

SDKガイド

● 概要

SDKの概要について説明します。

● Win32 APIリファレンス

APIの構文について説明します。

● .net APIリファレンス

.net環境で使用するAPIの構文について説明します。

ガイド対象バージョン	3.2.x.x
版数	第8版

ご注意

- ・本書の内容については、万全を期して作成いたしましたが、万一ご不審な点や誤り、記載もれなど、お気づきの点がございましたらご連絡ください。
- ・本書の内容については、予告なしに変更することがあります。最新の情報はお問い合わせください。
- ・本書の内容の一部または全部を無断で転載、複写、複製、改ざんすることは固くお断りします。
- ・ご利用いただいた結果の影響については、一切の責任を負いかねますのでご了承ください。
- ・本製品がお客様により不適切に使用されたり、本書の内容に従わずに取り扱われたり、または弊社以外の第三者により修理・変更されたことなどに起因して発生した損害などにつきましては、責任を負いかねますのでご了承ください。

商標について

- ・Bluetooth及びBluetoothロゴは、米国Bluetooth SIG. Inc. の登録商標です。
- ・MS-DOS®、Microsoft®、Win32®、Windows®、Windows Vista®、Visual Studio®、Visual Basic®、Visual C++®、Visual C#® は米国 Microsoft Corporation の米国およびその他の国における商標または登録商標です。
- ・その他の製品名及び会社名は、各社の商標または登録商標です。

記号について

本書では以下の記号が使われています。それぞれの記号の意味をよく理解してから製品を取り扱ってください。

 <p>注意</p>	ご使用上、必ずお守りいただきたいことを記載しています。 この表示を無視して間違った取り扱いをすると、製品の故障や動作不良の原因になる可能性があります。
 <p>参考</p>	補足説明や関連事項を記載しています。

使用制限

本製品を航空機・列車・船舶・自動車などの運行に直接関わる装置・防災防犯装置・各種安全装置など機能・精度等において高い信頼性・安全性が必要とされる用途に使用される場合は、これらのシステム全体の信頼性および安全を維持するためにフェールセーフ設計や冗長設計の措置を行うなど、システム全体の安全設計にご配慮いただいた上で当社製品をご使用いただきますようお願いいたします。

本製品は、航空宇宙機器、幹線通信機器、原子力制御機器、医療機器など、きわめて高い信頼性・安全性が必要とされている用途への使用を意図しておりませんので、これらの用途には本製品の適合性をお客様において十分ご確認のうえ、ご判断頂きますようお願いいたします。

もくじ

・ 概要	
－SDKを使用したときのシステム構成	6
・ 各種設定 / 注意事項	
－インストール	7
－ファイアウォールについて	8
－IPアドレス固定方法	10
－利用方法	11
－ドライバ接続、Bluetooth接続、シリアル接続について	11
－直接通信時の注意事項	11
－テンポラリ及び一時ファイルの保管場所	11
－ログ出力について	12
－バーコード設定ファイルについて	15
・ Win32 APIリファレンス	
－API一覧	17
メソッド	
－NCallback (インターフェース)	19
－NSetCallback	21
－NEnumPrinters	22
－NGetPrinterFromID	24
－NDeletePrinter	25
－NRenamePrinter	26
－NGetPrinterInf	27
－NAutoOpen	28
－NOpenPrinter	29
－NClosePrinter	30
－NClosePrinters	31
－NPrint	32
－NDPrint	33
－NImagePrint	34
－NImagePrintF	35
－NGetStatus	36
－NGetInformation	37
－拡張情報	38
－NResetPrinter	40
－NStartDoc	41

・ Win32 APIリファレンス

—NEndDoc	42
—NCancelDoc	43
—NEnumDoc	44
—NDeleteDoc	45
—NBarcodeSettings	46
—バーコード設定項目一覧	47
—NBarcode	52
—NBarcode2	54
—NFirmwareDL	56
—NInitializeNetwork	57
—NScanPrinters	58
—NtcpPortLock	60
—NBufferClear	61
—NBlockSendSetting	62
—エラーコード一覧	63
—API別エラーコード一覧	65

