

SDK guide

Overview

Describes overview of SDK

Win32 API Reference

Describes syntax of API

.net API Reference



Describes syntax of API used in .net environment

Document disclaimer

- The description in this document is taken all possible measures to ensure the correctness, however if you find any, please contact us.
- Descriptions are subject to change without prior notice.
Please ask for up-to-date information.
- We are firmly declining about reproduction, copying and a thing tampering with in some or all of the contents of this book without permission.
- We are not responsible for any influence by the results from the use.
- We assume no responsibility whatsoever for any damages resulting from improper use, the use without understanding of this description, repair and change by the third party.

Signs

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

| | |
|--|---|
|  Notice | It indicates the precautions to be observed fully. Disregard of the signs and wrong usage may cause malfunction and inoperative. |
|  Reference | It indicates supplement description and relevant matters. |

Usage restrictions

In the event that this product is used with devices that require high reliability and safety for function and accuracy on conveyance including aircraft, train, vehicle; disaster prevention and security device, users shall use products after considering the safety design of whole system by applying fail-safe and redundancy design to maintain reliability and safety.

This product is not designed to use with devices that require extremely high reliability and safety including aerospace mechanism, signal axis mechanism, nuclear control device, medical device. Users shall ascertain and evaluate suitability of this product for those application.

Table of contents

- Overview
 - [System configuration with using SDK](#) 6
- Various setting
 - [Installation](#) 7
 - [How to use it at Bluetooth or Serial connection](#) 8
 - [Notes at direct communication](#) 8
 - [Storage location of the temporary file](#) 8
 - [About fire wall](#) 9
 - [About log output](#) 11
 - [About barcode setting](#) 14
- Win32 API References
 - [Category of API](#) 16
 - Methods
 - [Ncallback \(Interface\)](#) 17
 - [NSetCallback](#) 18
 - [NEnumPrinters](#) 19
 - [NGetPrinterFromID](#) 21
 - [NDeletePrinter](#) 22
 - [NRenamePrinter](#) 23
 - [NGetPrinterInf](#) 24
 - [NAutoOpen](#) 25
 - [NOpenPrinter](#) 26
 - [NClosePrinter](#) 27
 - [NClosePrinters](#) 28
 - [NPrint](#) 29
 - [NDPrint](#) 30
 - [NImagePrint](#) 31
 - [NImagePrintF](#) 32
 - [NGetStaus](#) 33
 - [NGetInformation](#) 34
 - [NResetPrinter](#) 35
 - [NStartDoc](#) 36
 - [NEndDoc](#) 37

- Win32 API Reference

| | |
|--|----|
| -NCancelDoc | 38 |
| -NEnumDoc | 39 |
| -NDeleteDoc | 40 |
| -NBarcode | 41 |
| -NFirmwareDL | 43 |
| -NInitializeNetwork | 44 |
| -NScanPrinters | 45 |
| -NTCPPortLock | 47 |
| -NBufferClear | 48 |
| -NBlockSendSetting | 49 |
| -Error code table | 50 |
| -API-Specific error code table | 52 |
| -Extended Information | 56 |

- .net API Reference

| | |
|---|----|
| -Class description(NClassLib) | 58 |
| Method | |
| -NEnumPrinters | 59 |
| -NGetPrinterFromID | 59 |
| -NDeletePrinter | 60 |
| -NRenamePrinter | 60 |
| -NGetPrinterInf | 61 |
| -NAutoOpen | 61 |
| -NOpenPrinter | 62 |
| -NClosePrinter | 62 |
| -NClosePrinters | 63 |
| -NPrint | 63 |
| -NDPrint | 64 |
| -NImagePrint | 64 |
| -NImagePrintF | 65 |
| -NGetStaus | 65 |
| -NGetInformation | 66 |
| -NResetPrinter | 66 |
| -NStartDoc | 67 |

- .net API Reference

| | |
|-------------------------------------|----|
| -NEndDoc | 67 |
| -NCancelDoc | 68 |
| -NEnumDoc | 68 |
| -NDeleteDoc | 69 |
| -NBarcode | 69 |
| -NFirmwareDL | 70 |
| -NScanPrinters | 70 |
| -NTCPPortLock | 71 |
| -NBufferClear | 71 |
| -NBlockSendSetting | 72 |
| -NSetCallback | 72 |
| -NInitializeNetwork | 73 |

Overview

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

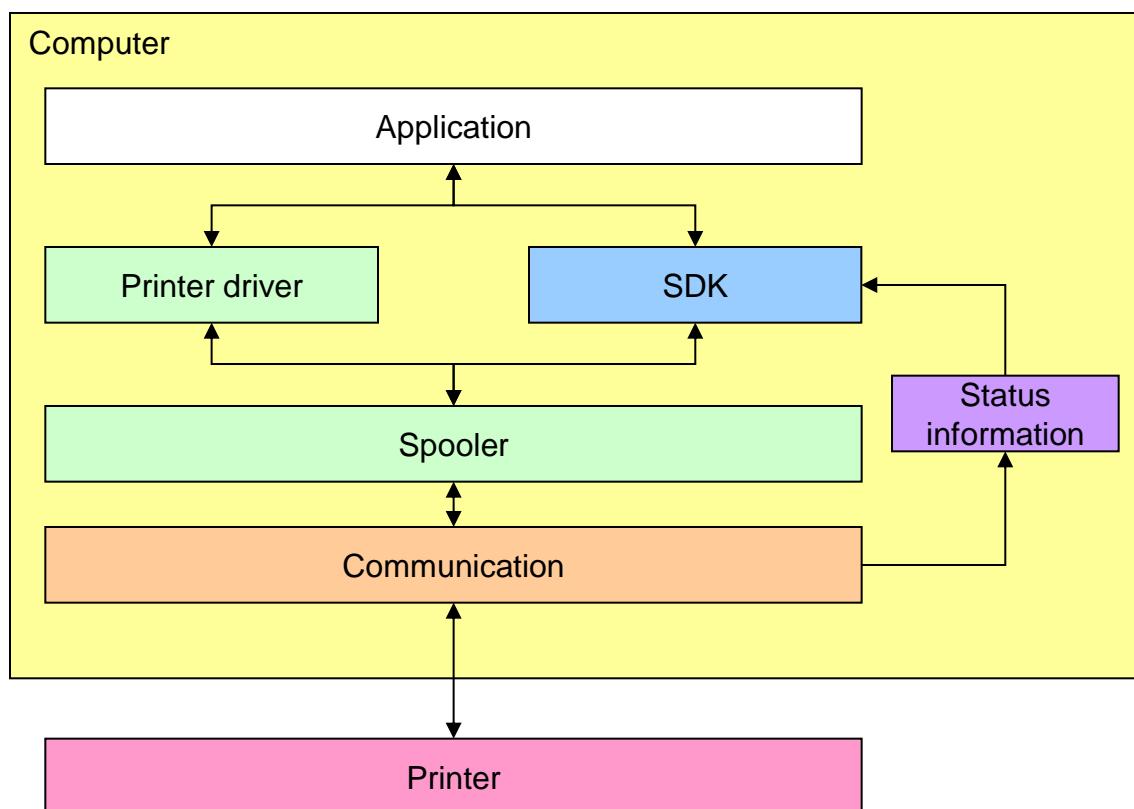
SDK is distributed as Dynamic Link Library (DLL) and various of Executable files (EXE).

File name : NPrinterLib.dll, BarCodeLib.dll, NPrinterCLib.dll

NServiceDrv.exe, NServiceApp.exe

* When you use C# application, please refer to NPrinterCLib.dll.

System configuration with SDK



Development language

Win32 : Visual C++

.NET : Visual C#

Interface

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

Installation

When you use it only by the library.

1. Please install NPrinterLib.dll, NPrinterCLib.dll. in the folder that can refer from the application to use
Please save BarCodeLib.dll into System32 folder.
Please refer to NPrinterCLib.dll when you use C# application.
2. Please register NServiceDrv.exe as service.
<32bit Environment>
 - Save the file in the System32 folder.
 - Activate command prompt by administrator rights and execute the command below,
sc create NServiceDrv binpath= "C: ¥Windows¥system32¥NServiceDrv.exe"
<64bit Environment>
 - Save the file in the SysWOW64 folder.
 - Activate command prompt by administrator rights and execute the command below,
sc create NServiceDrv binpath= "C: ¥Windows¥Sysnative¥NServiceDrv.exe"
3. Start each service that you already have registered.
 - Open [Windows management tools]-[Service].
 - Change the property of NServiceDrv [Type of Start-up] to [Automatic]
 - Execute [Start of the service]
4. Save NServiceAPP.exe into the same folder of NServiceDrv, and register it as OLE saver.
 - Activate command prompt by administrator rights and execute the command below,
C: ¥Windows¥System32¥NServiceAPP.exe /regserver
 - * To use in 64bit Environment, Please execute as below,
C: ¥Windows¥Sysnative¥NServiceAPP.exe /regserver

When you use it with driver.

When you set up the printer driver, it is installed automatically,
so you do not need any special setting.

For set-up of the driver, please refer to the [Installation guide] of the driver.

Reference

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

How to use

After installation completion, include ImportApi.h in the development environment.

And after that, read dll by Load library function.

When the function succeeds, the handle of the module is returned. And pass the handle to GetProcAddress function, you will get the address of the DLL function.

For more details, please refer to execution part of the LoadLibrary in sample program, [Document/API_Sample]. ImportApi.h is included in the sample.

About Bluetooth connection,

For this SDK, it is necessary to finish the pairing of PC and the printer, before connecting in Bluetooth.

