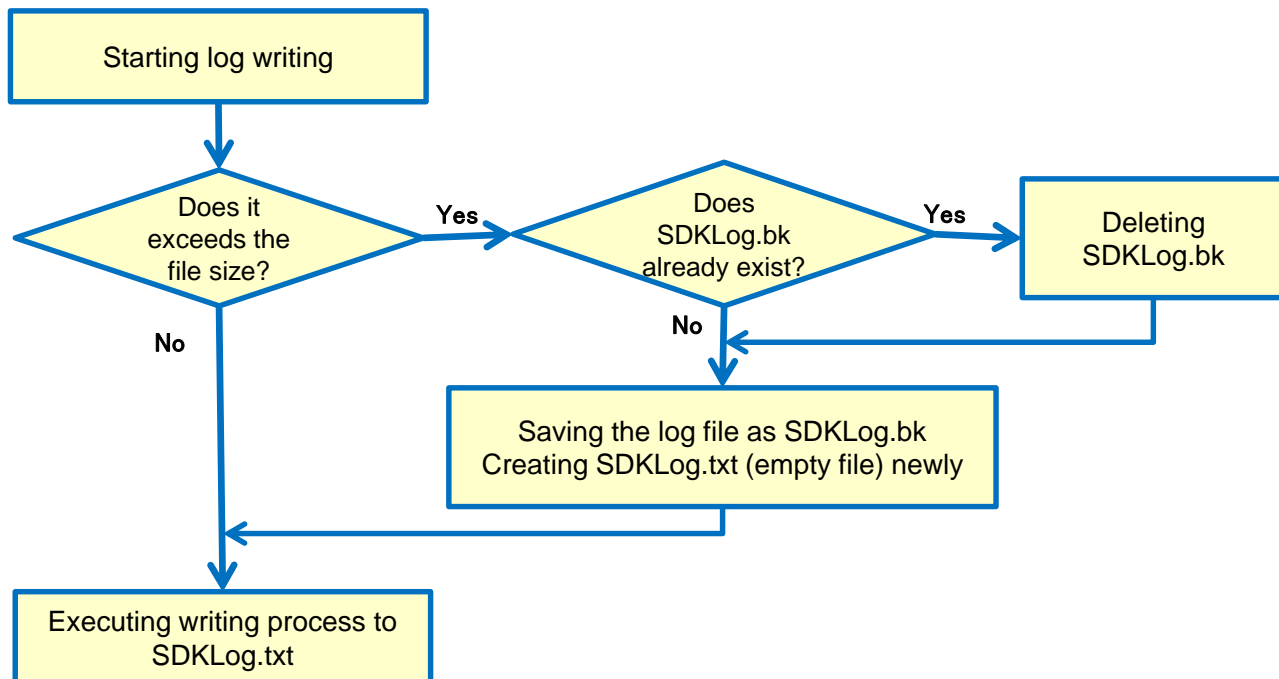
Regarding log output (notes)

- *1 Editing the file does not take effect immediately.
The reflected timing is when the NPrinterLib class is generated.
- *2 Please DO NOT put spaces (both half-width or full-width) before or after the “=”.
- *3 If nothing is set for OUTPUT, no log is output.
- *4 When an invalid log type value (numeric value other than 1 to 7 or character string) is specified, the value is disregarded.
If an invalid log size value (negative size value, character string) is specified, the default value 1MB will be used.
- *5 With respect to log types 4 (IN) and 5 (OUT), SDK operation may be slow when the amount of data transferred is large because logging takes a long time.
- *6 If you specify a large file size, the log file may not be opened. Please change the file size according to your device.
- *7 Output timing to the file at the asynchronous writing is approx. 3 seconds after it is set in the log output buffer. (For reducing the timing gap by the log output.)
- *8 When SDK is not be able to read and write the log file at the synchronous writing, the function will stop at the log writing process until reading and writing can be performed.
- *9 Please use the version 3.2.0.0 driver or later when performing the synchronous writing under the driver set up environment

Regarding log output (output file)

•If the number of bytes specified in the configuration file is exceeded, the file name is changed into SDKLog.bk and a new empty file is created.

If SDKLog.bk already exists, it will be deleted (only one log of the past log is stored).



•【Log output sample】

```
•2017/08/01 14:00:01.000 [FNC]  xxxx.
•2017/08/01 14:00:01.100 [WRN]  xxxx.
•2017/08/01 14:00:01.200 [ERR]  xxxx.
•2017/08/01 14:00:01.350 [FNC]  xxxx.
•2017/08/01 14:00:01.614 [I N]   xxxx.
•2017/08/01 14:00:02.100 [OUT]   xxxx.
•2017/08/01 14:00:03.040 [I N]   xxxx.
•2017/08/01 14:00:03.500 [CHK]   xxxx.
•2017/08/01 14:00:04.010 [PSK]   xxxx.
```

The date and time will be output in the format of YYYY/MM/DD hh:mm:ss,milliseconds.

Log types are indicated in [].

- 1 ERR : Error
- 2 WRN : Warning
- 3 FNC : Function call
- 4 I N : Port receiving data
- 5 OUT : Port transmitting data
- 6 CHK : When checking status and extended information
- 7 PSK :When printer status value is changing, when receiving extended information

If ID = 1 is set in the log setting file,
information in the form of <process ID, thread ID> will be added after the date and time.

Regarding barcode setting file

In order to print the barcode by the NBarcode function, allocating the barcode setting file is necessary.

When using the NBarcode2 function, barcode setting by using the NBarcodeSettings function instead of the barcode setting file.

1. Barcode setting file allocation

To use the “NBarcode” function with this SDK, creating a configuration file is required. This file must be created by the customer.

Barcode file name : NPIREG.inf

Note that this file refers to the following folders.

OS Drive ¥NPI¥DriverInf¥ Printer name ¥BFONT¥F0
OS Drive ¥NPI¥DriverInf¥ Printer name ¥BFONT¥F1
OS Drive ¥NPI¥DriverInf¥ Printer name ¥BFONT¥F2
OS Drive ¥NPI¥DriverInf¥ Printer name ¥BFONT¥F3
OS Drive ¥NPI¥DriverInf¥ Printer name ¥BFONT¥F4
OS Drive ¥NPI¥DriverInf¥ Printer name ¥BFONT¥F5
OS Drive ¥NPI¥DriverInf¥ Printer name ¥BFONT¥F6
OS Drive ¥NPI¥DriverInf¥ Printer name ¥BFONT¥F7
OS Drive ¥NPI¥DriverInf¥ Printer name ¥BFONT¥F8
OS Drive ¥NPI¥DriverInf¥ Printer name ¥BFONT¥F9
OS Drive ¥NPI¥DriverInf¥ Printer name ¥2DBFONT¥F0
OS Drive ¥NPI¥DriverInf¥ Printer name ¥2DBFONT¥F1
OS Drive ¥NPI¥DriverInf¥ Printer name ¥2DBFONT¥F2
OS Drive ¥NPI¥DriverInf¥ Printer name ¥2DBFONT¥F3
OS Drive ¥NPI¥DriverInf¥ Printer name ¥2DBFONT¥F4

Please create the part after OS drive part in each folder by customers.

* OS drive part is changed by the above registry setting.

Regarding barcode setting file

2. How to write a barcode setting file (1D barcode)

Set Barcode1 to Barcode10 in the following format with the following specifications.
In addition, each item is case-sensitive, so input them without mistakes.

```
"Kind"=dword:00000000
"Width"=dword:00000001
"Height"=dword:000000a2
"StartB"=dword:00000000
"StopB"=dword:00000000
"HRI"=dword:00000000
"Spin"=dword:00000000
```

Default values when there is not NPIREG.inf file in the target folder.

Kind	: 0	HRI	: 0
Width	: 1	Spin	: 0
Height	: 162		
StartB	: 0		
StopB	: 0		

Kind 0:UPC-A 1:UPC-E 2:EAN13 3:EAN8 4:CODE39 5:ITF 6:CODABAR
7:CODE128 8:CODE93

Width (data width magnification) 0=2dot 1=3dot 2=4dot **Height** data height (dots)

StartB, StopB (only CODABAR) 0:A 1:B 2:C 3:D

Spin 0:no rotation 1:90° rotation 2:180° rotation 3:270° rotation

HRI 0:None 1:Top(Font A) 2:Bottom(Font A) 3:Top&Bottom(Font A) 4:Top(Font B)
5:Bottom(Font B) 6:Top&Bottom(Font B)

3. How to write a barcode setting file (2D barcode)

Set 2D-Barcode1 to 2D-Barcode5 in the following format with the following specifications.

In addition, each item is case-sensitive. so input them without mistakes.

```
"Kind"=dword:00000000
"Size"=dword:00000000
"Height"=dword:00000001
"ECL"=dword:00000001
"ECP"=dword:00000000
"Spin"=dword:00000003
```

Default values when there is not NPIREG.inf file in the target folder.

Kind	: 0	Spin	: 0
Size	: 0		
Height	: 1		
ECL	: 1		
ECP	: 1		

Kind 0:QRCode Model2 1:PDF417

Size (data width magnification)

< QRCode Model2 > 0:double 1:triple 2:4 times 3:5 times 4:6 times 5:7times
6:8 times

< PDF417> 0:double 1:triple 2:4 times

HEIGHT (only PDF417) Set the height ration from 1 to 6. HEIGHT = height ration × WIDTH

ECL (error level)

< QRCode Model2 > 0:L 1:M 2:Q 3:H

< PDF417> 0:Auto 1:0 2:1 3:2 4:3 5:4 6:5 7:6 8:7

ECP 0:0% 1:10% 2:20% 3:30% ... 20:200%

Spin 0:no rotation 1:90° rotation 2:180° rotation 3:270° rotation

Category of API (1/2)

Following APIs are provided.

Application	API	Description
Setting Callback function	NSetCallback	Setting Callback function.
Obtaining printer information	NEnumPrinters	Obtaining the table of printer name.
Obtaining printer name	NGetPrinterFromID	Obtaining the printer name from various IDs.
Deleting printer name	NDeletePrinter	Deleting the printer name that is managed by this SDK.
Changing printer name	NRenamePrinter	Changing the printer name that is managed by this SDK.
Obtaining Printer Information.	NGetPrinterInf	Obtaining the printer information from the printer name.
Automatic printer setting	NAutoOpen	(This function has not been supported since ver. 3.0.0.0.)
Open printer	NOpenPrinter	Designating the printer name and opening it.
Close printer	NClosePrinter	Closing the printer which has been opened.
Close all printers	NClosePrinters	Closing all printers which have been opened.
Sending command and data	NPrint	Converting the specified hexadecimal character string data and sending it to the printer.
Sending command and data	NDPrint	Sending the specified data to the printer.
Sending image	NImagePrint	Sending the specified device context as the raster image to the printer.
Sending image	NImagePrintF	Sending the specified file (bmp/jpg/pang) as the raster image to the printer.
Obtaining status	NGetStatus	Returning status obtained from the specified printer.
Obtaining extended information	NGetInformation	Obtaining information saved in the target type ID.
Resetting printer	NResetPrinter	Resetting the printer (USB, TCP/UDP).
Document control	NStartDoc	Starting document.
Document control	NEndDoc	Ending document.
Document control	NCancelDoc	Canceling document.
Document control	NEnumDoc	Obtaining the transmission waiting documents list.
Document control	NDeleteDoc	Deleting the transmission waiting documents.

Category of API (2/2)

Following APIs are provided.

Application	API	Description
Setting barcode	NBarcodeSettings	Setting the barcode image to be generated.
Generating barcode	NBarcode	Generating barcode image.
Generating barcode	NBarcode2	Generating barcode image.
F/W upgrade	NFirmwareDL	Updating firmware by the target firmware file.
Operating UDP thread	NInitializeNetwork	Starting / stopping UDP receiving thread.
Listing TCP/UDP	NScanPrinters	Detecting TCP/UDP printers.
Locking TCP communication	NTCPPortLock	Locking / unlocking other printers' TCP communication.
Clearing TCP buffer	NBufferClear	Clearing the printing buffer of TCP/UDP printers.
TCP batch transmission	NBlockSendSetting	Starting / stopping TCP/UDP printers' batch storing transmission.