・ .net APIリファレンス

—クラス説明 (NClassLib)	70
メソッド	
—NEnumPrinters	72
—NGetPrinterFromID	72
—NDeletePrinter	73
—NRenamePrinter	73
—NGetPrinterInf	74
—NAutoOpen	74
—NOpenPrinter	75
—NClosePrinter	75
—NClosePrinters	76
—NPrint	76
—NDPrint	77
—NImagePrint	77
—NImagePrintF	78
—NGetStaus	78
—NGetInformation	79
—NResetPrinter	79

・ .net APIリファレンス

—NStartDoc	80
—NEndDoc	80
—NCancelDoc	81
—NEnumDoc	81
—NDeleteDoc	82
—NBarcodeSettings	82
—NBarcode	83
—NBarcode2	83
—NFirmwareDL	84
—NScanPrinters	84
—NtcpPortLock	85
—NBufferClear	85
—NBlockSendSetting	86
—NSetCallback	86
—NInitializeNetwork	87

概要

SDKを使用して、印刷機能を持たせたいアプリケーションに印刷及び、プリンタの監視の機能を組み込むことができます。

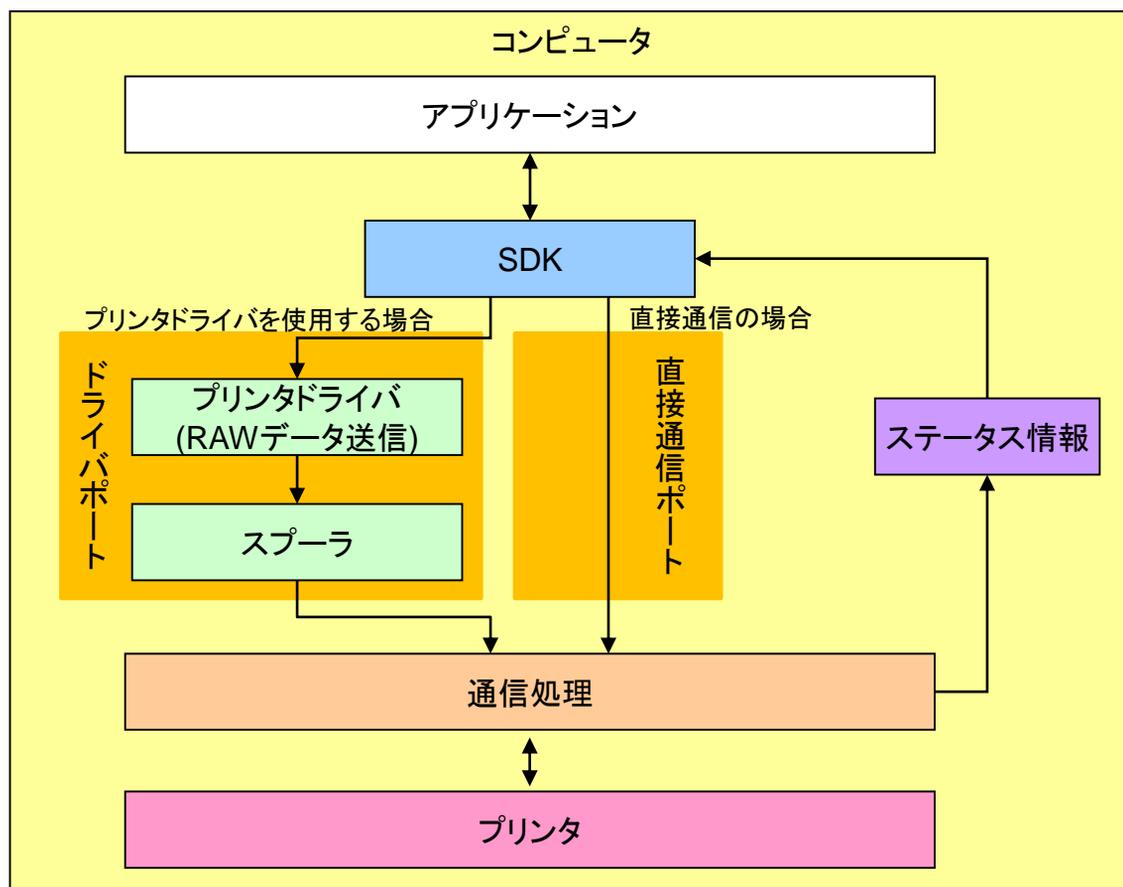
SDKとして提供されるファイルはDynamic Link Library (DLL)と各種実行ファイル(EXE)となります。
ファイル一覧