About serial connection

For this SDK, it is necessary to complete each port setting, before connecting in serial port..

Notes for direct communication

When you make direct communication of USB, Serial, Bluetooth of this SDK with the PC which the printer driver was set up, please stop the printer spooler service of the Windows management tool.

Storage location of the temporary file

At the default state, NPI folder is made right under an OS setup drive.

To change the storage location, make a key below in registry, under that, make a DirPath as String value and designate a folder that you use.

HKEY_LOCAL_MACHINE¥SOFTWARE¥NPI

E.g.

[HKEY_LOCAL_MACHINE¥SOFTWARE¥NPI]

"DirPath"="d:¥¥Printer"

About firewall (1/2)

Firewall permission is necessary for NServiceDrv.exe and NServiceApp, because of the program to perform UDP communication, beforehand.

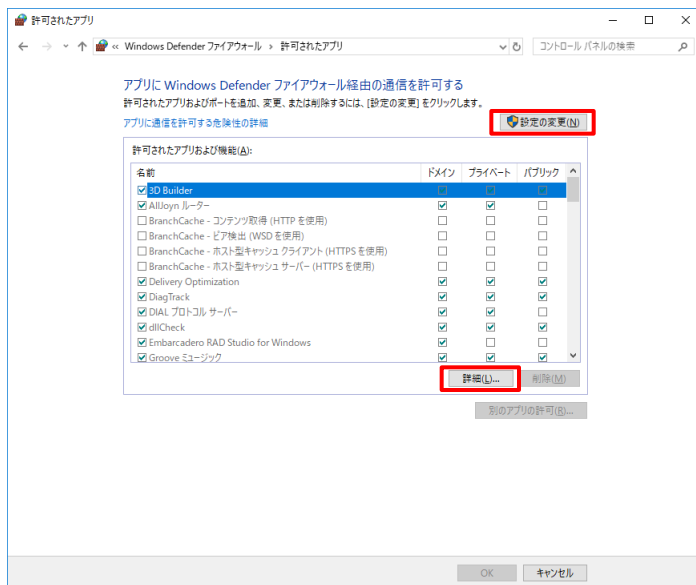
*At the time of the USB, serial and Bluetooth connection, the upper setting is unnecessary.

1)



Select [Control panel] – [Firewall] – [Permission of application or the function]

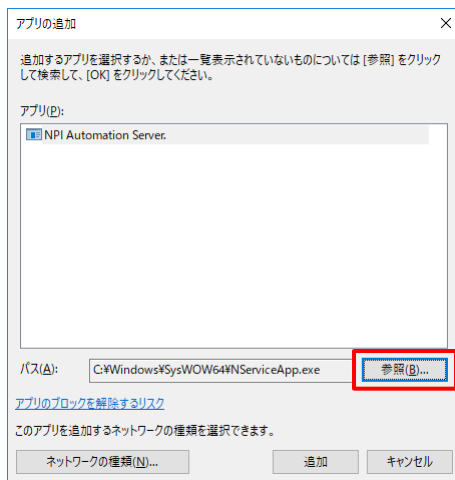
2)



Push the button in order of [Change of the setting] – [Advanced (L)].

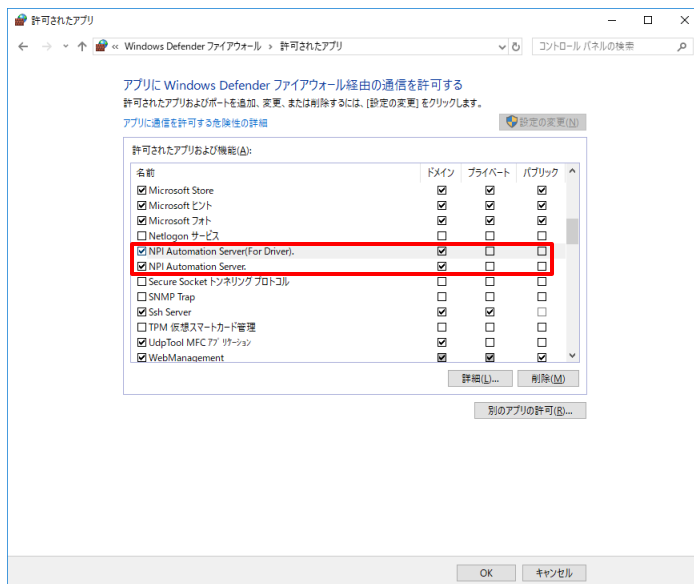
About firewall (2/2)

3)



Designate the pass of NserviceApp in [Reference] and permit the application to push the [Add] button. For NServiceDrv, permit it as the same way above.

4)



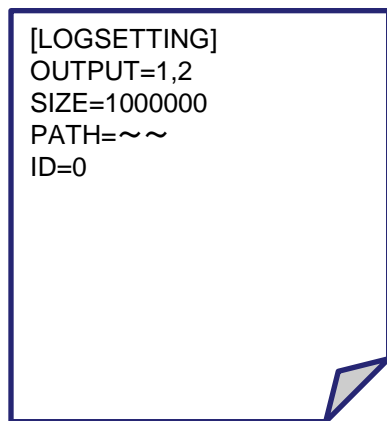
Please confirm that NServiceDrv and NServiceApp were added.

About log output (Output setting)

The SDK outputs log by a name SDKLog.txt in the designated folder.

In the log output, it is necessary to make a log setting file (OS Drive¥NPI¥NLogInf.inf).

(When a file does not exist, it does not output log)



It describes the setting file in the following form.

[LOGSETTING] <- Fixed
OUTPUT=1,2 <- Divide the log type that you want to output with a comma and designate it.
 (For an e.g. of the left, it outputs 1, 2 : ERROR, WARN)
SIZE=1000000 <- It designate maximum writing file size of the log.
PATH=C: ¥NPI¥log <- It designate the writing folder of the log.
ID=0 <- Execute output setting of the process ID, thread ID.
 (0:There is no output, 1: There is the output)

About Log type (OUTPUT)

- 1 ERROR : Error
- 2 WARN : Warning
- 3 FUNC : Function call (Except Status·Expanded information check)
- 4 IN : Port reception data
- 5 OUT : Port transmission data
- 6 CHK : Status·Expanded information check
- 7 PST : At the time Printer status is changing
or reception of expanded information.

*It outputs the log that along with the designated type.

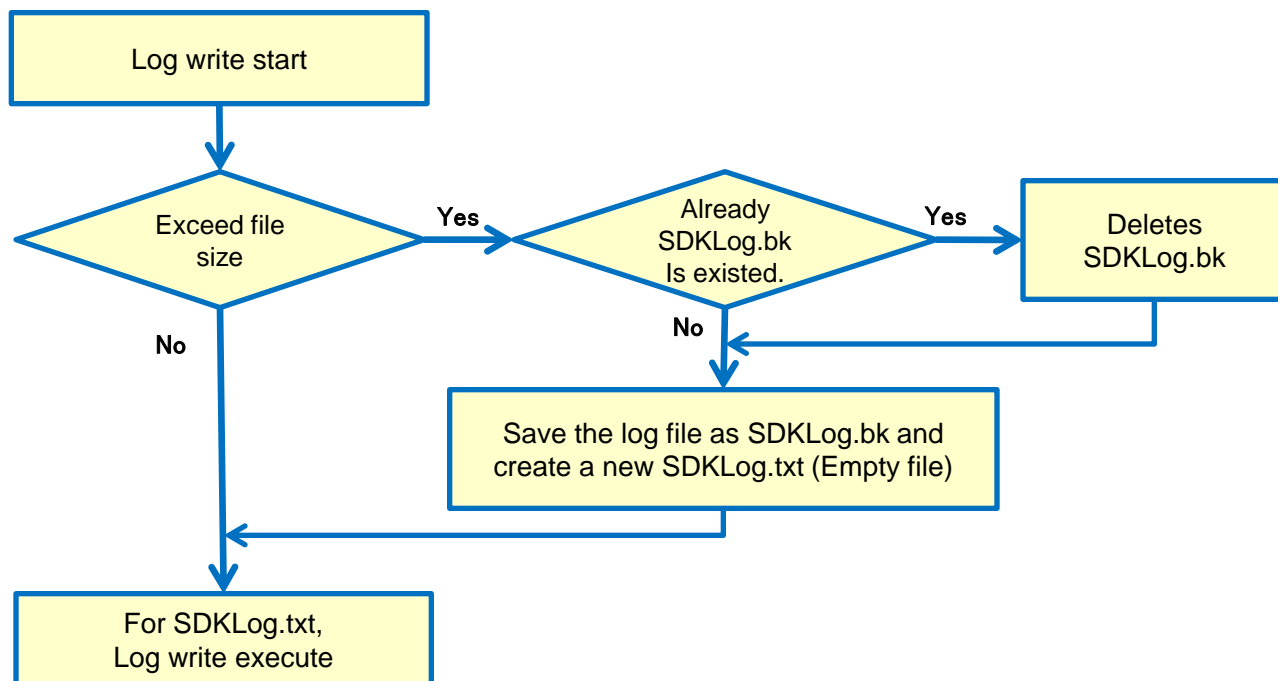
About log output (Pre cautions)

- *1 It is not reacted promptly even if you edit the file.
After generate NPrinterLib class, it will be reacted.
- *2 DO NOT add space before/after “=” .
- *3 When any setting is not made to OUTPUT, it does not OUTPUT the log.
- *4 It discards the value when the value of the log type is incorrect.
(The value except 1-7, Character string)
If you designate the incorrect value of log size
(Negative value of size, Character string) it use default value (1MB).
- *5 About log type4(IN) and 5(OUT), when there is much quantity of transfer of data, movement of SDK may become slow. (It takes time to output the log.)
- *6 You may not be able to open a log file when you designate the file size too big.
Then, please change the file size depending on the performance of the terminal.