Win32 API Reference

Interface name	NCallback																																																									
Argument name	IN/OUT	Type	Description																																																							
i_prt	I	PCHAR	Printer name																																																							
i_type	I	INT	Callback type																																																							
i_value1	I	INT	Execution result 1																																																							
i_value2	I	INT	Execution result 2																																																							
Return value	void																																																									
Nothing																																																										
Processing description																																																										
<p>- This is the functional form of the callback notification.</p> <p>Declaring the function with the same return values and arguments as in this declaration and specifying the function pointer as an argument to the NSetCallback function. It can also be used as the third argument to the NOpenPrinter function to specify a callback at the opening.</p> <p>When using an asynchronous function, the result is obtained in the callback. Be sure to execute the NSetCallback function first.</p> <p>The conditions for callback notifications are as follows:</p> <table><tr><th>Callback conditions</th><th>i_prt</th><th>I-type</th><th>i_value 1</th><th>I-value 2</th></tr><tr><td>When the status value is changed. (including the first receiving.)</td><td>Printer name</td><td>1</td><td>Old status value</td><td>New status value</td></tr><tr><td>When receiving the extended information.</td><td>Printer name</td><td>2</td><td>Extended information ID</td><td>0x00</td></tr><tr><td>When the transmitting data cue becomes empty.</td><td>Printer name</td><td>3</td><td>0x00</td><td>0x00</td></tr><tr><td>NScanPrinters result notification</td><td>NULL</td><td>4</td><td>Execution result</td><td>Numbers of detection</td></tr><tr><td>NOpenPrinter result notification</td><td>Printer name</td><td>5</td><td>Execution result</td><td>0x00</td></tr><tr><td>NResetPrinter result notification</td><td>Printer name</td><td>6</td><td>Execution result</td><td>0x00</td></tr><tr><td>NSFirmwareDL result notification</td><td>Printer name</td><td>7</td><td>Execution result</td><td>0x00</td></tr><tr><td>NTCPPortLock result notification</td><td>Printer name</td><td>8</td><td>Execution result</td><td>0x00</td></tr><tr><td>NBufferClear result notification</td><td>Printer name</td><td>9</td><td>Execution result</td><td>0x00</td></tr><tr><td>NBlockSend Setting result notification</td><td>Printer name</td><td>10</td><td>Execution result</td><td>0x00</td></tr></table>				Callback conditions	i_prt	I-type	i_value 1	I-value 2	When the status value is changed. (including the first receiving.)	Printer name	1	Old status value	New status value	When receiving the extended information.	Printer name	2	Extended information ID	0x00	When the transmitting data cue becomes empty.	Printer name	3	0x00	0x00	NScanPrinters result notification	NULL	4	Execution result	Numbers of detection	NOpenPrinter result notification	Printer name	5	Execution result	0x00	NResetPrinter result notification	Printer name	6	Execution result	0x00	NSFirmwareDL result notification	Printer name	7	Execution result	0x00	NTCPPortLock result notification	Printer name	8	Execution result	0x00	NBufferClear result notification	Printer name	9	Execution result	0x00	NBlockSend Setting result notification	Printer name	10	Execution result	0x00
Callback conditions	i_prt	I-type	i_value 1	I-value 2																																																						
When the status value is changed. (including the first receiving.)	Printer name	1	Old status value	New status value																																																						
When receiving the extended information.	Printer name	2	Extended information ID	0x00																																																						
When the transmitting data cue becomes empty.	Printer name	3	0x00	0x00																																																						
NScanPrinters result notification	NULL	4	Execution result	Numbers of detection																																																						
NOpenPrinter result notification	Printer name	5	Execution result	0x00																																																						
NResetPrinter result notification	Printer name	6	Execution result	0x00																																																						
NSFirmwareDL result notification	Printer name	7	Execution result	0x00																																																						
NTCPPortLock result notification	Printer name	8	Execution result	0x00																																																						
NBufferClear result notification	Printer name	9	Execution result	0x00																																																						
NBlockSend Setting result notification	Printer name	10	Execution result	0x00																																																						

Interface name	NCallback
Processing description	<div>Caution</div> <p>Please do not execute NOpenPrinter, NSetCallback or the processes which require the synchronous control with other processes such as the screen control.</p>

Function name	NSetCallback		
Argument name	IN/OUT	Type	Description
i_callback	I	NCallback	The callback interface implemented class
Return value	void		
Nothing			
Processing description			
<div><div>- Setting the NCallback interface implementing class. Please execute this function before doing callback notification. The interface settings can also be made by using the “NOpenPrinter” function.</div><div><div>Caution</div><div>Unless null is specified in this function, the callback notification continues to the class once it is set. Please note that if the class instance you set has already been deleted, an exception will be raised from the SDK.</div><div>Please implement the callback function side of the application using the SDK.</div><div>Please do not execute NOpenPrinter, NSetCallback or the processes which require the synchronous control with other processes such as the screen control.</div></div></div>			

Win32 API Reference

Function name	NEnumPrinters / NEnumPrintersA		
Argument name	IN/OUT	Type	Description
o_printers	O	PWCHAR/PCHAR	Printer name (csv : To enumerate them in a Comma Separated Value form) (Null can be specified.)
o_size	O	PINT	Byte size of o_printers
Return value	INT		
·Error (negative value), Normal end (0) * Please refer to the error code table.			
Processing description			
<p>- It creates a printer information management file (NPrinterInf.inf) under the NPI folder in the system drive and stores a list of connection available printer names in the o_printers argument. e.g.) PRT001, PRT002, AAA</p> <p>It detects available ports, assigns a name of "PRTxxx" (xxx is a number from 001 to 999), enumerates and returns them in csv (comma separated). * Only active printers are enumerated.</p> <p>Since the printer cannot be opened (NOpenPrinter) before the printer information management file has been created, be sure to call this function at the first use. It must be called when a printer is added. When using the printer name created once, there is no need to call this function again.</p> <p>The list of printer names obtained from this function can also be obtained by directly referring to the printer information management file. However, the driver name is not recorded in the file.</p> <p>* When using with Bluetooth, it is necessary to complete pairing with the printer before calling this function.</p> <p>* Once a printer name is generated, it will not be deleted even if Bluetooth pairing is canceled. If you want to delete it , please use NDeletePrinter function.</p> <p>* Assignment of maximum 999 printer names can be generated, and a generation error will occur if it is exceeded. It is necessary to delete unnecessary printer names with NDeletePrinter or rewrite the printer name with NRenamePrinter.</p>			

Processing description

Caution

- When the driver has been set up, the printer name of the driver and the printer name with the direct communication will be stored in o_printers. (One printer is stored by both the printer name of the driver and printer name with the direct communication.)

Example) **NPI Integration Driver**, PRT001, **PRT002**, PRT005

If there is a driver (USB) printer A, both names described by the bold characters will be printer A.

*** For the direct communication, please DO NOT use the port that has already been set up with the driver.**

Reference

- * This function does not have to be called when using the driver port.

Function name	NGetPrinterFromID / NGetPrinterFromIDA		
Argument name	IN/OUT	Type	Description
i_ID	I	PWCHAR/PCHAR	ID
o_printer	O	PWCHAR/PCHAR	Printer name (NULL can be set.)
Return value	INT		
·Error (0), Normal end (The size of printer names storing area)			
Processing description	<div><div><div></div></div><div><p>- Storing the printer name in the o _ printer argument by specifying the following IDs in the i_ID argument:.</p><p>Each ID can be checked by self-diagnostic printing by the printer.</p><p><Each ID></p><p>WLAN : MAC address</p><p>Bluetooth : serial ID</p><p>USB : vendor ID 4 digits (1051) + product ID 4 digits (1000/1001/1002//1003)</p></div></div> <div><div>Caution</div><p>The printer name must have been created in the printer information management file in advance using the NEnumPrinters function.</p></div>		

Function name	NDeletePrinter / NDeletePrinterA		
Argument name	IN/OUT	Type	Description
i_prt	I	PWCHAR/PCHAR	Printer name to be deleted.
Return value	INT		
·Error (negative value), Normal end (0) * Please refer to the error code table.			
Processing description	<div><div>- This is used to delete the printer name in the printer information management file. Specifying the printer name to be deleted in the argument i_prt. By specifying a blank character, the printer information management file itself is deleted (deleting all printer names).</div><div>If the printer to be deleted is open, an error is returned. Please use this function after finishing with the connection to the target printer.</div></div>		

Caution

The driver name cannot be deleted because it is not managed by this SDK.

Function name	NRenamePrinter / NRenamePrinterA		
Argument name	IN/OUT	Type	Description
i_beforeprt	I	PWCHAR / PCHAR	Printer name before the change
i_afterprt	I	PWCHAR / PCHAR	Printer name after the change
Return value	INT		

·Error (negative value), Normal end (0) * Please refer to the error code table.

Processing description

- This is used to rename the printer name in the Printer information management file. Although the printer name is created starting with "PRT" like "PRT001" by default, it can be changed to any printer name by using this function. The printer name is limited to 50 single-byte alphanumeric characters, and the following characters cannot be used.

Characters cannot be used : 「 」 ¥ / : ? * " < > | ' , . [] and an space

If the printer whose name is about to be changed is open, an error is returned. Please use this function after finishing with the connection to the target printer.

Caution

The driver name cannot be deleted because it is not managed by this SDK.

Win32 API Reference

Function name	NGetPrinterInf / NGetPrinterInfA		
Argument name	IN/OUT	Type	Description
i_prt	I	PWCHAR/ PCHAR	Printer name
o_ports	O	PWCHAR/PCHAR	Printer information (to be enumerated in csv (comma separated value) form.) (Null can be specified.)
o_size	O	PDWORD	Size of o_ports bytes
Return value	INT		
·Error (negative value), Normal end (0) * Please refer to the error code table.			
Processing description			
<div>- Searching the printer information management file for the printer name specified by the argument and storing the following information in o _ ports. Connection types USB : 1 Bluetooth : 2 RS232C : 3 TCP/UDP : 4 NPI Driver : DRIVER ZEBRA Driver : ZEBRADRIVER The other drivers : OTHERDRIVER Connection information USB : Device instance Bluetooth : Device instance ID RS232C : Serial port TCP/UDP : Model name, IP address, Port number and MAC address Example) 1, USB¥VID_1051&PID_1000¥7&2b99b1d4&0&2 2, COM3:BTHENUM¥{00001101-0000-1000-8000-00805F9B34FB}_VID&00020430_PID&0211¥8&2D7F776&0&000B5DB4BB46_C00000000 3, COM1 4, ETHE:LAN¥NEX-M330,192.168.92.42,9100,0023A7C7E9F4 The information obtained by o_ports can also be got by referring to the printer information management file.</div>			
Reference * This function does not have to be called when using the driver port.			

27

Function name	NAutoOpen		
Argument name	IN/OUT	Type	Description
i_flg	I	INT	-
Return value	BOOL		

· Always FALSE : automatic open ineffective

Processing description

- This function has already been abolished since Ver. 3.0.0.0.
Although function itself is left due to maintaining compatibility to the previous version, "FALSE (auto open ineffective)" is always returned as returned value.

Please do openings by executing NOpenPrinter.

Reference

- * This function does not have to be called when using the driver port.

Function name	NOpenPrinter / NOpenPrinterA		
Argument name	IN/OUT	Type	Description
i_prt	I	PWCHAR /PCHAR	Printer name to be opened
i_statusFlg	I	BYTE	Flag for receiving status
i_callback	I	SDK.Ncallback	Function pointer implementing callback interface (NULL can be set.)
Return value	Int		

· Error (negative value), Normal end (0) * Please refer to the error code table.
When callback is set, it returns normal end (0) at asynchronous process starting timing.