NPrinterLib.dll、BarCodeLib.dll、NBarcodeLib.dll、NPrinterCLib.dll

NServiceDrv.exe、NResetPrinter.exe

※ C#アプリケーションで使用する場合はNPrinterCLib.dllを参照設定して使用します。

SDKを使用したときのシステム構成



※直接通信ポートと、ドライバポートでは関数の呼び出し条件が変わります
※ドライバポートを使用する時のプリンタ名はプリンタドライバ名となります

開発言語

Win32 : Visual C++

.NET : Visual C#

Windows対応バージョン

Windows7 以降 32bit/64bit 対応

インターフェース

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

インストール

ライブラリのみで使用する場合

1. NPrinterLib.dll、NPrinterCLib.dllを使用されるアプリケーションから参照できるフォルダに保存してください。
NBarcodeLib.dllとNResetPrinter.exeをSystem32フォルダに保存してください。
64bit環境で32bit版SDKを使用する場合は、32bit版NBarcodeLib.dllとNResetPrinter.exeをSysWOW64フォルダに保存してください。
C#アプリケーションの場合はNPrinterCLib.dllを参照設定して使用します
2. NServiceDrv.exeをサービス登録してください。
<32bit環境>
 - ・System32フォルダにファイルを保存します。
 - ・コマンドプロンプトを管理者権限で起動して下記コマンドを実行します。

```
sc create NServiceDrv binpath= "C:¥Windows¥system32¥NServiceDrv.exe"
```  
<64bit環境>
 - ・SysWOW64フォルダにファイルを保存します。
 - ・コマンドプロンプトを管理者権限で起動して下記コマンドを実行します。

```
sc create NServiceDrv binpath= "C:¥Windows¥SysWOW64¥NServiceDrv.exe"
```
3. 登録したサービスを開始してください。
 - ・[Windows管理ツール]—[サービス]を起動します。
 - ・NServiceDrvのプロパティ[スタートアップの種類]を「自動」に変更します。
 - ・[サービスの開始]を実行します。

ドライバと併用して使用する場合

ドライバをセットアップすると自動的にインストールされるので、特別設定を必要としません
ドライバのセットアップについてはドライバの「導入ガイド」を参照してください

参考

各種環境の構築に必要なファイル等はマニュアル内の各サンプルに保存されています。

ファイアウォールについて(1/2)