About log output (Output file)

When data size of output log exceeds size appointed with the setting file, it saves it as the past log (SDKLog.bk) and make log in a new empty file.

It deletes SDKLog.bk, when it already exists. (As for the past log, only 1 file is kept.)



[Examples of Log outputs]

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.

It outputs the date/time as
YYYY/MM/DD HH:MM:SS.SSS [Log type]

Log type

1. ERR : Error
2. WRN : Warning
3. FNC : Calling function
4. I N : Port reception data
5. OUT : Port transmission data
6. CHK : Status·Expanded information check
7. PSK : Receives an expanded information when a printer status changes

When it set as ID=1 by log setting file, it adds the information of
<Process ID, Thread ID> after date/time.

Barcode setting

1. The allocation of the Barcode setting files

You need to create the setting file to us “NBarcode” function.
Please create the file by yourself.

Barcode file name : NPIREG.inf

The files refer the folders below,

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

Please create the folder after NPI, by your side.

*Registry setting that previously mentioned, the part of OSDrive would be changed.

Barcode Setting

2. How to write the Barcode setting file (1D Barcode)

Please set the Barcode1~Barcode10 as following format.

Beware of that, each items are case-sensitive.

| | | | | |
|-------------------------|---|-------------|-------------|-----------|
| "Kind"=dword:00000000 | Default value when there is not NPIREG.inf file in a target folder | | | |
| "Width"=dword:00000001 | Kind | :0 | HRI | :0 |
| "Height"=dword:000000a2 | Width | :1 | Spin | :0 |
| "StartB"=dword:00000000 | Height | :162 | | |
| "StopB"=dword:00000000 | StartB | :0 | | |
| "HRI"=dword:00000000 | StopB | :0 | | |
| "Spin"=dword:00000000 | | | | |

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 **Height** data height (dot)

StartB, StopB (Only for 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 the Barcode setting file (2D Barcode)

Please set the Barcode1~Barcode10 as following format.

Beware of that, each items are case-sensitive.

| | | | | |
|-------------------------|---|-----------|-------------|-----------|
| "Kind"=dword:00000000 | Default value when there is not NPIREG.inf file in a target folder | | | |
| "Size"=dword: 00000000 | Kind | :0 | Spin | :0 |
| "Height"=dword:00000001 | Size | :0 | | |
| "ECL"=dword:00000001 | Height | :1 | | |
| "ECP"=dword:00000000 | ECL | :1 | | |
| "Spin"=dword:00000003 | ECP | :1 | | |

Kind 0:QRCode Model2 1:PDF417

Size (data width magnification)

< QR code model2 > 0: 2 times 1: 3 times 2: 4 times 3: 5 times 4: 6 times 5: 7 times 6: 8 times

< PDF417> 0:2 times 1: 3 times 2: 4 times

HEIGHT (Only for PDF417) Designate the Height ratio as 1~6. HEIGHT = Height ratio x WIDTH

ECL (error level)

< QR code 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

Following APIs are available.

| Application | API | Description |
|----------------------------------|-------------------|---|
| Designate callback function | NSetCallback | Designate Callback function. |
| Obtaining a list of printer name | NEnumPrinters | Obtains a list of printer names managed by this SDK |
| Obtaining printer name | NGetPrinterFromID | Obtains a printer name from various ID |
| Deleting printer name | NDeletePrinter | Deletes a printer name managed by this SDK |
| Changing Printer name | NRenamePrinter | Changes a printer name managed by this SDK |
| Obtaining Printer Information. | NGetPrinterInf | Obtains information from printer name, |
| Automatic printer setting | NAutoOpen | (This function is obsolete from Ver.3.0.0.0) |
| Open printer | NOpenPrinter | Designate about Printer name and open it |
| Close printer | NClosePrinter | Close the printer which has been opened, |
| Close all the printer. | NClosePrinters | Close the all of the printers which have been opened. |
| Sending command and data | NPrint | Sends specified hexadecimal character data to printer. |
| Sending command and data | NDPrint | Sends specified data to printer |
| Image output | NImagePrint | Sends specified raster image device context to printer. |
| Image output | NImagePrintF | Sends specified raster image file (bmp/jpg/png) to printer. |
| Obtaining status | NGetStatus | Returns status obtained from specified printer. |
| Obtaining extended information | NGetInformation | Obtains information from specified type ID. |
| Reset printer | NResetPrinter | Resets printer (USB, TCP/UDP) |
| Document control | NStartDoc | Starts document. |
| Document control | NEndDoc | Ends document. |
| Document control | NCancelDoc | Cancels document. |
| Document control | NEnumDoc | Obtains the list of documents at the transmission waiting state |
| Document control | NDeleteDoc | Deletes the document awaiting transmission. |
| Generate barcode | NBarcode | Generate barcode image. |
| F/W upgrade | NFirmwareDL | Upgrade firmware with the target firmware file. |

Win32 API Reference

| | | | | | |
|---|--------|--------------|--------------------|----------------------------|------------------|
| Interface name | | NCallback | | | |
| Argument name | IN/OUT | Type | Description | | |
| i_prt | I | PCHAR | Printer name | | |
| i_type | I | int | Call back Type | | |
| i_value1 | I | int | Execution result 1 | | |
| i_value2 | I | int | Execution result 2 | | |
| Return value | void | | | | |
| ·None | | | | | |
| Processing Description | | | | | |
| <p>·The function format is callback notification.</p> <p>It declares with the same return value·argument to this declaration, and designate its function pointer to NSetCallback function.</p> <p>When you use Asynchronous function, you will obtain the result by call back.</p> <p>Beforehand, please execute NSetCallback function, definitely.</p> <p>The conditions of the callback notifications are as below.</p> | | | | | |
| Callback conditions | | i_prt | i_type | i_value | i_value2 |
| When a status value changed (Including the first reception) | | Printer name | 1 | Old status value | New status value |
| When it received expansion information | | Printer name | 2 | Expanded information ID | 0x00 |
| When submission data queue emptied | | Printer name | 3 | 0x00 | 0x00 |
| NScanPrinters result notification | | NULL | 4 | Execution result | Detection number |
| NOpenPrinter result notification | | Printer name | 5 | Execution result | 0x00 |
| NResetPrinter result notification | | Printer name | 6 | Execution result | 0x00 |
| NFirmwareDL 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 |
| NBlockSendSetting result notification | | Printer name | 10 | Execution result | 0x00 |

| | | | |
|--|---------------|-----------|--|
| Function name | NSSetCallback | | |
| Argument name | IN/OUT | Type | Description |
| i_callback | I | NCallback | The class which implemented callback interface |
| Return value | Void | | |
| ·None | | | |
| Processing Description | | | |
| <div>·Set up the class which implemented NCallback interface . Execute this function before execute the callback notification, In addition, you can perform the interface setting in function “NOpenPrinter”.</div> <div><div>Notice</div><div><p>It continues sending callback notification in the class that you set before, until you designate null by this function. Please note that this SDK generate an exception when the class instance which you set has been already deleted.</p><p>Please implement abstract Method of the interface in the class side. It is impossible to perform the screen drawing processing from the callback function.</p></div></div> | | | |

| | | | |
|--|--------------------------------|----------------|--|
| Function name | NEnumPrinters / NEnumPrintersA | | |
| Argument name | IN/OUT | Type | Description |
| o_printers | O | PWCHAR / PCHAR | Printer name (csv_To enumerate it in |
| o_size | O | PDWORD | a Comma Separated Value form) Byte size of o_printers |
| Return value | INT | | |
| ·Error (negative value), Normal end (0) * Refer to error code table | | | |
| Processing description | | | |
| <ul style="list-style-type: none">·To create the printer information management file (npidport.inf) under home directly, and store the table of connectable printers in the Argument o printers. E.g.). PRT001, PRT002,AAA <p>To detect the available ports, assign the printer name as “PRTxxx” (xxx is 001 ~999) and return them by csv (Comma Separated Value form)</p> <p>* Only the active printers are enumerated.</p> <p>Before to generate the printer information management file, it can not execute printer open (NOpenPrinter). Please make sure to call this function at first.</p> <p>You need to call this function again to add the printer.</p> <p>Once you generate the printer name, and use it, you do not have to call this function again.</p> <p>Printer table that is obtained by this function is also obtained to refer the printer information management file (npidport.inf) directly.</p> <p>* When you use this function via Bluetooth, before that, please make a paring with the printer.</p> <p>* Once the printer name is generated, it will not delete to release paring. To delete it, please use the function NDeletePrinter.</p> <p>* It can assign the printer 999 at maximum. More than that, it will be generation error. Then please the Printer name no use by NDeletePrinter or rename it by NRenamePrinter.</p> | | | |

| | |
|------------------------|--|
| Function name | NEnumPrinters /NEnumPrintersA |
| Processing Description | <p>•When there is a printer used with driver, then, the printer name of the direct communication printer is also stored at o printers. (1 printer is stored in both Driver and Direct Communication.)</p> <p>E.g.) NPI Integration Driver, PRT001, PRT002, PRT005 If there is Driver (USB) Printer A, the both of bold texts above will be Printer A.</p> <div>Notice</div> <p>* You can not use NPI Integration Driver to open both of them at the same time. Please open it after closing the other when you use it.</p> |