Processing description

Function available for asynchronous processing

- Executing the printer opening process.
Specifying the printer name obtained by NEnumPrinters to the argument i_prt.
- Documents waiting to be sent will be deleted.
- This function executes the openings by asynchronous processing when the function pointer is passed to i_callback.
Although an error before starting asynchronous processing is returned by the value of this function, please obtain the execution result by using the callback function when the return value is 0 (normal end).
- After opening, please confirm that the status can be obtained before sending data.
If communication is performed before status acquisition, data loss may occur.

Reference

- * This function does not have to be called when using the driver port.

Function name	NClosePrinter / NClosePrinterA		
Argument name	IN/OUT	Type	Description
i_prt	I	PWCHAR / PCHAR	Opened printer name
Return value	Int		

·Error (negative value), Normal end (0) * Please refer to the error code table.

Processing description

- This function executes the printer closing process.

Caution

- Please call this function after confirming that printing is finished.
The status of finishing print can be obtained by using the command [Print START/FINISH Setting] and monitoring the status.
For details, refer to the Product Specifications of each printer.
- Documents waiting to be sent will be deleted.

Reference

- * This function does not have to be called when using the driver port.

Function name	NClosePrinters		
Argument name	IN/OUT	Type	Description
Return value	Int		
·Normal end (0)			
Processing description			
<div>- Closing all opened printers.</div> <div><div>Caution</div><div>This function returns only success as a returned value except when there is no printer which can be closed. . Please use the function “NClosePrinter” (closing only one printer) for obtaining the error value.</div></div>			

Function name	NPrint / NPrintA		
Argument name	IN/OUT	Type	Description
i_prt	I	PWCHAR/PCHAR	Target printer name
i_dat	I	PCHAR	Transmitting data (hexadecimal character string)
i_size	I	DWORD	Number of output bytes
o_jobid	O	PINT	Print job ID (NULL can be specified.)
Return value	Int		
·Error (negative value), Normal end (0) * Please refer to the error code table.			
Processing description			
<div>- Transmitting the specified hexadecimal character string data to the printer.</div> <p>If the following three patterns are detected during data analysis, they will be processed as special data.</p> <div>1. Character strings enclosed in "" (double quotation mark). ==> Being converted as character strings.</div> <div>2. File names enclosed in <> (signs of inequality). ==> Outputting the file contents (binary data).</div> <div>3. Image file names enclosed in [] (square brackets) (including the path, bmp format) ==> Outputting the image after converting it into raster bit image command.</div> <div>4. Character string preceded by "(single quotation mark). ==> Being processed as comments (not being output).</div> <div><div>Caution</div><div>- Regarding sequential printing and batch printing. When NPrint, NImagePrint, and NImagePrintF are executed independently, data is sent to the printer each time. By using NStartDoc and NEndDoc, it is possible to send data to the printer in bulk. Please use sequential printing or batch printing according to the purpose of use.</div></div>			

Function name	NDPrint / NDPrintA		
Argument name	IN/OUT	Type	Description
i_prt	I	PWCHAR/PCHAR	Target printer name
i_dat	I	PCHAR	Transmitting data (hexadecimal)
i_size	I	DWORD	Number of output bytes
o_jobid	O	PINT	Print job ID (NULL can be specified.)
Return value	Int		

·Error (negative value), Normal end (0) * Please refer to the error code table.

Processing description

- Transmitting the specified data to the printer (without conversion).

Caution

- Regarding sequential printing and batch printing.

When NPrint, NImagePrint, and NImagePrintF are executed independently, data is sent to the printer each time.

By using NStartDoc and NEndDoc, it is possible to send data to the printer in bulk. Please use sequential printing or batch printing according to the purpose of use.

Function name	NImagePrint / NImagePrintA		
Argument name	IN/OUT	Type	Description
i_prt	I	PWCHAR / PCHAR	Target printer name
i_bmp	I	HDC	Device context handle
i_width	I	INT	Width
i_height	I	INT	Height
i_putType	I	BYTE	Output system 0x00: Raster format by line. 0x01: Raster format by block 0x02: Raster format by block with gray scale expression 0x10: Bit image format
o_jobid	O	PDWORD	Print job ID (NULL can be specified.)
Return value	Int		

·Error (negative value), Normal end (0) * Please refer to the error code table.

Processing description

- Transmitting the specified device context to the printer.

Caution

- Regarding sequential printing and batch printing.

When NPrint, NImagePrint, and NImagePrintF are executed independently, data is sent to the printer each time.

By using NStartDoc and NEndDoc, it is possible to send data to the printer in bulk. Please use sequential printing or batch printing according to the purpose of use.

Function name	NImagePrintF / NImagePrintFA		
Argument name	IN/OUT	Type	Description
i_prt i_bmp i_putType o_jobid	I I I O	PWCHAR / PCHAR PCHAR BYTE PDWORD	Target printer name BMP file name Output system 0x00: Raster format by line. 0x01: Raster format by block 0x02: Raster format by block with gray scale expression 0x10: Bit image format Print job ID (NULL can be specified.)
Return value	Int		
·Error (negative value), Normal end (0) * Please refer to the error code table.			
Processing description			
<div>- Reading the specified BMP file and transmitting the image to the printer.</div> <div><div>Caution</div><div><div>- Regarding sequential printing and batch printing. When NPrint, NImagePrint, and NImagePrintF are executed independently, data is sent to the printer each time. By using NStartDoc and NEndDoc, it is possible to send data to the printer in bulk. Please use sequential printing or batch printing according</div></div></div>			

Function name		NGetStatus / NGetStatusA		
Argument name		IN/OUT	Type	Description
i_prt		I	PWCHAR / PCHAR	Target printer name
o_status		O	LPDWORD	Status
Return value	Int			
·Error (negative value), Normal end (0) , Warning (positive value) * Please refer to the error code table.				
Processing description				
<div>- Returning the status obtained from the specified printer. * Please refer to “Processing Errors” in the target printer’s Product Specifications for values to be returned.</div> <div>- Even if the returned value is a negative value, the printer status is returned by “o_status” when it is “N_ERR_SEND_ERROR (-3) ”.</div> <div>- When it is off-line during a transmission error, the returned value will be “N_ERR_SEND_ERROR (-3)” and “o_status” will “0xFFFFFFFF” (off-line).</div>				

Function name	NGetInformation / NGetInformationA		
Argument name	IN/OUT	Type	Description
i_prt	I	PWCHAR/PCHAR	Target printer name
i_id	I	BYTE	Type ID
o_dat	O	PVOID	Extended Information storing area
o_time	O	PDWORD	Update flag (elapsed time since system startup) (NULL can be specified.)
Return value	Int		
·Error (negative value), Normal end (0) * Please refer to the error code table.			
Processing description			
<div>- Obtaining information stored in target type of ID of extended information. * It is necessary to send a request of desired information from a higher-level application to the printer in advance. (Some do not require requests for extended status, transmission completion, print completion information, etc.) * Please refer to the part of the command 《ESC v》 in the target printer's Product Specifications for values to be returned.</div>			

Extended Information (1/2)

Type 1 : 4 bytes (fixed) : update flag (4bytes)	<Extended status> 1st byte : 7~0, 2nd byte : 15~8, 3rd byte : 23~16, 4th byte : 31~24
Type 2 :32 bytes (delimiter) : update flag (4bytes)	<Model name>
Type 3 : 8 bytes (fixed) : update flag (4 bytes)	<F/W version>
Type 4 : 8 bytes (fixed) : update flag (4 bytes)	<Boot version>
Type 5 : 4 bytes (fixed) : update flag (4 bytes)	<Reserved>
Type 6 : 4 bytes (fixed) : update flag (4 bytes)	<Number of dot lines energizing head>
Type 7 : 4 bytes (fixed) : update flag (4 bytes)	<Number of fed dot lines>
Type 8 : 4 bytes (fixed) : update flag (4 bytes)	<Number of cuts>
Type 9 :16 bytes (fixed) : update flag (4 bytes)	<User maintenance counter: Number of dot lines energizing head, Number of fed dot lines, Number of cuts, Reserved>
Type10 :16 bytes (fixed) : update flag (4 bytes)	<Reserved>
Type11 :64 bytes (delimiter) : update flag (4 bytes)	
Type12 :32 bytes (delimiter) : update flag (4 bytes)	
Type13 :32 bytes (fixed) : update flag (4 bytes))	<NV registration status>
Type14 :32 bytes (fixed) : update flag (4 bytes)	<Reserved>
Type15 :16 bytes (fixed) : update flag (4 bytes)	
Type16 :16 bytes (fixed) : update flag (4 bytes)	
Type17 :16 bytes (fixed) : update flag (4 bytes)	
Type18 :16 bytes (fixed) : update flag (4 bytes)	
Type19 : 8 bytes (fixed) : update flag (4 bytes)	<End of print notification: Arbitrary ID and end status are described at the time of end command processing by specifying the print start/end command.>
	<Reserved>
Type20 : 8 bytes (fixed) : update flag (4 bytes)	
Type21 : 8 bytes (fixed) : update flag (4 bytes)	
Type22 : 8 bytes (fixed) : update flag (4 bytes)	
Type23 : 8 bytes (fixed) : update flag (4 bytes)	
Type24 : 4 bytes (fixed) : update flag (4 bytes)	<Reserved>
Type25 : 4 bytes (fixed) : update flag (4 bytes)	<Transfer completion notification: transferred job ID will be described>
Type26 : 4 bytes (fixed) : update flag (4 bytes)	<Reserved>
Type27 : 4 bytes (fixed) : update flag (4 bytes)	<Reserved>
Type28 : 2 bytes (fixed) : update flag (4 bytes)	<F/W checksum>
Type29 : 2 bytes (fixed) : update flag (4 bytes)	
Type30 : 2 bytes (fixed) : update flag (4 bytes)	
Type31 : 2 bytes (fixed) : update flag (4 bytes)	<Communication status information: USB: 0x0000 fixed COM: 1 st Byte, CTS 2 nd byte, DSR * Final signal status obtaining time stamp is set to update flag.>

* Except for types 25 and 31, the obtained information is not valid for information that is not implemented by the printer.

* The contents described here may not be available for all printers.

Extended Information (2/2)

Type 32 : 8 bytes hexadecimal character string / character string

Type 33 : 4 bytes hexadecimal character string / character string

Type 34 : 2 bytes hexadecimal character string / character string

Type 35 : 8 bytes hexadecimal character string

Type 36 : 8 bytes hexadecimal character string

Type 37 : 4 bytes hexadecimal character string

Type 38 : 4 bytes hexadecimal character string

Type 39 : 2 bytes hexadecimal character string

Type 40 : 2 bytes hexadecimal character string

* The contents described here may not be available for all printers. Information returned by sub ID for type 11, 32, 33 and 34 will be changed.

The list of extended information types related to network

MAC address (6 bytes)

IP address (4 bytes)

Sub net address (4 bytes)

Default gateway (4 bytes)

DNS server address (4 bytes * 5)

Printing time-out AC (milliseconds unit : 4 bytes)

Communication mode (infrastructure, ad hock)

Band (2,4GHz / 5GHz)

Transmission power level (Low / Medium / High)

Channel

Security type address (Open / WPA / WPA2 / WEP)

Encryption type (6 bytes)

IP setting system (Automatic (DHCP / APIPA) / Manual)

Type32-0 response : 16 hexadecimal character string

Type33-0 response : 16 hexadecimal character string

Type33-1 response : 16 hexadecimal character string

Type33-2 response : 16 hexadecimal character string

Type33-3 response : 16 hexadecimal character string

Type33-14 response : 16 hexadecimal character string

Type34-4 response : 16 hexadecimal character string

Type34-5 response : 16 hexadecimal character string

Type34-6 response : 16 hexadecimal character string

Type34-7 response : 16 hexadecimal character string

Type34-8 response : 16 hexadecimal character string

Type34-9 response : 16 hexadecimal character string

Type34-10 response : 16 hexadecimal character string

Function name	NResetPrinter / NResetPrinterA		
Argument name	IN/OUT	Type	Description
i_prt	I	PWCHAR / PCHAR	Target printer name
Return value	Int		

·Error (negative value), Normal end (0) * Please refer to the error code table.
When callback is set, it returns normal end (0) at asynchronous process starting timing.