| | | | |
|---|--|----------------|---------------------|
| Function name | NGetPrinterFromID | | |
| Argument name | IN/OUT | Type | Description |
| i_ID | I | PWCHAR / PCHAR | Bluetooth serial ID |
| o_printer | O | PWCHAR / PCHAR | Printer name |
| Return value | int | | |
| ·Error (0), Successful termination(The size of printer name storage area) | | | |
| Processing description | <p>·To designate Serial ID to in the i_ID of the argument, to store Printer name to the argument o_printer. You can check the each of ID to execute self-printing of the printer.</p> <p><Each of ID> WLAN: MAC address Bluetooth: Serial ID USB: Vender ID 4-digit + Product ID 4-digit</p> <div>Notice</div> <p>Printer name must be generated in the Printer information management file (npidport.inf) by NEnumPrinters function.</p> | | |

| | | | |
|--|----------------|---|---------------------|
| Function name | NDeletePrinter | | |
| Argument name | IN/OUT | Type | Description |
| i_prt | l | PWCHAR / PCHAR | Delete Printer name |
| Return value | int | | |
| ·Error (negative value), Normal end (0) * Refer to error code table | | | |
| Processing description | | <p>•To use to delete the printer name in the Printer information management file. Designate printer name to delete on the argument ,i_prt. Or designate null character, you can delete the Printer information management file. (All the Printer names are deleted.)</p> <p>When the printer which you want to delete is opened, it returns an error. Pleas use this function after you disconnect and close the printer.</p> <p>After to execute this function, please adapt changes by the function NEnumPrinters.</p> <div>Notice</div> <p>It is impossible to change the driver name, because of it is not managed by this SDK.</p> | |

| | | | |
|---------------|----------------------------------|----------------|--------------------------------|
| 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 | PCHAR | | |

·Error (negative value), Normal end (0) * Refer to error code table

Processing description

To use to rename the printer name in the Printer information management file.
As default the printer name is starting with PRT~ (E.g. "PRT001"), but to use this function, you can change the printer name as you like it.

Printer name is less than 50 characters in half-width alphanumeric,
and the characters below are banned.

Ban on use letters (including en space) 「¥/: ?*"<>|', .

Use this function to connect to close the printer. If the printer to rename is opening,
then, an error message will be returned.

After executing this function please adapt changes by the function NEnumPrinters.

Notice

It is impossible to change the driver name, because of it is not managed by this SDK.

| | | | |
|---|--------|----------------------------------|--|
| Function name | | NGetPrinterInf / NGetPrinterInfA | |
| Argument name | IN/OUT | Type | Description |
| i_prt | I | PWCHAR / PCHAR | Printer name |
| o_ports | O | PWCHAR / PCHAR | Port information (csv_To enumerate it in a Comma Separated Value form) |
| o_size | O | PDWORD | o_ports byte size |
| Return value | INT | | |
| ·Error (negative value), Normal end (0) * Refer to error code table | | | |
| Processing description | | | |
| <p>It searches printer information management file from the printer name that is specified by argument and store the following information into o_ports,</p> <ul style="list-style-type: none">·Connection classification<ul style="list-style-type: none">USB : 1Bluetooth : 2RS232C : 3TCP/UDP : 4·Connection information<ul style="list-style-type: none">USB : Device instanceBluetooth : Device instance IDRS232C : Serial portTCP/UDP : Model name, IP address, Port No. , MAC Address <p>E.g.) 1. USB¥VID_1051&PID_1000¥7&2b99b1d4&0&2</p> <p>2. COM3: BTHENUM¥{00001101-0000-1000-8000-00805F9B34FB}_VID&00020430_PID&0211¥8&2D7F776&0&000B5DB4BB46_C00000000</p> <p>3. COM1</p> <p>4. ETHE: LAN¥NEX-M330,192.168.92.42,9100,0023A7C7E9F4</p> <p>Printer information that is obtained by o_ports is also obtained to refer the printer information management file (npidport.inf).</p> | | | |

| | | | |
|--|-----------|------|-------------|
| Function name | NAutoOpen | | |
| Argument name | IN/OUT | Type | Description |
| i_flg | I | INT | - |
| Return value | BOOL | | |
| ·FALSE at all time : Auto Open Invalid | | | |
| Processing description | | | |
| | | | |

- This function is obsolete from Ver.3.0.0.0
It is still remaining to keep the compatibility to previous version.
And return value is FALSE at all time (Auto open invalid.)

Execute NOpenPrinter to do the open process,

| | | | |
|---|--------|------------------------------|---|
| Function name | | NOpenPrinter / NOpenPrinterA | |
| Argument name | IN/OUT | Type | Description |
| i_prt | I | PWCHAR / PCHAR | Printer name to open |
| i_statusFlg | I | BOOL | 0: Not sending 1: Turn off at the error sending. 2: Reconnect automatically at the error sending. |
| i_callback | I | SDK.NCallback | A class pointer which implemented callback interface (Optional) |
| Return value | | INT | |
| ·Error (negative value), Normal end (0) * Refer to error code table | | | |
| Processing description | | | |

·Execute to open the printer

Argument i_prt designate the printer name that is obtained by NEnumPrinters.

Argument i_statusFlg is setting the movement of the transmission and reception thread.

0: Do not activate the thread is unsupported.

1: Activate the thread, and at the transmission error, cut it and stop communication.

2: Activate the thread, and at the transmission error, it connect the line automatically and transmit the data again.

Notice

·The document awaiting the transmission is deleted.

·This function executes to open by asynchronous processing.

To pass the pointer of application screen class to “i callback” of the 3rd argument, you can obtain the result, at process completion.

(When you do not use call back, the 3rd argument is optional.)

·After open the printer , execute a data transmission after having confirmed that you can acquire status. You may lose the data, when you execute a transmission before acquiring the status.

| | | | |
|---------------|--------------------------------|----------------|---------------------|
| 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) * Refer to error code table

Processing description

- Execute to close the printer by this function

Notice

- Call this function after to confirm the end of printing.
You can get the status as end of printing, to use [Print Start/End] command and see the status.
For the detail, please refer to the product specifications for each of printers.
- The document awaiting the transmission is deleted.

| | | | |
|------------------------|----------------|------|-------------|
| Function name | NClosePrinters | | |
| Argument name | IN/OUT | Type | Description |
| | | | |
| Return value | INT | | |
| ·Normal end (0) | | | |
| Processing description | | | |
| | | | |

·To close all the printer at open status.

Notice

This function only return Success as return value.
Even it has been already closed, it return 0 (Success).
If you want to get error value, Please use NClosePrinter function.
(Close one printer, only)

| | | | |
|--|--------|------------------|--|
| Function name | | NPrint / NPrintA | |
| Argument name | IN/OUT | Type | Description |
| i_prt | I | PWCHAR / PCHAR | Target printer name |
| i_dat | I | PCHAR | Transmit data |
| i_size | I | DWORD | (hexadecimal) |
| io_jobid | IO | PDWORD | Number of output bytes Print job ID (NULL can be specified) |
| Return value | INT | | |
| ·Error (negative value), Normal end (0) * Refer to error code table | | | |
| Processing description | | | |
| <div>·Sends specified hexadecimal character data to printer.</div> <div>When the following 3 patterns are detected at analyzing the data, it handles them as special data.</div> <div>1. Character string which is surrounded by double quotation (“”). ⇒ Convert as a character string. ("ABC" ⇒ 0x41,0x42,0x43)</div> <div>2. File name which is surrounded by <> (inequality sign, Pass is included.) ⇒ Output the file content (binary data)</div> <div>3. ‘(Single quotation) is character string in the top. ⇒ It is handled as comment. (It is not output.)</div> | | | |
| <div>Notice</div> <div>·Sequential print and Batch print</div> <div>The data is sent to printer sequentially when Nprint,NlimagePrint and NimagePrintnF is executed. NStartDoc and NEndDoc enable batch data sending to printer. Please use sequential print and batch print depending on the purpose.</div> | | | |

| | | | |
|--|--------|--------------------|--------------------------------------|
| Function name | | NDPrint / NDPrintA | |
| Argument name | IN/OUT | Type | Description |
| i_prt | I | PWCHAR / PCHAR | Target printer name |
| i_dat | I | PCHAR | Transmit data (hexadecimal) |
| i_size | I | DWORD | Number of output bytes |
| io_jobid | IO | PDWORD | Print job ID (NULL can be specified) |
| Return value | | INT | |
| ·Error (negative value), Normal end (0) * Refer to error code table | | | |
| Processing description | | | |
| ·Sends specific data to printer (No conversion) | | | |
| <div><div>Notice</div></div> | | | |
| ·Sequential print and Batch print | | | |
| The data is sent to printer sequentially when NPrint , Nimage Print and Nimage printf is executed. NStartDoc and NEndDoc enable batch data sending to printer. | | | |
| Please use sequential print and batch print depending on the purpose. | | | |

| | | | |
|--|--------|----------------------------|---|
| 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 | Sending type 0x00: Raster type by line. 0x01: Raster type by block 0x02: Raster type by block with gradation 0x10: Bit image type |
| io_jobid | IO | PDWORD | Print job ID (NULL can be specified) |
| Return value | | INT | |
| ·Error (negative value), Normal end (0) * Refer to error code table | | | |
| Processing description | | | |
| ·Sends Specified device context to printer. | | | |
| <div>Notice</div> | | | |
| ·Sequential print and Batch print The data is sent to printer sequentially when NPrint or NImagePrintF is executed. NStartDoc and NEndDoc enable batch data sending to printer. Please use sequential print and batch print depending on the purpose. | | | |