Processing description

Function available for asynchronous processing

- Resetting the printer. The print job being printer is also canceled.
- Please check the return value of this API and that the printer is reset and online (check by NGetStatus) in order to confirm that this API has been executed normally.
- This function will be executed asynchronously when a callback function is set. Although an error before starting asynchronous processing is returned by the value of this function, please obtain the execution result by using the callback function when the return value is 0 (normal end).

Caution

Resetting can be executed only under USB or TCP/UDP connection.

Function name	NStartDoc / NStartDocA		
Argument name	IN/OUT	Type	Description
i_prt o_jobid	I O	PWCHAR / PCHAR PINT	Target printer name Print job ID (NULL can be specified.)
Return value	Int		
·Error (negative value), Normal end (0) * Please refer to the error code table.			
Processing description	<div>- Starting the document. After NStartDoc is published, the data is stored in a temporary buffer for NPrint, NDPrint, NImagePrint, and NImagePrintF. The saved data can be output to the printer by calling NEndDoc. The accumulated data will be cleared by calling NCancelDoc. One document can be handled per printer.</div>		

Function name	NEndDoc / NEndDocA		
Argument name	IN/OUT	Type	Description
i_prt	I	PWCHAR / PCHAR	Target printer name
Return value	Int		
·Error (negative value), Normal end (0) * Please refer to the error code table.			
Processing description			

- Ending the document.
The data stored after calling NStartDoc is output to the printer.
If the stored data does not exist, processing is not performed.

Function name	NCancelDoc / NCancelDocA		
Argument name	IN/OUT	Type	Description
i_prt	I	PWCHAR / PCHAR	Target printer name
Return value	Int		
·Error (negative value), Normal end (0) * Please refer to the error code table.			
Processing description			

- Canceling the document.
If the stored data does not exist, processing is not performed.

Function name	NEnumDoc / NEnumDocA		
Argument name	IN/OUT	Type	Description
i_prt	I	PWCHAR / PCHAR	Target printer name
o_list	O	PWCHAR / PCHAR	Job ID list (CSV) (NULL can be specified.)
Return value	Int		
·Error / no transmission waiting (0), Normal end (list saving size)			
Processing description		<div><ul style="list-style-type: none">- Returning the waiting job ID list (CSV).* One document information is composed of document stats, document number (18 digits) and line feeding.* Document status 0 : transmission waiting, 1 : under being transmitted 2 : transmission error* Only the size required for the list saving is obtained by specifying NULL for o_list.<div><div></div><div>This function does not have to be called when using the driver port.</div></div></div>	

Function name	NDeleteDoc / NDeleteDocA		
Argument name	IN/OUT	Type	Description
i_prt	I	PWCHAR / PCHAR	Target printer name
i_jobID	I	INT	Print job ID
Return value	Int		
·Error (negative value), Normal end (0) * Please refer to the error code table.			
Processing description		<div>- Canceling the document. Deleting transmission waiting document. If the transmission waiting document does not exist, processing is not performed.</div> <div><div>Caution</div><div>* This function is not effective when using the driver port.</div></div>	

Win32 API Reference

Function name	NBarcodeSettings		
Argument name	IN/OUT	Type	Description
i_id	I	UINT	Barcode type
i_width	I	UINT	Magnification of data width
i_height	I	UINT	Height of one module
i_option1	I	CHAR	Option 1
i_option2	I	CHAR	Option 2
i_option3	I	CHAR	Option 3
i_option4	I	CHAR	Option 4
i_hri	I	UINT	HRI characters position
i_spin	I	UINT	Rotation
i_code	I	UINT	Character code type
Return value	Int		
·Error (negative value), Normal end (0) * Please refer to the error code table.			
Processing description			
<div><div>- Before calling the NBarcode2 function, this function sets the output settings.</div><div>- NULL can be specified for the items not to be used. In this case, the default values will be applied.</div><div>- There are 2 types in i_id of CODE39, CPDE93 and CODE128. One is i_id whose options are fixed, and the other will be generated by the specified options. When options are fixed, automatic check digit and start code assign is ineffective for CODE39, □ assign to before and after HRI characters is effective for CODE93 and manual input of start code or SHIFT is effective for CODE128.</div><div>- Please refer to the “Barcode setting items list” on the following page for the setting values to each item.</div></div>			

Barcode setting items list

i_id (barcode type) (default 0)	1D barcodes 0 UPC-A 1 UPC-E 2 JAN-13 (EAN-13) 3 JAN-8 (EAN-8) 4 CODE39 (option fixed) 5 ITF 6 CODABAR (NW-7) 7 CODE128 (option fixed) 8 CODE93 (option fixed) 9 ISBN 10 Industrial 2 of 5 11 Matrix 2 of 5 12 NEC Matrix 13 Customer barcode 14 MSI Code 15 EAN-128 (GS1-128) 16 GS1 DataBar Omnidirectional 17 GS1 DataBar Truncated 18 GS1 DataBar Limited 19 GS1 DataBar Expanded 20 CODE39 (with option setting) 21 CODE128 (with option setting) 22 CODE93 (with option setting)	2D barcodes 30 QR Code 31 Micro QR code 32 Aztec Code 33 PDF417 34 Micro PDF417 35 MaxiCode 36 DataMatrix 37 GS1 DataBar Stacked 38 GS1 DataBar Stacked Omnidirectional 39 GS1 DataBar Expanded Stacked 40 EAN-8 Composite 41 EAN13 Composite 42 UPC-A Composite 43 UPC-E Composite 44 GS1-128 Composite 45 GS1 DataBar Omnidirectional Composite 46 GS1 DataBar Truncated Composite 47 GS1 DataBar Limited Composite 48 GS1 DataBar Expanded Composite 49 GS1 DataBar Stacked Composite 50 GS1 DataBar Stacked Omnidirectional Composite 51 GS1 DataBar Expanded Stacked Composite 52 GS1 QR Code 53 GS1 Aztec Code 54 GS1 DataMatrix
i_width (default 3)	Width for one module : 1~20 dots	
i_height (default 162)	Height of the bar (dots) : max. 3776 dots * Specifiable only for 1-d barcode.	
i_option1~4	1 to 6 can be set to option 4. (option1 default 0, option2 default 0, option3 default 0, option4 default 3) Please refer to the next page for details.	
i_hri (default 0)	0: None 1: Top (Font A) 2: Bottom (Font A) 3: Top and Bottom (Font A) 4: Top (Font B) 5: Bottom (Font B) 6: Top and Bottom (Font B) 7: Top (Font D) 8: Bottom (Font D) 9: Top and Bottom (Font D) *Specifiable for 1D barcodes and some 2D barcodes described below. 2D barcodes to which this can be specified (UPC-A Composite / UPC-E Composite / JAN-13 Composite / JAN-8 Composite / GS1 128 Composite / GS1 DATABAR Omnidirectional Composite / GS1 DATABAR Truncated Composite / GS1 Limited Composite / GS1 Expanded Composite)	
i_spin (default 0)	0: no rotation 1: 90° rotation 2: 180° rotation 3: 270° rotation	
i_code (default 0)	0: ISO-8859-1 1: Shift-JIS 2: UTF-8 * Shift-JIS and UTF-8 can be specified with only QR Code, PDF417, MicroPDF417 and Maxicode.	

Win32 API Reference

Options list

Options will be disregarded when values not described are specified.

1-D barcodes options

Barcode system	Options	
CODE39	Option1	Automatic check digit assign = effective(1) Automatic start code assign = effective(2) Both above effective(3)
CODE93	Option1	Adding □ before and after HRI characters = effective(2)
CODE128	Option1	Manual input of start code or SHIFT = effective(2)
MSI Code	Option1	Check digit None(0), Mod-10(1), Mod-1010(2), Mod-11(3), Mod-1110(4)

2-D barcodes options

Barcode system	Options	
QR Code	Option1	Error correction level (-1,0 = automatic, 1=L, 2=M, 3=Q, 4=H)
	Option2	Version specifying (0 = automatic, 1~40 = specifying) "1~40" specifies "21X21" ~ "177X177" modules. (Value of 4 in "21" to "177" part will be increased by one value increase in "1" to "40" part.)
	Option3	Mask pattern level (0 = automatic, 1~8 = specifying)
Micro QR Code	Option1	Error correction level (-1,0 = automatic, 1=L, 2=M, 3=Q)
	Option2	Version specifying (0 = automatic, 1~4 = specifying) "1~4" specifies "11X11" ~ "17X17" modules. (Value of 2 in "11" to "17" part will be increased by one value increase in "1" to "4" part.)
Aztec Code	Option1	Error correction level (-1,0 = automatic, 1~4 = specifying)
	Option2	Size specifying (0 = automatic, 1~36 = specifying) "1~4" specifies "15X15" ~ "27X27" modules Aztec Code. "5~36" specifies "19X19" ~ "151X151" modules Aztec Code. (Value of 4 in "15" to "27" and "19" to "151" part will be increased by one value increase in "1" to "36" part.)
	Option3	Reader initialization data assign (1 = effective) Not generating Compact Aztec at automatic sizing (2 = effective) Both above effective = 3

Win32 API Reference

Barcode system	Options	
PDF417	Option1	Check digit number specifying (-1 = automatic, 1~8 = specifying) (-1 = automatic, 0 = 2 characters, 1 = 4 characters, 2 = 8 characters, 3 = 16 characters, 4 = 32 characters, 5 = 64 characters, 6 = 128 characters, 7 = 256 characters, 8 = 512 characters)
	Option2	Data part rows number specifying (0 = automatic, 1~30 = rows number specifying)
	Option3	Reader initialization data assign (0 = ineffective, 1 = effective)
	Option4	Height ratio specifying (1~6 = specifying, not specified : 3)
Micro PDF417	Option2	Data part rows number specifying (0 = automatic, 1~4 = rows number specifying) When it is beyond the specified number, it will be automatic.
	Option3	Reader initialization data assign (0 = ineffective, 1 = effective)
MaxiCode	Option1	Mode selecting (-1,0 = automatic, 2~6 specifying) 2 = handling the primary message as 9 digits zip code 3 = handling the primary message as 6 digits zip code 4 = error correction level (primary message EEC, secondary message SEC) 5 = error correction level (all messages EEC) 6 = error correction level (all messages SEC)
	Option2	Primary message (0 = ineffective, 1 = effective) Since the first 15 digits of i_dat will be the primary message, more than 16 digits input is required.

Win32 API Reference

Barcode system	Options				
DataMatrix	Option1	Reader initialization data assign (-1 = ineffective, 1 = effective)			
	Option2	Size specifying (0 = automatic, 1~30 = specifying)			
		1	10 x 10	16	64 x 64
		2	12 x 12	17	72 x 72
		3	14 x 14	18	80 x 80
		4	16 x 16	19	88 x 88
		5	18 x 18	20	96 x 96
		6	20 x 20	21	104 x 104
		7	22 x 22	22	120 x 120
		8	24 x 24	23	132 x 132
		9	26 x 26	24	144 x 144
		10	32 x 32	25	8 x 18
		11	36 x 36	26	8 x 32
		12	40 x 40	27	12 x 26
		13	44 x 44	28	12 x 36
		14	48 x 48	29	16 x 36
	15	52 x 52	30	16 x 48	
	Option3	Whether it generates the rectangle layout size at automatic sizing. 1 = Always generating the square layout size. 2 = Generating the rectangle layout size if it is smaller than the square layout size.			

Barcode system	Options	
GS1 DataBar Stacked	Without options	
GS1 DataBar Stacked Omnidirectional	Without options	
GS1 DataBar Expanded Stacked	Option2	Data part rows number specifying (1~10 = rows number specifying, it will be 2 rows when it is not specified.)
EAN-8 Composite	Option1	Type of composite component (-1 = automatic, 1 =CC-A, 2 = CC-B)
EAN13 Composite	Option1	Type of composite component (-1 = automatic, 1 =CC-A, 2 = CC-B)
UPC-A Composite	Option1	Type of composite component (-1 = automatic, 1 =CC-A, 2 = CC-B)
UPC-E Composite	Option1	Type of composite component (-1 = automatic, 1 =CC-A, 2 = CC-B)
GS1-128 Composite	Option1	Type of composite component (-1 = automatic, 1 =CC-A, 2 = CC-B, 3 = CC-C)
GS1 DataBar Omnidirectional Composite	Option1	Type of composite component (-1 = automatic, 1 =CC-A, 2 = CC-B)
GS1 DataBar Truncated Composite	Option1	Type of composite component (-1 = automatic, 1 =CC-A, 2 = CC-B)
GS1 DataBar Limited Composite	Option1	Type of composite component (-1 = automatic, 1 =CC-A, 2 = CC-B)
GS1 DataBar Expanded Composite	Option1	Type of composite component (-1 = automatic, 1 =CC-A, 2 = CC-B)
GS1 DataBar Stacked Composite	Option1	Type of composite component (-1 = automatic, 1 =CC-A, 2 = CC-B)
GS1 DataBar Stacked Omnidirectional Composite	Option1	Type of composite component (-1 = automatic, 1 =CC-A, 2 = CC-B)
GS1 DataBar Expanded Stacked Composite	Option1	Type of composite component (-1 = automatic, 1 =CC-A, 2 = CC-B)

Barcode system	Options	
GS1 QR CODE	Option1	Error correction level (-1 = automatic, 1 = L, 2=M, 3=Q, 4=H)
	Option2	Version specifying (0 = automatic, 1~40 = specifying)
	Option3	Mask pattern level (0 = automatic, 1~8 = specifying)
GS1 Aztec CODE	Option1	Error correction level (-1 = automatic, 1~4 = specifying)
	Option2	Size specifying (0 = automatic, 1~36 = specifying)
	Option3	Excluding Compact Aztec from the alternatives at automatic sizing = 2
GS1 DataMatrix	Option2	Size specifying (0 = automatic, 1~30 = specifying)
	Option3	Whether it generates the rectangle layout size at automatic sizing. 1 = Always generating the square layout size. 2 = Generating the rectangle layout size if it is smaller than the square layout size.

Function name	NBarcode / NBarcodeA		
Argument name	IN/OUT	Type	Description
i_prt	I	PWCHAR / PCHAR	Target printer name
i_fontName	I	PWCHAR / PCHAR	Font name
o_bmp	O	HDC	Device context handle
i_x	I	DWORD	Left
i_y	I	DWORD	Top
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			
<p>- Drawing the barcode on o_bmp with the settings specified in the printer's barcode font. Please note that 2D barcodes are not supported under iOS and SDK does not support 2D barcodes. Only 1D barcodes can be used..</p> <p>Please specify the font name used in the barcode configuration file (NBarcodeInf) for the font name in the 2nd argument. (Barcode 1 through Barcode 10)</p> <p>Please refer to “Regarding barcode setting file” in this document for NBarcodeInf.inf.</p> <div><div>Caution</div><p>- Drawing the barcode by using the NBarcode2 function is recommended from Ver. 3.2.2.0. The support of this function may be stopped in the future.</p><p>- If the barcode data is large and the created bar code exceeds the data size, no error occurs and the barcode data deleting the excess part is created. In this case, the created barcode cannot be read. Please adjust the data size so that all the data can be included.</p></div>			

Function name	NBarcode / NBarcodeA
Processing description	


Image of setting arguments

UIImage class area (third argument – gray color frame part)



- * Setting the width, height, presence or absence of HRI characters, etc. of the barcode part **enclosed in red square** in the barcode are required to be set by the barcode setting file (NBarcodeInf).
- * The barcode rotation setting rotates the part **enclosed in red square**. Since the white part in the above figure does not rotate, please set the area with the width and height after rotation.

Function name	NBarcode2		
Argument name	IN/OUT	Type	Description
o_bmp	O	HDC	Device context handle
i_x	I	DWORD	Left
i_y	I	DWORD	Top
io_width	IO	PDWORD	Width
io_height	IO	PDWORD	Height
i_dat	I	PBYTE	Barcode data
i_size	I	INT	Data size
Return value	Int		
·Error (negative value), Normal end (0) * Please refer to the error code table.			
Processing description	<ul style="list-style-type: none">- Drawing the barcode on “o_bmp” in accordance with the setting specified by using the function “NBarcodeSettings”.- The barcode to be generated is required to be set before executing this function.- By passing “NULL” for “o_bmp”, required width and height for generating “io_width” and “io_height” will be returned.- When passing the device context handle to “o_bmp”, the width and the height of the device context handle will be passed.		
<div>Caution</div> <ul style="list-style-type: none">- If the barcode data is large and the created bar code exceeds the data size, no error occurs and the barcode data deleting the excess part is created. In this case, the created barcode cannot be read. Please adjust the data size so that all the data can be included.			

Function name	NBarcode2
Processing description	
Image of setting arguments	<p>HDC area (first argument – gray color frame part)</p>  <p>The diagram shows a gray rectangular area representing the HDC. Inside it is a white rectangular area representing the barcode. A red square is drawn over the barcode, indicating the area that can be rotated. Dimensions are labeled: i_x and i_y for the offset of the barcode from the top-left of the gray area; i_width and i_height for the dimensions of the white barcode area.</p> <ul style="list-style-type: none">* Setting the width, height, presence or absence of HRI characters, etc. of the barcode part enclosed in the red square in the barcode are required to be set by using the NBarcodeSettings function.* The barcode rotation setting rotates the part enclosed in red square. Since the white part in the above figure does not rotate, please set the area with the width and height after rotation.

Function name	NFirmwareDL / NFirmwareDLA		
Argument name	IN/OUT	Type	Description
i_prt	I	PWCHAR/ PCHAR	Target printer name
i_file	I	PWCHAR/ PCHAR	Fwf file name (NULL can be specified.)
i_errflg	I	BYTE	Error check 0x00: invalid (forced transmission) 0x01: valid
io_chksum	IO	PWORD	Fwf file checksum
o_jobid	O	PINT	Print job ID (NULL can be specified.)
Return value	Int		
<p>·Error (negative value), Normal end or checksum match (0)</p> <p>* Please refer to the error code table</p> <p>It will return the normal end (0) at the asynchronous processing start timing when the callback is set.</p>			
Processing description		Function available for asynchronous processing	
<p>- Transmitting the specified FWF file to the printer.</p> <p>If the FWF file is specified as NULL, the checksum is obtained from the printer and compared with the specified checksum.</p> <p>- When the callback function is set, this function performs firmware rewriting process asynchronously. Although an error before starting asynchronous process is returned by the return value of this function, please obtain the execution result by the callback function when the return value 0 (normal end) is returned.</p> <p>However, when the checksum is checked by setting NULL for FWF file, the return value of this function will be the result of the checksum check because it is processed synchronously.</p> <p>* Before using this function, be sure to perform NGetStatus in advance in order to ensure that the printer is working properly, then please execute this function.</p>			

Function name	NInitializeNetwork		
Argument name	IN/OUT	Type	Description
i_flg	I	INT	The flag to activate / stop UDP receiving thread 0 : stop 1 : activate
Return value	Int		
·Error (negative value), Normal end (0) * Please refer to the error code table			
Processing description			
<p>- Activating / stopping UDP receiving thread. It is stopped at the initial state.</p> <p>During TCP / UDP communication, please activate the UDP receiving thread by this function before executing open processing (NOpenPrinter).</p> <p>Please execute this function at the first timing of the service start when using TCP/UDP communication. When TCP/UDP communication is not used, please DO NOT execute this function.</p>			

Caution

- When multiple applications use UDP receiving thread, UDP receiving thread activating and stopping will be applied to all applications.

- This process will result in time-out if UDP receiving thread activation / stop cannot be detected after 5 seconds. In this case, there is a risk that NServiceDrv is not working properly. Please check whether NServiceDrv is executed by service management tool.

- When you open the printer setup screen with LAN-connected driver, NInitializeNetwork will be performed and UDP receiving service will be started automatically.

58

Function name	NScanPrinters		
Argument name	IN/OUT	Type	Description
i_waitmsec	l	long	Number of milliseconds to wait for enumeration response / data creation
Return value	Int		
<div>·Error (negative value), Normal end (number of detected printers)</div> <div>* Please refer to the error code table</div> <div>It will return the normal end (0) at the asynchronous processing start timing when the callback is set.</div>			
Processing description		Function available for asynchronous processing * This function is available only for TCP/UDP printers.	
<div><div>- The is a function for detecting TCP/UDP printers. It does not have to be executed when TCP/UDP printers are not used.</div><div>Request for enumerating printers is transmitted broadcast by executing this function.</div><div>Receiving a printer enumeration response for milliseconds specified by the argument i_waitmsec and creating printer information from the response data.</div><div>Please specify 3000 (3 seconds) to i_waitmsec in normal cases.</div><div>(NEnumPrinters function is used for obtaining printer information separately.)</div><div>- When the callback function is set, this function performs firmware rewriting process asynchronously. Although an error before starting asynchronous process is returned by the return value of this function, please obtain the execution result by the callback function when the return value 0 (normal end) is returned.</div></div>			

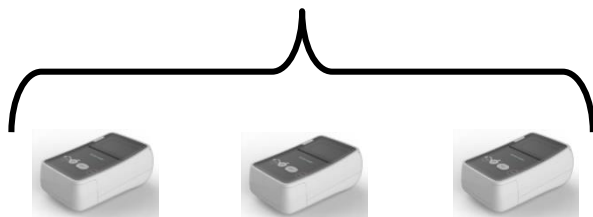
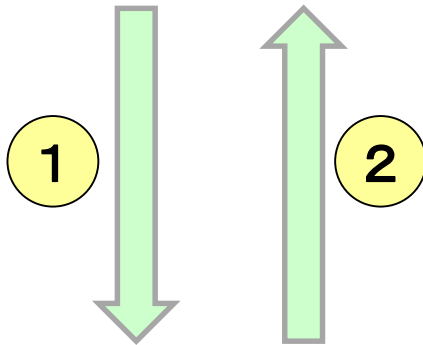
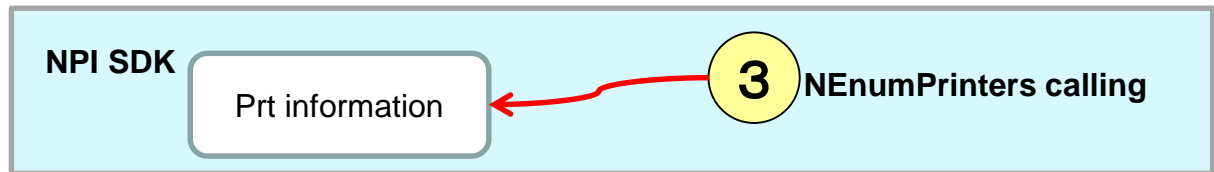
Caution

If 0 is specified for i_waitmsec, no printer enumeration request is made and the number of printers already created is returned as the return value.

Printer name obtaining when using TCP/UDP(WLAN)



PC / tablet, etc.



192.168.0.1

192.168.0.2

192.168.0.3

* NScanPrinters function is called to create printer information.

Executing this function is needed when

- TCP/UDP printer is newly added.
- IP address of the printer is changed.

This function does not have to be executed every time NEnumPrinters function is used.

- 1** NScanPrinters calling
(Printer enumeration request : UDP / transmitting it widely to each printer)
- 2** Printer enumeration response from each printer with TCP/UDP connection :
receiving UDP as printer information in SDK.
- 3** NEnumPrinters calling
(Obtaining TCP/UDP printer information based on the receiving data.)