| | | | |
|---|--------|------------------------------|--|
| Function name | | NImagePrintF / NImagePrintFA | |
| Argument name | IN/OUT | Type | Description |
| i_prt | I | PWCHAR / | Target printer name |
| i_bmp | I | PCHAR | BMP file name |
| i_putType | I | PCHAR | Sending type |
| | | BYTE | 0x00: Raster type by line. 0x01: Raster type by block 0x02: Raster type by block with gradation 0x10: Bit image type by line. |
| io_jobid | IO | PDWORD | Print job ID (NULL can be specified) |
| Return value | | INT | |
| ·Error (negative value), Normal end (0) * Refer to error code table | | | |
| Processing description | | | |
| ·Sends Specified BMP file to printer. | | | |
| <div>Notice</div> | | | |
| ·Sequential print and Batch print | | | |
| The data is sent to printer sequentially when NPrint ,NImagePrint and NImagePrintF is executed. NStartDoc and NEndDoc enable batch data sending to printer. | | | |
| Please use sequential print and batch print depending on the purpose. | | | |

| | | | |
|--|--------------------------|---------------------------|-------------------------------|
| Function name | NGetStatus / NGetStatusA | | |
| Argument name | IN/OUT | Type | Description |
| i_prt o_status | I O | PWCHAR / PCHAR LPDWORD | Target printer name Status |
| Return value | INT | | |
| ·Error (negative value), Normal end (0) , Warning (positive value) * Refer to error code table | | | |
| Processing description | | | |
| ·Returns a status obtained from a specified printer. * See command 《ESC v》 in target printer's specification sheet for return values. | | | |

| | | | |
|--|--------|------------------------------------|--|
| 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 storage area |
| o_time | O | PDWORD | Update flag (elapsed time since system boots up) (NULL can be specified) |
| Return value | INT | | |
| ·Error (negative value), Normal end (0) * Refer to error code table | | | |
| Processing description | | | |
| ·Obtain information reserved in target type ID of Extended Information | | | |
| * Host application should send request for information to printer previously. (Not all require the information as extended status, transfer complete, print complete, etc.) | | | |
| * See command 《ESC v》 in target printer's specification sheet for return values. | | | |

| | | | |
|---------------|--------------------------------|----------------|---------------------|
| 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) * Refer to error code table

Processing Description

- Reset printer. Print job during printing is cancelled.
- Confirm that the return value of this API and printer is reset and online in order to confirm that the API is executed normally. (Confirm by NGetStatus)

Notice

This function is available only for USB and TCP/UDP connection.

| | | | |
|---|------------------------|----------------|---------------------|
| Function name | NStartDoc / NStartDocA | | |
| Argument name | IN/OUT | Type | Description |
| i_prt | I | PWCHAR / PCHAR | Target printer name |
| o_jobid | O | PDWORD | Print job ID |
| Return value | INT | | |
| ·Error (negative value), Normal end (0) * Refer to error code table | | | |
| Processing description | | | |
| <div>·Starts document</div> <div>After having issued NStartDoc, about, NPrint,NDPrint,NImagePrint,NImagePrintF they accumulate the data into temporally file.</div> <div>To call NEndDoc, it can output the accumulated data to the printer.</div> <div>Or else, it can clear the accumulated data to call NCancelDoc.</div> <div>1 Document for 1 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) * Refer to error code table | | | |
| Processing description | | | |
| | | | |

- Ends document
To output the accumulated data to the printer, after calling NStartDoc.
When the data does not exist, it does not execute.

| | | | |
|---|--------------------------|----------------|---------------------|
| 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) * Refer to error code table | | | |
| Processing description | | | |
| | | | |

- Cancels document

When the data does not exist, it does not execute.

| | | | |
|--|----------------------|--|----------------------------|
| Function name | NEnumDoc / NEnumDocA | | |
| Argument name | IN/OUT | Type | Description |
| i_prt | I | PWCHAR / PCHAR | Target printer name |
| o_docnolist | O | PWCHAR / PCHAR | Document number list (CSV) |
| Return value | INT | | |
| ·Error / No transmission waiting (0) , Successful termination (List saving size) | | | |
| Processing description | | <div>▪To return the document number list(CSV) as the status of transmission waiting. * Document Status, document number (18Digits) line feed is one document information. * Document Status 0: Transmission waiting, 1: Under the transmission, 2: Transmission error. * To designate o_docnolist as NULL, it obtains only the size that needs to save the list.</div> | |
| | | | |

| | | | |
|--|--------------------------|--|---------------------|
| Function name | NDeleteDoc / NDeleteDocA | | |
| Argument name | IN/OUT | Type | Description |
| i_prt | I | PWCHAR / PCHAR | Target printer name |
| i_docno | I | PWCHAR | Document number |
| Return value | INT | | |
| ·Error (negative value), Normal end (0) * Refer to error code table | | | |
| Processing description | | <div>·Delete the document awaiting the transmission. If there is no document awaiting the transmission, it dose not execute.</div> | |
| | | | |

| | | | |
|---|----------------------|----------------|-----------------------|
| 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 |
| i_bmp | IO | 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 | | | |

- To draw the Barcode of setting specified in Barcode font of the printer, on i_bmp. iOS SDK dose not support 2D Barcode. It is applicable only 1D Barcode.

For the Font name of 2nd argument, Please designate it from Barcode setting file (NBarcodeInf : Barcode1~10).

About NBarcodeInf, Please refer [Barcode setting] on this document.

Notice

- Even if, created Barcode exceeds data size, it does not become the error either, and Barcode data where an exceeded part was deleted is created.

In this case because it cannot read Barcode created, please coordinate data size, and, please designate it so that all data finishes entering.

| | |
|------------------------|--|
| Function name | NBarcode |
| Processing description | The Figure of designated image of the argument |

UIImage Class Area (3rd Argument • Grey frame below)



- * Please specify Width, Height, HRI character, etc. on the Barcode area of the **Red frame** above by Barcode setting file(NBarcodeInf).
- * Red frame above is rotated at the rotation setting of Barcode.
After the rotation, Please specify the white area above by width/ height, because of the white area is not rotated.

| | | | |
|--|----------------------------|---------------|---|
| 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 Check sum |
| io_jobid | IO | PDWORD | Print job ID (NULL can be specified) |
| Return value | INT | | |
| ·Error (negative value), Normal end or coincidence of check sum (0/1) * Refer to error code table | | | |
| Processing description | | | |
| <div>· Transmit the specified FWF file to the printer. Obtain the check sum form the printer and compare it with specified check sum.</div> <div>* Be sure to execute function NGetStatus before executing this function to check whether the printer works properly .</div> | | | |

| | | | |
|--|--------------------|------|--|
| Function name | NInitializeNetwork | | |
| Argument name | IN/OUT | Type | Description |
| i_flg | I | INT | UDP reception thread Activate/Stop flag. 0: Stop 1: Activate |
| Return value | INT | | |
| ·Error (negative value), Normal end (0) * Refer to error code table | | | |
| Processing Description | | | |
| | | | |

- Activate/Stop UDP reception thread.
At the initial state, it stops.
- At TCP/UDP connection, please activate UDP reception thread by this function, before Open process (NOpenPrinter).

In case of not using TCP/UDP connection, please do not execute this function.

| | | | |
|---------------|---------------|-------|---|
| Function name | NScanPrinters | | |
| Argument name | IN/OUT | Type | Description |
| i_waitmsec | I | DWORD | A waiting time from enumeration response to data generation. (m sec.) |
| Return value | INT | | |

·Error (negative value), Normal end (0) * Refer to the NOTE below. * Refer to error code table.

Processing Description

Asynchronous function

- To execute this function, it broadcasts printer enumeration request of maintenance protocol.

It receives printer enumeration response while the time (m sec.) which is appointed by the argument i waitmmsec, and generate a printer information from response data.

Generally appoint it as 3000 (3 seconds) by i_waitmsec.

(To get printer information, please use another function NEnumPrinters.)

Notice

If it appointed "0" for i_waitmsec, then it does not execute printer enumeration request but just the number of the created printer is returned as return value.

This function detects the printer as asynchronous processing.

To finish the detection, pass a Ncallback type function pointer to NSetCallback and obtain finishing the process and its output.

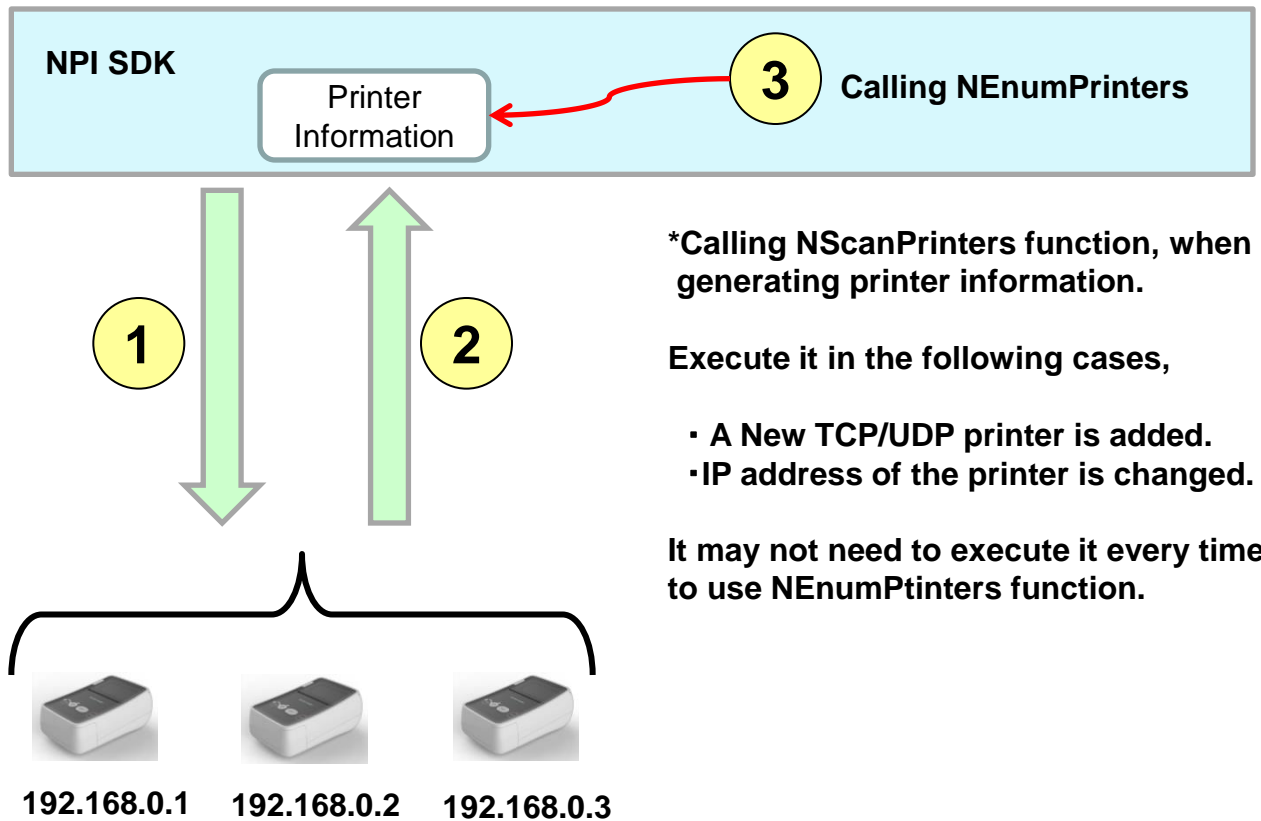
(Error code, the numbers of detected printers.)

Win32 API Reference

Obtaining Printer name at when TCP/UDP (WLAN) using.



PC, Tablet etc.



1 Calling NScanPrinters
(Printer enumeration request : UDP-broadcasting)

2 It receives the printer information in SDK as the printer enumeration response : UDP, from each printers that connected by TCP/UDP.
(Receive the processing completion of NScanPrinters, as a callback function.)

3 Calling NEnumPrinters
(To obtain TCP/UDP printer information by received data.)

| | | | |
|---|----------------------------|------------------------------|---------------------|
| Function name | NTCPortLock / NTCPortLockA | | |
| Argument name | IN/OUT | Type | Description |
| i_prt | I | PWCHAR / PCHAR | Output printer name |
| i_type | I | BYTE | Setting type |
| Return value | INT | | |
| ·Error (negative value), Normal end (0) * Refer to error code table | | | |
| Processing Description | | Asynchronous function | |
| <div>▪ Execute Lock/Unlock of TCP communication. Only the locked user can transmit the data via TCP communication. The other users also can connect but can not transmit.</div> <div>Setting type Lock : 1 Unlock : 0</div> | | | |

| | | | |
|--|------------------------------|------------------------------|---------------------|
| Function name | NBufferClear / NBufferClearA | | |
| Argument name | IN/OUT | Type | Description |
| i_prt | I | PWCHAR / PCHAR | Output printer name |
| Return value | INT | | |
| ·Error (negative value), Normal end (0) * Refer to error code table | | | |
| Processing Description | | Asynchronous function | |
| <div>▪ Clear reception buffer (All buffer) Return error except TCP/UDP connection.</div> | | | |

| | | | |
|--|--|--------------|---------------------|
| Function name | NBlockSendSetting / NBlockSendSettingA | | |
| Argument name | IN/OUT | Type | Description |
| i_prt | I | PWCHAR／PCHAR | Output printer name |
| i_type | I | BYTE | Setting type |
| Return value | INT | | |
| ·Error (negative value), Normal end (0) * Refer to error code table | | | |
| Processing Description | Asynchronous function · It is possible to transmit the data which stored in the buffer of TCP/UDP module in a batch. This function can switch ON/OFF of the data transmission in a batch. (Error is returned except TCP/UDP connection.) Setting type ON : 1 OFF : 0 | | |

Error code table (1/2)

This is the error code table which are using by the SDK. Mainly it is used as return value of the functions. Otherwise, some of them are used in only SDK inside processing. Or else, they exist for the compatibility with other OS.

For return value of each function, please refer to [Function return value table].

| | | |
|------------------------|-----|--|
| N_SUCCESS | 0 | Successful termination |
| N_ERR_HANDLE | -1 | Handle error (Device context handle, etc.) |
| N_ERR_PRTOPEN | -2 | Printer open error |
| N_ERR_SEND_ERROR | -3 | Transmission error |
| N_ERR_OFFLINE | -5 | Off line |
| N_ERR_PRTCLOSE | -6 | Printer close error |
| N_ERR_FILEOPEN | -10 | File open error |
| N_ERR_NOT_MAPPING | -11 | Expanded information obtaining error (Map View of File failure) |
| N_ERR_NOT_OPEN_MAPFILE | -12 | Unable to open the file map |
| N_ERR_PRTOUTPUT | -13 | Printer output error |
| N_ERR_NONE_PRTLIST | -21 | There is no printer available. |
| N_ERR_NOHANDLE | -22 | Printer is not opened |
| N_ERR_LACKRESOURCE | -31 | Resource shortage |
| N_ERR_NOTSUPPORTED | -40 | Function is not supported |
| N_ERR_LOADFROMFILE | -50 | Fail to read an image file |
| N_ERR_IMAGESIZE | -51 | Image size is incorrect. |
| N_ERR_RESETPRINTER | -60 | Fali to reset |
| N_ERR_ACCESSDENIED | -61 | Access denied |
| N_ERR_DOCNOTSTARTED | -71 | Document is not in opening status. |
| N_ERR_ALREADYSTARTDOC | -72 | Document has already been opened |
| N_ERR_FWFFILE | -80 | fwf file error |
| N_ERR_FWF_CHECKSUM | -81 | Check sum of fwf file and Check sum obtained from printer dose not accord. |
| N_ERR_FWDL_TIMEOUT | -82 | Firmware download time out (Print start command check is time out.) |
| N_ERR_FWCHK_TIMEOUT | -83 | Firmware check sum verification time out. |
| N_ERR_FOUNDERERROR | -84 | Detect an error at status verification on firmware downloading. |
| N_ERR_ARGUMENT | -90 | Argument is not correct |
| N_ERR_ARGUMENT_01 | -91 | 1st argument is incorrect. |
| N_ERR_ARGUMENT_02 | -92 | 2nd argument is incorrect. |
| N_ERR_ARGUMENT_03 | -93 | 3rd argument is incorrect. |
| N_ERR_ARGUMENT_04 | -94 | 4th argument is incorrect. |
| N_ERR_ARGUMENT_05 | -95 | 5th argument is incorrect. |

Error code table (2/2)

| | | |
|-------------------------|------|--|
| N_ERR_ARGUMENT_06 | -96 | 6th argument is incorrect. |
| N_ERR_ARGUMENT_07 | -97 | 7th argument is incorrect. |
| N_ERR_ARGUMENT_08 | -98 | 8th argument is incorrect. |
| N_ERR_ARGUMENT_09 | -99 | 9th argument is incorrect. |
| N_ERR_SOCKRECV | -108 | TCP/UDP reception error |
| N_ERR_UDPTHREADSTARTED | -111 | UDP thread has already been started |
| N_ERR_UDPTHREADSTOPPED | -112 | UDP thread is not started |
| N_ERR_UDPTHREADSTOP | -113 | UDP thread is not stopped |
| N_ERR_PRTINFO_READ | -131 | Fail to read a printer information file |
| N_ERR_PRTINFO_WRITE | -132 | Fail to write a printer information file |
| N_ERR_PRTNAME_ALLOC | -133 | Fail to allocate a printer name |
| N_ERR_PRTRENAME_BRFORE | -134 | A printer name before the change does not exist |
| N_ERR_PRTRENAME_AFTER | -135 | A printer name after the change has already been used. |
| N_ERR_PRTINFO_DELETE | -138 | Fail to delete the printer name |
| N_ERR_PRTINFO_NOTFOUND | -139 | The printer name does not exist |
| N_ERR_DEVICE_NOTSUPPORT | -150 | Connection type is not supported |
| N_ERR_CREATEBCDDATA | -162 | Fail to generate a barcode data |
| N_ERR_MNT_HEADER | -200 | Maintenance response header is incorrect. |
| N_ERR_FLAGSQR | -201 | QR flag of maintenance response(Flags) is incorrect |
| N_ERR_FLAGSFORMAT | -202 | Detect a format error at maintenance response(Flags) |
| N_ERR_FLAGSBUSY | -203 | Detect busy at maintenance response(Flags) |
| N_ERR_FLAGSUNDEFINED | -204 | Detect undefined at maintenance response(Flags) |
| N_ERR_FLAGSREJECT | -205 | Detect rejection at maintenance response(Flags) |
| N_ERR_FLAGSOTHER | -206 | Detect other error at maintenance response(Flags) |
| N_ERR_MNT_QID | -207 | Query ID of maintenance response is incorrect |
| N_ERR_MNT_QR | -208 | Preliminary flag of maintenance response is incorrect |
| N_ERR_MNT_QPARAM | -209 | Query parameter of maintenance response is incorrect |
| N_ERR_MNT_OTHER | -210 | There is another error at maintenance response |
| N_ERR_OTHER | -999 | Other error |
| N_WRN_PRTALREADYOPEN | 10 | Printer has already been opened. |
| N_WRN_NOTEXITDOC | 11 | Document does not exist. |