Function name	NTCPortLock / NTCPortLockA		
Argument name	IN/OUT	Type	Description
i_prt	I	PWCHAR / PACHA	Target printer name
i_type	I	BYTE	Setting type
Return value	Int		
·Error (negative value), Normal end (0) * Refer to error code table			
Processing description	Function available for asynchronous processing		
<div>- Locking / unlocking TCP communication. Only the locked users can send data via TCP communication. Other users can be connected but cannot send data. Setting type Lock : 0 x 01 Unlock : 0 x 00</div> <div>- When the callback function is set, this function performs firmware rewriting process asynchronously. Although an error before starting asynchronous process is returned by the return value of this function, please obtain the execution result by the callback function when the return value 0 (normal end) is returned.</div>			

Function name	NBufferClear / NBufferClearA		
Argument name	IN/OUT	Type	Description
i_prt	I	PWCHAR / PACHA	Target printer name
Return value	Int		
·Error (negative value), Normal end (0) * Refer to error code table			
Processing description		Function available for asynchronous processing	
<div><div>- Clearing receiving buffer (all buffers). When it is connected except for TCP/UDP connection, an error is returned.</div><div>- When the callback function is set, this function performs firmware rewriting process asynchronously. Although an error before starting asynchronous process is returned by the return value of this function, please obtain the execution result by the callback function when the return value 0 (normal end) is returned.</div></div>			

Function name	NBlockSendSetting / NBlockSendSettingA		
Argument name	IN/OUT	Type	Description
i_prt	I	PWCHAR / PACHA	Target printer name
i_type	I	BYTE	Setting type
Return value	Int		
·Error (negative value), Normal end (0) * Refer to error code table			
Processing description	Function available for asynchronous processing		
<div>- Transmitting data can be stored in the buffer of TCP/UDP module for batch transmission. This function can switch ON / OFF of the batch transmission. (When it is connected except for TCP/UDP connection, an error is returned.)</div> <div>Setting type Lock : 0 x 01 Unlock : 0 x 00</div> <div>- When the callback function is set, this function performs firmware rewriting process asynchronously. Although an error before starting asynchronous process is returned by the return value of this function, please obtain the execution result by the callback function when the return value 0 (normal end) is returned.</div>			

Error codes table (1/2)

This is the list of error codes used by this SDK. Although these are mainly used for the return values from the functions there are some error codes which are used only for SDK internal processes, and others are for maintaining compatibility to other OSes. Please refer to “List of error codes by functions” for return values from the functions.

N_SUCCESS	0	Normal end
N_ERR_HANDLE	-1	Handle error
N_ERR_PRTOPEN	-2	Printer open error
N_ERR_SEND_ERROR	-3	In a transmitting error
N_ERR_OFFLINE	-5	Offline
N_ERR_PRTCLOSE	-6	Printer close error
N_ERR_FILEOPEN	-10	File open error
N_ERR_NOT_MAPPING	-11	Extended information obtaining error (MapViewOfFile failure)
N_ERR_NOT_OPEN_MAPFILE	-12	Map file open error
N_ERR_PRTOUTPUT	-13	Printer output error
N_ERR_NONE_PRTLST	-21	Not existing available printers
N_ERR_NOHANDLE	-22	Printer has not been opened.
N_ERR_LACKRESOURCE	-31	Lack of resources
N_ERR_NOTSUPPORTED	-40	Function not being supported
N_ERR_LOADFROMFILE	-50	Failed to load image file
N_ERR_IMAGESIZE	-51	Incorrect image size
N_ERR_RESETPRINTER	-60	Resetting failure
N_ERR_ACCESSDENIED	-61	Access denied
N_ERR_STARTDOC	-70	StartDoc function error
N_ERR_DOCNOTSTARTED	-71	Not in document start state
N_ERR_ALREADYSTARTDOC	-72	Already in document start state
N_ERR_FWFFILE	-80	fwf file error
N_ERR_FWF_CHECKSUM	-81	The checksum entered in the argument does not match the checksum obtained from the printer.
N_ERR_FWDL_TIMEOUT	-82	Firmware download time-out (Time-out of checking the printing start command)
N_ERR_FWCHK_TIMEOUT	-83	Time-out of checking the firmware checksum
N_ERR_FWDL_FOUNDEROR	-84	Detecting error in checking the status during firmware download
N_ERR_ARGUMENT	-90	Argument incorrect value error
N_ERR_ARGUMENT_01	-91	1st argument error
N_ERR_ARGUMENT_02	-92	2nd argument error
N_ERR_ARGUMENT_03	-93	3rd argument error
N_ERR_ARGUMENT_04	-94	4th argument error
N_ERR_ARGUMENT_05	-95	5th argument error

Error codes table (2/2)

N_ERR_ARGUMENT_06B	-96	6th argument error
N_ERR_ARGUMENT_07	-97	7th argument error
N_ERR_ARGUMENT_08	-98	8th argument error
N_ERR_ARGUMENT_09	-99	9th argument error
N_ERR_SOCKRECV	-108	TCP/UDP receiving error
N_ERR_UDPTHREADSTARTED	-111	UDP thread has already been started.
N_ERR_UDPTHREADSTOPPED	-112	UDP thread is stopped or cannot be activated.
N_ERR_UDPTHREADSTOP	-113	UDP thread is not able to be stopped.
N_ERR_PRTINFO_READ	-131	Failed in reading printer information file
N_ERR_PRTINFO_WRITE	-132	Failed in writing printer information file
N_ERR_PRTNAME_ALLOC	-133	Filed in printer name allocation
N_ERR_PRTRENAME_BEFORE	-134	Printer name before being changed does not exist.
N_ERR_PRTRENAME_AFTER	-135	Printer name after being changed has already been used.
N_ERR_PRTINFO_ILLEGAL	-137	Incorrect printer information file
N_ERR_PRTINFO_DELETE	-138	Failed in deleting printer name
N_ERR_PRTINFO_NOTFOUND	-139	Not existing printer name
N_ERR_DEVICE_NOTSUPPORT	-150	Connection type is not supported.
N_ERR_CREATEBCDDATA	-162	Failed in generating barcode data
N_ERR_BARCODE_DATA_LONG	-163	Barcode character string is too long.
N_ERR_BARCODE_DATA_INVALID	-164	Character string not supporting barcode is included.
N_ERR_BARCODE_OPTION	-165	Incorrect combination of options
N_ERR_BARCODE_BUFFER	-166	Insufficient buffer for barcode generation
N_ERR_MNT_HEADER	-200	Incorrect maintenance response header
N_ERR_FLAGSQR	-201	Incorrect maintenance response (Flags) QR flag
N_ERR_FLAGSFORMAT	-202	Detecting format error in maintenance response (Flags)
N_ERR_FLAGSBUSY	-203	Detecting busy in maintenance response (Flags)
N_ERR_FLAGSUNDEFINED	-204	Detecting unimplemented in maintenance response (Flags)
N_ERR_FLAGSREJECT	-205	Detecting denial in maintenance response (Flags)
N_ERR_FLAGSOTHER	-206	Detecting other errors in maintenance response (Flags)
N_ERR_MNT_QID	-207	Incorrect maintenance response query ID
N_ERR_MNT_QR	-208	Incorrect maintenance response spare flag
N_ERR_MNT_QPARAM	-209	Incorrect maintenance response query parameter
N_ERR_MNT_OTHER	-210	Other errors in maintenance response
N_ERR_NO_LIBRARY	-900	DDL cannot be read.
N_ERR_OTHER	-999	Other errors
N_WRN_PRTALREADYOPEN	10	Printer has already been opened.
N_WRN_NOTEXISTDOC	11	Document does not exist.

List of error codes by functions (1/6)

Function name	Definition name	Defined value	Notes
NSetCallback	No return value	—	
NEnumPrinters	N_SUCCESS	0	Function return value
	N_ERR_NONE_PRTLIST	-21	Function return value
	N_ERR_ARGUMENT_02	-92	Function return value
	N_ERR_PRTNAME_ALLOC	-133	Function return value
NDeletePrinter	N_SUCCESS	0	Function return value
	N_WRN_PRTALREADYOPEN	10	Function return value
	N_ERR_PRTINFO_READ	-131	Function return value
	N_ERR_PRTINFO_NOTFOUND	-139	Function return value
	N_ERR_DEVICE_NOTSUPPORT	-150	Function return value
NRenamePrinter	N_SUCCESS	0	Function return value
	N_WRN_PRTALREADYOPEN	10	Function return value
	N_ERR_ARGUMENT_01	-91	Function return value
	N_ERR_ARGUMENT_02	-92	Function return value
	N_ERR_PRTRENAME_BEFORE	-134	Function return value
	N_ERR_PRTRENAME_AFTER	-135	Function return value
NGetPrinterInf	N_SUCCESS	0	Function return value
	N_ERR_ARGUMENT_01	-91	Function return value
	N_ERR_ARGUMENT_03	-93	Function return value
	N_ERR_PRTINFO_ILLEGAL	-137	Function return value
	N_ERR_PRTINFO_NOTFOUND	-139	Function return value
NGetPrinterFromID	Printer name size	+ value	Function return value
	N_ERR_ARGUMENT_01	-91	Function return value
	N_ERR_PRTINFO_NOTFOUND	-139	Function return value
NAutoOpen	Being abolished at Ver. 3.0.0.0	FALSE	Please DO NOT use this function.
NOpenPrinter	N_SUCCESS	0	Function return value - callback
	N_WRN_PRTALREADYOPEN	10	Function return value - callback
	N_ERR_HANDLE	-1	Function return value - callback
	N_ERR_PRTOPEN	-2	Function return value - callback
	N_ERR_FILEOPEN	-10	Function return value - callback
	N_ERR_NONE_PRTLIST	-21	Function return value - callback
	N_ERR_ARGUMENT_01	-91	Function return value
	N_ERR_UDPTHREADSTOPPED	-112	Function return value - callback
	N_ERR_PRTINFO_NOTFOUND	-139	Function return value - callback
	N_ERR_DEVICE_NOTSUPPORT	-150	Function return value - callback
NClosePrinter	N_SUCCESS	0	Function return value
	N_ERR_PRTCLOSE	-6	Function return value
	N_ERR_NONE_PRTLIST	-21	Function return value
	N_ERR_NOHANDLE	-22	Function return value
	N_ERR_ARGUMENT_01	-91	Function return value
	N_ERR_DEVICE_NOTSUPPORT	-150	Function return value
NClosePrinters	N_SUCCESS	0	Function return value
	N_ERR_NONE_PRTLIST	-21	Function return value

Win32 APIリファレンス

List of error codes by functions (2/6)