Win32 API Reference

API-Specific error code table(1/4)

| Function name | Defined name | Defined value | Note |
|-------------------|-----------------------------------|---------------|-----------------------------------|
| NSetCallback | There is no return value | — | |
| NEnumPrinters | N_SUCCESS | 0 | Function return value |
| | N_ERR_NONE_PRTLIST | -21 | 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 |
| 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_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 | It is deprecated from ver.3.0.0.0 | FALSE | DO NOT use this function |
| NOpenPrinter | N_SUCCESS | 0 | Function return value · Call back |
| | N_WRN_PRTALREADYOPEN | 10 | Function return value · Call back |
| | N_ERR_PRTOPEN | -2 | Function return value · Call back |
| | N_ERR_ARGUMENT_01 | -91 | Function return value |
| | N_ERR_UDPTHREADSTOPPE | -112 | Function return value · Call back |
| | N_ERR_PRTINFO_NOTFOUND | -139 | Function return value · Call back |
| | N_ERR_DEVICE_NOTSUPPORT | -150 | Function return value · Call back |
| NClosePrinter | N_SUCCESS | 0 | Function return value |
| | N_ERR_NONE_PRTLIST | -21 | 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 |
| NPrint | N_SUCCESS | 0 | Function return value |
| | N_ERR_FILEOPEN | -10 | Function return value |
| | N_ERR_NONE_PRTLIST | -21 | Function return value |
| | N_ERR_LOADFROMFILE | -50 | 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 52 | -93 | Function return value |

API-Specific error code table(2/4)

| Function name | Defined name | Defined value | Note |
|-----------------|-------------------------|---------------|-----------------------------------|
| NDPrint | N_SUCCESS | 0 | Function return value |
| | N_ERR_NONE_PRTLIST | -21 | 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_NONE_PRTLIST | -21 | 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 |
| NImagePrintF | N_SUCCESS | 0 | Function return value |
| | N_ERR_PRTOUTPUT | -13 | Function return value |
| | N_ERR_NONE_PRTLIST | -21 | Function return value |
| | N_ERR_LACKRESOURCE | -31 | Function return value |
| | N_ERR_LOADFROMFILE | -50 | 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 |
| NGetStatus | N_SUCCESS | 0 | Function return value |
| | N_ERR_SEND_ERROR | -3 | Function return value |
| | N_ERR_OFFLINE | -5 | Function return value |
| | N_ERR_NONE_PRTLIST | -21 | 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_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 |
| NResetPrinter | N_SUCCESS | 0 | Function return value • Call back |
| | N_ERR_NONE_PRTLIST | -21 | Function return value |
| | N_ERR_LACKRESOURCE | -31 | Function return value • Call back |
| | N_ERR_RESETPRINTER | -60 | Function return value • Call back |
| | N_ERR_ARGUMENT_01 | -91 | Function return value |
| | N_ERR_DEVICE_NOTSUPPORT | -150 | Function return value • Call back |
| NStartDoc | N_SUCCESS | 0 | Function return value |
| | N_ERR_PRTOUTPUT | -13 | Function return value |
| | N_ERR_NONE_PRTLIST | -21 | Function return value |
| | N_ERR_ALREADYSTARTDOC | -72 | Function return value |
| | N_ERR_ARGUMENT_01 | -91 | Function return value • Call back |

API-Specific error code table(3/4)

| Function name | Defined name | Defined value | Note |
|---------------|-----------------------|---------------|-----------------------------------|
| NEndDoc | N_SUCCESS | 0 | Function return value |
| | N_ERR_NONE_PRTLIST | -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_PRTLIST | -21 | Function return value |
| | N_ERR_DOCNOTSTARTED | -71 | Function return value |
| | N_ERR_ARGUMENT_01 | -91 | Function return value |
| NEnumDoc | N_SUCCESS | 0 | Function return value |
| | N_ERR_NONE_PRTLIST | -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_PRTLIST | -21 | Function return value |
| | N_ERR_ARGUMENT_01 | -91 | Function return value |
| Nbarcode | N_SUCCESS | 0 | Function return value |
| | N_ERR_NONE_PRTLIST | -21 | 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 |
| | N_ERR_CREATEBCDDATA | -162 | Function return value |
| NFirmwareDL | N_SUCCESS | 0 | Function return value · Call back |
| | N_ERR_NONE_PRTLIST | -21 | Function return value |
| | N_ERR_ARGUMENT_01 | -91 | Function return value |
| | N_ERR_ARGUMENT_03 | -93 | Function return value · Call back |
| | N_ERR_PRTALREADYOPEN | 10 | Function return value · Call back |
| | N_ERR_FWF_CHECKSUM | -81 | Function return value · Call back |
| | N_ERR_FWCHK_TIMEOUT | -83 | Function return value · Call back |
| | N_ERR_OFFLINE | -5 | Function return value · Call back |
| | N_ERR_ALREADYSTARTDOC | -72 | Function return value · Call back |
| | N_ERR_FWFFILE | -80 | Function return value · Call back |
| | N_ERR_FWDL_TIMEOUT | -82 | Function return value · Call back |
| | N_ERR_FOUNDERERROR | -84 | Function return value · Call back |

API-Specific error code table(4/4)

| Function name | Defined name | Defined value | Note |
|--------------------|-------------------------|---------------|-----------------------------------|
| NSetUSBProtocol | N_SUCCESS | 0 | Function return value |
| NInitializeNetwork | N_SUCCESS | 0 | Function return value |
| | N_ERR_ARGUMENT_01 | -91 | Function return value |
| | N_ERR_UDPTHREADBUSY | -115 | Function return value |
| NScanPrinters | N_SUCCESS | 0 | Function return value · Call back |
| | N_ERR_ARGUMENT_01 | -91 | Function return value · Call back |
| | N_ERR_UDPTHREADSTOPPED | -112 | Function return value · Call back |
| | N_ERR_PRTNAME_ALLOC | -133 | Function return value · Call back |
| | N_ERR_PRTINFO_NOTFOUND | -139 | Function return value · Call back |
| | LAN Printer number | + value | Function return value · Call back |
| NTCPPortLock | N_SUCCESS | 0 | Function return value · Call back |
| | N_ERR_NONE_PRTLIST | -21 | Function return value |
| | N_ERR_ARGUMENT_01 | -91 | Function return value |
| | N_ERR_ARGUMENT_02 | -92 | Function return value · Call back |
| | N_ERR_UDPTHREADSTOPPED | -112 | Function return value · Call back |
| | N_ERR_DEVICE_NOTSUPPORT | -150 | Function return value · Call back |
| NBufferClear | N_SUCCESS | 0 | Function return value · Call back |
| | N_ERR_NONE_PRTLIST | -21 | Function return value |
| | N_ERR_ARGUMENT_01 | -91 | Function return value |
| | N_ERR_UDPTHREADSTOPPED | -112 | Function return value · Call back |
| | N_ERR_DEVICE_NOTSUPPORT | -150 | Function return value · Call back |
| NBlockSendSetting | N_SUCCESS | 0 | Function return value · Call back |
| | N_ERR_NONE_PRTLIST | -21 | Function return value |
| | N_ERR_ARGUMENT_01 | -91 | Function return value |
| | N_ERR_ARGUMENT_02 | -92 | Function return value · Call back |
| | N_ERR_UDPTHREADSTOPPED | -112 | Function return value · Call back |
| | N_ERR_DEVICE_NOTSUPPORT | -150 | Function return value · Call back |

Extended Information

Type1 : 4 bytes (fixed) : update flag (4bytes) <Extended status> 1 byte: 7~0, 2 bytes: 15~8,
3 bytes: 23~16, 4 bytes: 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 head current dot line>
Type 7 : 4 bytes (fixed) : update flag (4 bytes) <Number of drive dot line >
Type 8 : 4 bytes (fixed) : update flag (4 bytes) <Number of cut>
Type 9 : 16 bytes (fixed) : update flag (4 bytes) <User maintenance counter:
Number of head current dot line,
Number of drive dot line,
Number of cut,
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 notice: arbitrary ID and end status will be described
at proceeding end command by assigning print start/end command.

Type20 : 8 bytes (fixed) : update flag (4 bytes) <Reserved>
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) <Notice of transfer completion: 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 check sum>
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: 1st Byte, CTS 2nd Byte, DSR
* Final signal status acquisition time stamp is set to update flag>

* There is no validity against acquired content with respect to information that is not functionally installed in printer except Type25 and Type31.

* Not all descriptions can be used with the printer.

Type32: 8 Byte hexadecimal string/Character string
Type33: 4 Byte hexadecimal string/Character string
Type34: 2 Byte hexadecimal string/Character string
Type35: 8 Byte hexadecimal string
Type36: 8 Byte hexadecimal string
Type37: 4 Byte hexadecimal string
Type38: 4 Byte hexadecimal string
Type39: 2 Byte hexadecimal string
Type40: 2 Byte hexadecimal string

* Replied information of Type11/32/33/34 are changed by the sub ID.

List of network-related expansion information classification

| | |
|--|--|
| MAC address(6 Byte) | Type32-0 response: hexadecimal string |
| IP address(4 Byte) | Type33-0 response: hexadecimal string |
| Subnet mask (4 Byte) | Type33-1 response: hexadecimal string |
| Default gateway (4 Byte) | Type33-2 response: hexadecimal string |
| DNS saver address(4 Byte × 5) | Type33-3 response: hexadecimal string |
| Print time out (Unit millisecond: 4 Byte) | Type33-14 response: hexadecimal string |
| Communication mode (infrastructure, ad hoc) | Type34-4 response: hexadecimal string |
| Band(2.4GHz/5GHz) | Type34-5 response: hexadecimal string |
| Transmitter power level(Low/Medium/High) | Type34-6 response: hexadecimal string |
| Channel | Type34-7 response: hexadecimal string |
| Security type(OPEN/WPA/WPA2/WEAP) | Type34-8 response: hexadecimal string |
| Encryption type(OPEN/TKIP/AES) | Type34-9 response: hexadecimal string |
| IP setting method (Automatic(DHCP/APIPA) / Manual) | Type34-10 response: hexadecimal string |

Class description

Name space: NPrinterCLib

Class name : NClassLib

| Type | Name | Remarks |
|--------|-----------------|-------------------|
| Method | NEnumPrinters | Win32 API wrapper |
| Method | NRenamePrinter | Win32 API wrapper |
| Method | NGetPrinterInf | Win32 API wrapper |
| Method | NAutoOpen | Win32 API wrapper |
| 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 | NBarcode | Win32 API wrapper |
| Method | NFirmwareDL | Win32 API wrapper |

| | | | |
|------------------------|---------------|---|---------------------------------------|
| Function name | NEnumPrinters | | |
| Argument name | IN/OUT | Type | Description |
| o_printers o_size | I O | [MarshalAs(UnmanagedType.LPWStr)] StringBuilder out long | Printer name Byte size of printers |
| Return value | int | | |
| | | | |
| Processing description | | | |
| | | | |
| Note | | | |
| | | | |

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

| | | | |
|------------------------|-------------------|---|-------------------------------------|
| Function name | NGetPrinterFromID | | |
| Argument name | IN/OUT | Type | Description |
| i_ID o_printer | I O | [MarshalAs(UnmanagedType.LPWStr)] string [MarshalAs(UnmanagedType.LPWStr)] StringBuilder | Bluetooth serial ID Printer name |
| Return value | int | | |
| | | | |
| Processing description | | | |
| | | | |
| Note | | | |
| | | | |

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

| | | | |
|------------------------|----------------|--|--------------|
| Function name | NDeletePrinter | | |
| Argument name | IN/OUT | Type | Description |
| o_printers | 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

| | | | |
|---------------------------|----------------|--|---|
| Function name | NRenamePrinter | | |
| Argument name | IN/ OUT | Type | Description |
| i_beforeprt i_afterprt | I I | [MarshalAs(UnmanagedType.LPWStr)] string [MarshalAs(UnmanagedType.LPWStr)] string | Printer name before the change Printer name after the change |
| Return value | int | | |
| | | | |
| Processing description | | | |
| | | | |
| Note | | | |
| | | | |

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

| | | | |
|----------------------------|----------------|--|--|
| Function name | NGetPrinterInf | | |
| Argument name | IN/ OUT | Type | Description |
| i_prt o_ports o_size | I O O | [MarshalAs(UnmanagedType.LPWStr)] string MarshalAs(UnmanagedType.LPWStr)] StringBuilder out long | Printer name Port information Byte size of o_ports |
| Return value | int | | |
| | | | |
| Processing description | | | |
| | | | |
| Note | | | |
| | | | |

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

| | | | |
|------------------------|--|------|---|
| Function 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 is deprecated from ver.3.0.0.0. Return value is always false. | | |
| | | | |

This function is deprecated from ver.3.0.0.0.
Return value is always false.

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

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

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

| | | | |
|------------------------|---------------|--|--------------|
| Function 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

| | | | |
|------------------------|----------------|------|-------------|
| Function 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

| | | | |
|------------------------|------------|--|---|
| Function name | NPrint | | |
| Argument name | IN/ OUT | Type | Description |
| i_prt | I | [MarshalAs(UnmanagedType.LPWStr)] string byte[] | Target printer name |
| i_dat | I | | Send data (Hexadecimal string) |
| i_size | I | uint | Output number of bytes |
| 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

| | | | |
|------------------------|---------|--|---|
| Function name | NDPrint | | |
| Argument name | IN/OUT | Type | Description |
| i_prt | I | [MarshalAs(UnmanagedType.LPWStr)] string | Target printer name |
| i_dat | I | byte[] | Transmit data (hexadecimal) |
| i_size | I | uint | Number of bytes |
| 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

| | | | |
|------------------------|-------------|----------------------|---|
| Function name | NImagePrint | | |
| Argument name | IN/OUT | Type | Description |
| i_prt | I | [MarshalAs(~LPWStr)] | Target printer name |
| i_bmp | I | string | Device context handle |
| i_width | I | IntPrt | Width |
| i_height | I | int | Height |
| i_putType | I | int | Sending type |
| | | byte | 0x00: Raster form line unit. 0x01: Raster form block unit. 0x02: Raster form block unit, gradation. 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

| | | | |
|------------------------|--------------|-----------------------------|--|
| Function 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 method 0x00: Raster form line unit. 0x01: Raster form block unit. 0x02: Raster form block unit, gradation. 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

| | | | |
|--|------------|--|--------------|
| Function name | NGetStatus | | |
| Argument name | IN/OUT | Type | Description |
| i_prt | I | [MarshalAs(UnmanagedType.LPWStr)] string | Printer name |
| o_status | O | out long | Status |
| Return value | int | | |
| | | | |
| Processing description | | | |
| | | | |
| Note | | | |
| ▪ Use it after property setting of Printer Name. | | | |

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

| | | | |
|----------------------------------|------------------|--|---|
| Function name | NGetInformation | | |
| Argument name | IN/ OUT | Type | Description |
| i_prt i_id o_dat o_time | I I O O | [MarshalAs(UnmanagedType.LPWStr)] string byte byte[] out long | Printer name Classification ID Expansion information storage area Update Flag (Elapsed time after system booting.) (NULL can be specified.) |
| Return value | int | | |
| | | | |
| Processing description | | | |
| | | | |
| Note | | | |
| | | | |

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

| | | | |
|------------------------|---------------|--|--------------|
| Function 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

| | | | |
|------------------------|-----------|--|--------------|
| Function name | NStartDoc | | |
| Argument name | IN/OUT | Type | Description |
| i_prt o_jobid | I O | [MarshalAs(UnmanagedType.LPWStr)] string out long | Printer name |
| Return value | int | | |
| | | | |
| Processing description | | | |
| | | | |
| Note | | | |
| | | | |

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

| | | | |
|------------------------|---------|--|--------------|
| Function 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

| | | | |
|------------------------|------------|--|--------------|
| Function 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

| | | | |
|------------------------|----------|---|--------------------------------------|
| Function name | NEnumDoc | | |
| Argument name | IN/OUT | Type | Description |
| i_prt o_docnolist | I O | [MarshalAs(UnmanagedType.LPWStr)] string [MarshalAs(UnmanagedType.LPWStr)] StringBuilder | Printer name Document number list |
| Return value | int | | |
| | | | |
| Processing description | | | |
| | | | |
| Note | | | |
| | | | |

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

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

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

| | | | |
|------------------------|----------|--|-----------------------|
| Function 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 | int | Left |
| i_y | I | int | TOP |
| i_width | I | int | Width |
| i_height | I | int | Height |
| i_dat | I | byte[] | Barcode data |
| i_size | I | int | Data size |
| Return value | int | | |
| | | | |
| Processing description | | | |
| | | | |
| Note | | | |
| | | | |

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

| | | | |
|------------------------|-------------|---|--|
| Function 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) |
| io_chksum | O | out short | 0x01: Valid |
| io_jobid | O | out long | Fwf File check sum 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

| | | | |
|------------------------|---------------|------|--|
| Function name | NScanPrinters | | |
| Argument name | IN/OUT | Type | Description |
| i_waitmsec | I | UInt | A waiting time from printer enumeration request to enumeration response· data generation. (m sec.) |
| Return value | int | | |
| | | | |
| Processing Description | | | |
| | | | |
| Note | | | |
| | | | |

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

| | | | |
|------------------------|--------------|--|-----------------|
| Function name | NTCPPortLock | | |
| Argument name | IN/OUT | Type | Description |
| i_prt | I | [MarshalAs(UnmanagedType.LPWStr)] 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

| | | | |
|---------------------------|--------------|---|--------------|
| Function name | NBufferClear | | |
| 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

| | | | |
|------------------------|-------------------|--|--------------|
| Function name | NBlockSendSetting | | |
| Argument name | IN/OUT | Type | Description |
| i_prt | I | [MarshalAs(UnmanagedType.LPWStr)] string | 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

| | | | |
|---------------------------|--------------|---------------------|----------------------------|
| Function name | NSetCallback | | |
| Argument name | IN/OUT | Type | Description |
| i_callback | I | NClassLib.NCALLBACK | Call back function pointer |
| Return value | int | | |
| | | | |
| Processing Description | | | |
| | | | |
| Note | | | |
| | | | |

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

| | | | |
|------------------------|--------------------|------|--------------|
| Function 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