Function name	Definition name	Defined value	Notes
NPrint	N_SUCCESS	0	Function return value
	N_ERR_HANDLE	-1	Function return value
	N_ERR_FILEOPEN	-10	Function return value
	N_ERR_PRTOUTPUT	-13	Function return value
	N_ERR_NONE_PRTLIST	-21	Function return value
	N_ERR_NOHANDLE	-22	Function return value
	N_ERR_LACKRESOURCE	-31	Function return value
	N_ERR_LOADFROMFILE	-50	Function return value
	N_ERR_IMAGESIZE	-51	Function return value
	N_ERR_ARGUMENT	-90	Function return value
	N_ERR_ARGUMENT_01	-91	Function return value
	N_ERR_ARGUMENT_02	-92	Function return value
	N_ERR_ARGUMENT_03	-93	Function return value
NDPrint	N_SUCCESS	0	Function return value
	N_ERR_PRTOUTPUT	-13	Function return value
	N_ERR_NONE_PRTLIST	-21	Function return value
	N_ERR_NOHANDLE	-22	Function return value
	N_ERR_ARGUMENT_01	-91	Function return value
	N_ERR_ARGUMENT_02	-92	Function return value
	N_ERR_ARGUMENT_03	-93	Function return value
NImagePrint	N_SUCCESS	0	Function return value
	N_ERR_PRTOUTPUT	-13	Function return value
	N_ERR_NONE_PRTLIST	-21	Function return value
	N_ERR_NOHANDLE	-22	Function return value
	N_ERR_LACKRESOURCE	-31	Function return value
	N_ERR_IMAGESIZE	-51	Function return value
	N_ERR_ARGUMENT_01	-91	Function return value
	N_ERR_ARGUMENT_02	-92	Function return value
	N_ERR_ARGUMENT_03	-93	Function return value
	N_ERR_ARGUMENT_04	-94	Function return value
NImagePrintF	N_SUCCESS	0	Function return value
	N_ERR_PRTOUTPUT	-13	Function return value
	N_ERR_NONE_PRTLIST	-21	Function return value
	N_ERR_NOHANDLE	-22	Function return value
	N_ERR_LACKRESOURCE	-31	Function return value
	N_ERR_LOADFROMFILE	-50	Function return value
	N_ERR_IMAGESIZE	-51	Function return value
	N_ERR_ARGUMENT_01	-91	Function return value
	N_ERR_ARGUMENT_02	-92	Function return value
	N_ERR_ARGUMENT_03	-93	Function return value
	N_ERR_SEND_ERROR	-3	Function return value
NGetStatus	N_SUCCESS	0	Function return value
	N_ERR_SEND_ERROR	-3	Function return value

List of error codes by functions (3/6)

Function name	Definition Name	Defined value	Notes
NGetStatus	N_ERR_OFFLINE	-5	Function return value
	N_ERR_NOT_MAPPING	-11	Function return value
	N_ERR_NOT_OPEN_MAPFILE	-12	Function return value
	N_ERR_PRTOUTPUT	-13	Function return value
	N_ERR_LACKRESOURCE	-31	Function return value
	N_ERR_ARGUMENT_01	-91	Function return value
	N_ERR_ARGUMENT_02	-92	Function return value
NGetInformation	N_SUCCESS	0	Function return value
	N_ERR_NOT_MAPPING	-11	Function return value
	N_ERR_NOT_OPEN_MAPFILE	-12	Function return value
	N_ERR_PRTOUTPUT	-13	Function return value
	N_ERR_NONE_PRTLST	-21	Function return value
	N_ERR_NOHANDLE	-22	Function return value
	N_ERR_ARGUMENT_01	-91	Function return value
	N_ERR_ARGUMENT_02	-92	Function return value
	N_ERR_ARGUMENT_03	-93	Function return value
NResetPrinter	N_SUCCESS	0	Function return value - callback
	N_ERR_NONE_PRTLST	-21	Function return value
	N_ERR_RESETPRINTER	-60	Function return value - callback
	N_ERR_ARGUMENT_01	-91	Function return value
	N_ERR_DEVICE_NOTSUPPORT	-150	Function return value - callback
NStartDoc	N_SUCCESS	0	Function return value
	N_ERR_PRTOUTPUT	-13	Function return value
	N_ERR_NONE_PRTLST	-21	Function return value
	N_ERR_ALREADYSTARTDOC	-72	Function return value
	N_ERR_ARGUMENT_01	-91	Function return value
NEndDoc	N_SUCCESS	0	Function return value
	N_ERR_NONE_PRTLST	-21	Function return value
	N_ERR_LACKRESOURCE	-31	Function return value
	N_ERR_DOCNOTSTARTED	-71	Function return value
	N_ERR_ARGUMENT_01	-91	Function return value
NCancelDoc	N_SUCCESS	0	Function return value
	N_ERR_NONE_PRTLST	-21	Function return value
	N_ERR_DOCNOTSTARTED	-71	Function return value
	N_ERR_ARGUMENT_01	-91	Function return value
NEnumDoc	List saving size	+ value	Function return value
	N_ERR_NONE_PRTLST	-21	Function return value
	N_ERR_ARGUMENT_01	-91	Function return value
NDeleteDoc	N_SUCCESS	0	Function return value
	N_WRN_NOTEXISTDOC	11	Function return value
	N_ERR_NONE_PRTLST	-21	Function return value

List of error codes by functions (4/6)

Function name	Definition Name	Defined value	Notes
NDeleteDoc	N_ERR_ARGUMENT_01	-91	Function return value
	N_ERR_DEVICE_NOTSUPPORT	-150	Function return value
NBarcodeSettings	N_SUCCESS	0	Function return value
	N_ERR_ARGUMENT	-90	Function return value (10 th argument error)
	N_ERR_ARGUMENT_01	-91	Function return value
	N_ERR_ARGUMENT_02	-92	Function return value
	N_ERR_ARGUMENT_07	-97	Function return value
	N_ERR_ARGUMENT_08	-98	Function return value
	N_ERR_ARGUMENT_09	-99	Function return value
	N_ERR_BARCODE_OPTION	-165	Function return value
NBarcode	N_SUCCESS	0	Function return value
	N_ERR_NONE_PRTLIST	-21	Function return value
	N_ERR_LACKRESOURCE	-31	Function return value
	N_ERR_ARGUMENT_01	-91	Function return value
	N_ERR_ARGUMENT_02	-92	Function return value
	N_ERR_ARGUMENT_03	-93	Function return value
	N_ERR_ARGUMENT_04	-94	Function return value
	N_ERR_ARGUMENT_05	-95	Function return value
	N_ERR_ARGUMENT_06	-96	Function return value
	N_ERR_ARGUMENT_07	-97	Function return value
	N_ERR_ARGUMENT_08	-98	Function return value
	N_ERR_ARGUMENT_09	-99	Function return value
NBarcode2	N_SUCCESS	0	Function return value
	N_ERR_LACKRESOURCE	-31	Function return value
	N_ERR_ARGUMENT_01	-91	Function return value
	N_ERR_ARGUMENT_04	-94	Function return value
	N_ERR_ARGUMENT_05	-95	Function return value
	N_ERR_ARGUMENT_06	-96	Function return value
	N_ERR_ARGUMENT_07	-97	Function return value
	N_ERR_CREATEBCDDATA	-162	Function return value
	N_ERR_BARCODE_DATA_LONG	-163	Function return value
	N_ERR_BARCODE_DATA_INVALID	-164	Function return value
	N_ERR_BARCODE_OPTION	-165	Function return value
	N_ERR_BARCODE_BUFFER	-166	Function return value
	N_ERR_NO_LIBRARY	-900	Function return value
NFirmwareDL (Firmware check) (Firmware download)	N_SUCCESS	0	Function return value - callback
	N_ERR_OFFLINE	-5	Function return value - callback
	N_ERR_NONE_PRTLIST	-21	Function return value
	N_ERR_ARGUMENT_01	-91	Function return value
	N_ERR_ARGUMENT_02	-92	Function return value

List of error codes by functions (5/6)

Function name	Definition name	Defined value	Notes
NFirmwareDL (Firmware check) (Firmware download)	N_ERR_FWF_CHECKSUM	-81	Function return value
	N_ERR_FWCHK_TIMEOUT	-83	Function return value
	N_ERR_SEND_ERROR	-3	Function return value
	N_ERR_PRTOUTPUT	-13	Function return value – callback
	N_ERR_NOHANDLE	-22	Function return value – callback
	N_ERR_LACKRESOURCE	-31	Function return value – callback
	N_ERR_DOCNOTSTARTED	-71	Function return value – callback
	N_ERR_ALREADYSTARTDOC	-72	Function return value
	N_ERR_FWFFILE	-80	Function return value – callback
	N_ERR_FWDL_TIMEOUT	-82	Function return value – callback
	N_ERR_FOUNDERROR	-84	Function return value
NInitializeNetwork	N_SUCCESS	0	Function return value
	N_ERR_ARGUMENT_01	-91	Function return value
	N_ERR_UDPTHREADSTARTED	-111	Function return value
	N_ERR_UDPTHREADSTOPPED	-112	Function return value
	N_ERR_UDPTHREADSTOP	-113	Function return value
NScanPrinters	N_SUCCESS	0	Function return value – callback
	N_ERR_PRTOUTPUT	-13	Function return value – callback
	N_ERR_ARGUMENT_01	-91	Function return value
	N_ERR_UDPTHREADSTOPPED	-112	Function return value – callback
	N_ERR_PRTNAME_ALLOC	-133	Function return value – callback
	N_ERR_PRTINFO_NOTFOUND	-139	Function return value – callback
	Number of LAN printers	+ value	Function return value – callback
NTCPPortLock	N_SUCCESS	0	Function return value – callback
	N_ERR_PRTOUTPUT	-13	Function return value – callback
	N_ERR_NONE_PRTLIST	-21	Function return value – callback
	N_ERR_NOHANDLE	-22	Function return value – callback
	N_ERR_ARGUMENT_01	-91	Function return value – callback
	N_ERR_ARGUMENT_02	-92	Function return value – callback
	N_ERR_UDPTHREADSTOPPED	-112	Function return value – callback

List of error codes by functions (6/6)

Function name	Definition name	Defined value	Notes
NBufferClear	N_SUCCESS	0	Function return value – callback
	N_ERR_PRTOUTPUT	-13	Function return value – callback
	N_ERR_NONE_PRTLST	-21	Function return value – callback
	N_ERR_NOHANDLE	-22	Function return value – callback
	N_ERR_ARGUMENT_01	-91	Function return value – callback
	N_ERR_ARGUMENT_02	-92	Function return value – callback
	N_ERR_UDPTHREADSTOPPED	-112	Function return value – callback
	N_ERR_DEVICE_NOTSUPPORT	-150	Function return value – callback
NBlockSendSetting	N_SUCCESS	0	Function return value – callback
	N_ERR_PRTOUTPUT	-13	Function return value – callback
	N_ERR_NONE_PRTLST	-21	Function return value – callback
	N_ERR_NOHANDLE	-22	Function return value – callback
	N_ERR_ARGUMENT_01	-91	Function return value – callback
	N_ERR_ARGUMENT_02	-92	Function return value – callback
	N_ERR_UDPTHREADSTOPPED	-112	Function return value – callback
	N_ERR_DEVICE_NOTSUPPORT	-150	Function return value – callback

- In addition to the error codes described above, “N _ ERR _ LACKRESOURCE (-31)” is returned for “NPrintA” and other functions with a suffix A.

Class description (1/2)

Name space: NPrinterCLib

Class name: NClassLib

Type	Name	Remarks
Method	NEnumPrinters	Win32 API wrapper
Method	NGetPrinterFromID	Win32 API wrapper
Method	NDeletePrinter	Win32 API wrapper
Method	NRenamePrinter	Win32 API wrapper
Method	NGetPrinterInf	Win32 API wrapper
Method	NAutoOpen	Being abolished at Ver 3.0.0.0. Please DO NOT use this.
Method	NOpenPrinter	Win32 API wrapper
Method	NClosePrinter	Win32 API wrapper
Method	NClosePrinters	Win32 API wrapper
Method	NPrint	Win32 API wrapper
Method	NDPrint	Win32 API wrapper
Method	NImagePrint	Win32 API wrapper
Method	NImagePrintF	Win32 API wrapper
Method	NGetStatus	Win32 API wrapper
Method	NGetInformation	Win32 API wrapper
Method	NResetPrinter	Win32 API wrapper
Method	NStartDoc	Win32 API wrapper
Method	NEndDoc	Win32 API wrapper
Method	NCancelDoc	Win32 API wrapper
Method	NEnumDoc	Win32 API wrapper
Method	NDeleteDoc	Win32 API wrapper
Method	NBarcodeSettings	Win32 API wrapper
Method	NBarcode	Win32 API wrapper
Method	NBarcode2	Win32 API wrapper

Class description (2/2)

Name space: NPrinterCLib

Class name: NClassLib

Type	Name	Remarks
Method	NFirmwareDL	Win32 API wrapper
Method	NScanPrinters	Win32 API wrapper
Method	NTCPPortLock	Win32 API wrapper
Method	NBufferClear	Win32 API wrapper
Method	NBlockSendSetting	Win32 API wrapper
Method	NSetCallback	Win32 API wrapper
Method	NInitializeNetwork	Win32 API wrapper

Method name	NEnumPrinters		
Argument name	IN/OUT	Type	Description
o_printers	O	[MarshalAs(UnmanagedType.LPWStr)] StringBuilder	Printer name (NULL can be specified.)
o_size	O	out long	Size of o_printers bytes
Return value	Int		
Processing description			
Note			

Please refer to [Win32 API Reference] for Return value and Processing description.

Method name	NGetPrinterFromID		
Argument name	IN/OUT	Type	Description
i_ID	I	[MarshalAs(UnmanagedType.LPWStr)] string	Bluetooth serial ID
O_printer	I	[MarshalAs(UnmanagedType.LPWStr)] StringBuilder	Printer name (NULL can be specified.)
Return value	Int		
Processing description			
Note			

Please refer to [Win32 API Reference] for Return value and Processing description.

Method name	NDeletPrinter		
Argument name	IN/OUT	Type	Description
i_printers	I	[MarshalAs(UnmanagedType.LPWStr)] StringBuilder	Printer name
Return value	Int		
Processing description			
Note			

Please refer to [Win32 API Reference] for Return value and Processing description.

Method name	NRenamePrinter		
Argument name	IN/ OUT	Type	Description
i_beforeprt	I	[MarshalAs(UnmanagedType.LPWStr)] string	Printer name before being changed
i_afterprt	I	[MarshalAs(UnmanagedType.LPWStr)] string	Printer name after being changed
Return value	Int		
Processing description			
Note			

Please refer to [Win32 API Reference] for Return value and Processing description.

Method name	NGetPrinterInf		
Argument name	IN/ OUT	Type	Description
i_prt	I	[MarshalAs(UnmanagedType.LPWStr)] string	Printer name
o_ports	O	MarshalAs(UnmanagedType.LPWStr)] StringBuilder	Port information
o_size	O	out long	Size of o_ports bytes
Return value	Int		
Processing description			
Note			

Please refer to [Win32 API Reference] for Return value and Processing description.

Method name	NAutoOpen		
Argument name	IN/OUT	Type	Description
i_flg	I	int	Auto printer open flag 0 : Do not open automatically (Manually open) 1 : Open automatically Other: Only to obtain the status (Not setting)
Return value	Bool		
Processing description			
Note			
This function was abolished at Ver.3.0.0.0. The return value will be always “false”.			

Method name	NOpenPrinter		
Argument name	IN/OUT	Type	Description
i_prt i_statusFlg	I I	[MarshalAs(UnmanagedType.LPWStr)] string bool	Printer name to open Status receiving thread activating flag
Return value	Int		
Processing description			
Note			

Please refer to [Win32 API Reference] for Return value and Processing description.

Method name	NClosePrinter		
Argument name	IN/OUT	Type	Description
i_prt	I	[MarshalAs(UnmanagedType.LPWStr)] string	Printer name
Return value	Int		
Processing description			
Note			

Please refer to [Win32 API Reference] for Return value and Processing description.

Method name	NClosePrinters		
Argument name	IN/OUT	Type	Description
Return value	Int		
Processing description			
Note			

Please refer to [Win32 API Reference] for Return value and Processing description.

Method name	NPrint		
Argument name	IN/ OUT	Type	Description
i_prt i_dat	I I	[MarshalAs(UnmanagedType.LPWStr)] string byte[]	Target printer name Transmitting data (Hexadecimal character string)
i_size io_jobid	I IO	uint out long	Number of bytes to be output Print job ID (NULL can be specified.)
Return value	Int		
Processing description			
Note			

Please refer to [Win32 API Reference] for Return value and Processing description.

Method name	NDPrint		
Argument name	IN/OUT	Type	Description
i_prt	I	[MarshalAs(UnmanagedType.LPWStr)] string	Target printer name
i_dat	I	byte[]	Transmitting data (Hexadecimal character string)
i_size	I	uint	Number of bytes to be output
io_jobid	IO	out long	Print job ID (NULL can be specified.)
Return value	Int		
Processing description			
Note			

Please refer to [Win32 API Reference] for Return value and Processing description.

Method name	NImagePrint		
Argument name	IN/OUT	Type	Description
i_prt	I	[MarshalAs(~LPWStr)] string	Target printer name
i_bmp	I	IntPtr	Device context handle
i_width	I	int	Width
i_height	I	int	Height
i_putType	I	byte	Output system 0x00: Raster form line unit 0x01: Raster form block unit 0x02: Raster form block unit with gray scale expression 0x10: Bit image form
io_jobid	IO	out long	Print job ID (NULL can be specified.)
Return value	Int		
Processing description			
Note			

Please refer to [Win32 API Reference] for Return value and Processing description.

Method name	NImagePrintF		
Argument name	IN/OUT	Type	Description
i_prt	I	[MarshalAs(~LPWStr)] string	Target printer name
i_bmp	I	[MarshalAs(~LPWStr)] string	BMP file name
i_putType	I	byte	Output system 0x00: Raster form line unit 0x01: Raster form block unit 0x02: Raster form block unit with bray scale expression 0x10: Bit image form
io_jobid	IO	out long	Print job ID (NULL can be specified.)
Return value	Int		
Processing description			
Note			

Please refer to [Win32 API Reference] for Return value and Processing description.

Method name	NGetStatus		
Argument name	IN/OUT	Type	Description
i_prt o_status	I O	[MarshalAs(UnmanagedType.LPWStr)] string out long	Printer name Status
Return value	Int		
Processing description			
Note			
- Please use this function after setting the Printer Name property.			

Please refer to [Win32 API Reference] for Return value and Processing description.

Method name	NGetInformation		
Argument name	IN/ OUT	Type	Description
i_prt	I	[MarshalAs(UnmanagedType.LPWStr)] string	Printer name
i_id	I	byte	Classification ID
o_dat	O	byte[]	Extended information storing area
o_time	O	Out long	Update flag (elapsed time from system activation) (NULL can be specified.)
Return value	Int		
Processing description			
Note			

Please refer to [Win32 API Reference] for Return value and Processing description.

Method name	NResetPrinter		
Argument name	IN/OUT	Type	Description
i_prt	I	[MarshalAs(UnmanagedType.LPWStr)] string	Printer name
Return value	Int		
Processing description			
Note			

Please refer to [Win32 API Reference] for Return value and Processing description.

Method name	NStartDoc		
Argument name	IN/OUT	Type	Description
i_prt o_jobid	I O	[MarshalAs(UnmanagedType.LPWStr)] string out long	Printer name Job ID (NULL can be specified.)
Return value	Int		
Processing description			
Note			

Please refer to [Win32 API Reference] for Return value and Processing description.

Method name	NEndDoc		
Argument name	IN/OUT	Type	Description
i_prt	I	[MarshalAs(UnmanagedType.LPWStr)] string	Printer name
Return value	Int		
Processing description			
Note			

Please refer to [Win32 API Reference] for Return value and Processing description.

Method name	NCancelDoc		
Argument name	IN/OUT	Type	Description
i_prt	I	[MarshalAs(UnmanagedType.LPWStr)] string	Printer name
Return value	Int		
Processing description			
Note			

Please refer to [Win32 API Reference] for Return value and Processing description.

Method name	NEnumDoc		
Argument name	IN/OUT	Type	Description
i_prt o_docno list	I O	[MarshalAs(UnmanagedType.LPWStr)] string [MarshalAs(UnmanagedType.LPWStr)] StringBuilder	Printer name Document number list (NULL can be specified.)
Return value	Int		
Processing description			
Note			

Please refer to [Win32 API Reference] for Return value and Processing description.

Method name	NDeleteDoc		
Argument name	IN/OUT	Type	Description
i_prt	I	[MarshalAs(UnmanagedType.LPWStr)] string	Printer name
i_docno	I	int	Document number
Return value	Int		
Processing description			
Note			

Please refer to [Win32 API Reference] for Return value and Processing description.

Method name	NBarcodeSettings		
Argument name	IN/OUT	Type	Description
i_id	I	uint	Barcode type
i_width	I	uint	Magnification of data width
i_height	I	uint	Height of one module
i_option1	I	char	Option 1
i_option2	I	char	Option 2
i_option3	I	char	Option 3
i_option4	I	char	Option 4
i_hri	I	uint	HRI characters position
i_spin	I	uint	Rotation
i_code	I	uint	Character code type
Return value	Int		
Processing description			
Note			

Please refer to [Win32 API Reference] for Return value and Processing description.

Method name	NBarcode		
Argument name	IN/OUT	Type	Description
i_prt	I	[MarshalAs(UnmanagedType.LPWStr)] string	Printer name
i_fontName	I	[MarshalAs(UnmanagedType.LPWStr)] string	Font name
i_bmp	IO	IntPtr	Device context handle
i_x	I	uint	Left
i_y	I	uint	Top
i_width	I	uint	Width
i_height	I	uint	Height
i_dat	I	byte[]	Barcode data
i_size	I	uint	Data size
Return value	Int		
Processing description			
Note			
Please refer to [Win32 API Reference] for Return value and Processing description.			

Method name	NBarcoe2		
Argument name	IN/OUT	Type	Description
i_bmp	IO	IntPtr	Device context handle
i_x	I	uint	Left
i_y	I	uint	Top
io_width	IO	out uint	Width
io_height	IO	out uint	Height
i_dat	I	byte[]	Barcode data
i_size	I	uint	Data size
Return value	Int		
Processing description			
Note			
Please refer to [Win32 API Reference] for Return value and Processing description.			

Method name	NFirmwareDL		
Argument name	IN/OUT	Type	Description
i_prt	I	[MarshalAs(UnmanagedType.LPStr)] string	Target printer name
i_file	I	[MarshalAs(UnmanagedType.LPStr)] string	fwf file name (NULL can be specified.)
i_errflg	I	byte	Error check 0x00: Invalid (forced transmission) 0x01: Valid
io_chksum	O	out short	fwf File checksum
io_jobid	O	out long	Print job ID (NULL can be specified.)
Return value	Int		
Processing description			
Note			

Please refer to [Win32 API Reference] for Return value and Processing description.

Method name	NScanPrinters		
Argument name	IN/OUT	Type	Description
i_waitmsec	I	UInt	Number of milliseconds to wait for enumeration response and data creation from the printer enumeration request
Return value	Int		
Processing description			
Note			

Please refer to [Win32 API Reference] for Return value and Processing description.

Method name	NTCPPortLock		
Argument name	IN/OUT	Type	Description
i_prt	I	[MarshalAs(UnmanagedType.LPStr)] string	Printer name
i_type	I	byte	Connection type
Return value	Int		
Processing description			
Note			

Please refer to [Win32 API Reference] for Return value and Processing description.

Method name	NBufferClear		
Argument name	IN/OUT	Type	Description
i_prt	I	[MarshalAs(UnmanagedType.LPStr)] string	Printer name
Return value	Int		
Processing description			
Note			

Please refer to [Win32 API Reference] for Return value and Processing description.

Method name	NBlockSendSetting		
Argument name	IN/OUT	Type	Description
i_prt	I	[MarshalAs(UnmanagedType.LPStr)] string	Target printer name
i_type	I	byte	Setting type
Return value	Int		
Processing description			
Note			

Please refer to [Win32 API Reference] for Return value and Processing description.

Method name	NSetCallback		
Argument name	IN/OUT	Type	Description
i_callback	I	NClassLib.NCALLBACK	Callback function pointer
Return value	Int		
Processing description			
Note			

Please refer to [Win32 API Reference] for Return value and Processing description.

Method name	NInitializeNetwork		
Argument name	IN/OUT	Type	Description
i_flg	I	int	Setting type
Return value	Int		
Processing description			
Note			

Please refer to [Win32 API Reference] for Return value and Processing description.