NServiceDrv.exeはUDP通信を行なうプログラムのため、使用前準備としてファイアウォール許可が必要となります。

※USB、シリアル、Bluetooth使用時は不要です



「コントロールパネル」-「ファイアウォール」-「アプリまたは機能の許可」を選択します。

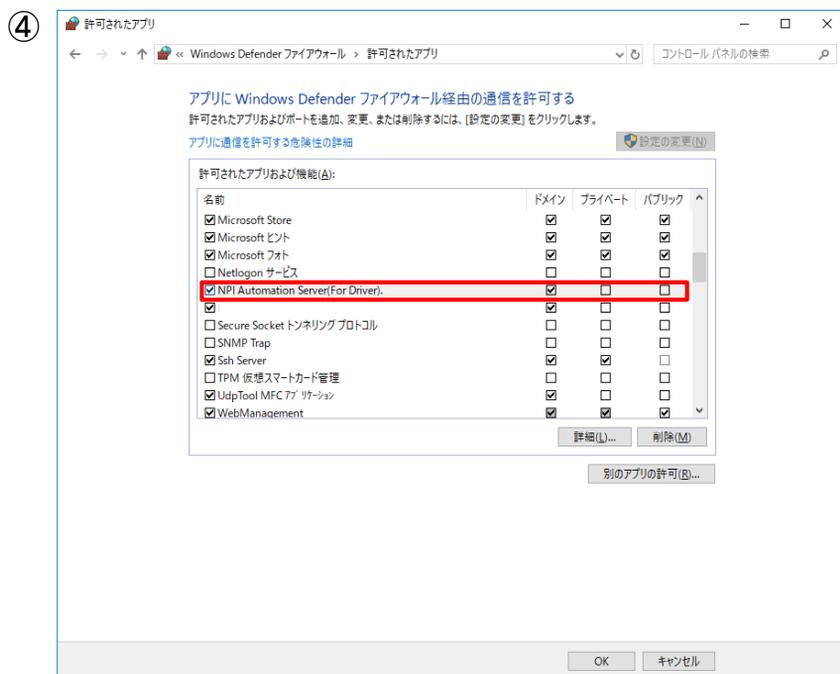


「設定の変更」-「別のアプリの許可(R)」の順でボタンを押します。

ファイアウォールについて(2/2)



「参照」でNServiceDrvのパスを指定し、「追加」ボタンでアプリ許可を実行します。



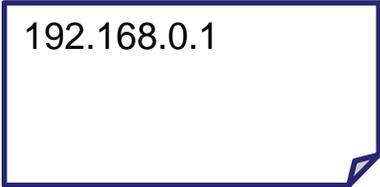
NServiceDrvが追加されたことを確認し、使用するネットワークのチェックボックスにチェックを入れてください。

IPアドレス固定方法

当SDKではブロードキャストを行うためにパソコンに設定されているIPアドレスを自動で取得しています。

しかし、ネットワークアダプタが複数存在したり、仮想環境によりネットワーク設定が複数存在する場合は、プリンタとの通信に使用しているネットワークアダプタ以外のIPアドレスを取得してしまう恐れがあります。

SDKで取得するIPアドレスを指定する場合は、NPIフォルダ直下へ IPAddress.inf ファイルを作成し、パソコンに設定されているIPアドレスを記述する必要があります。



192.168.0.1

IPAddress.inf

※IPアドレスのみを記述します。

参考

ブロードキャストはNScanPrinters関数でプリンタの検出を行う際に使用しています。

利用方法

インストール完了後、開発環境にImportApi.hをインクルードします。
それからLoadLibrary関数にてNPrinterLib.dllを読み込みます。
関数が成功するとDLLのハンドルが返りますので、そのハンドルをGetProcAddress関数に渡すと、DLL関数のアドレスを取得できます。
FreeLibrary又はアプリケーション終了によってNPrinterLib.dllをデタッチする際、NClosePrinter関数を実行してクローズ処理が正常に完了してから終了することを推奨します。
詳しくはリリース媒体のサンプルプログラム(Document/APISample)のLoadLibrary/FreeLibrary実行部分をご確認下さい。ImportApi.hはサンプルに同梱されています。

ドライバ接続について

ドライバポートを使用する際はNEnumPrinters、NOpenPrinterの呼び出しは必要ありません。

Bluetooth接続について

当SDKではBluetooth接続を行う前に、PCとプリンタのペアリングを済ませておく必要があります。

シリアル接続について

当SDKではシリアル接続を行う前に、使用するポートのポート設定を済ませておく必要があります。
プリンタとPCのポート設定が合わないと正しくデータの送受信を行えません。

直接通信時の注意事項

直接通信で使用するポートについては、ドライバをセットアップしないでください

テンポラリ及び一時ファイルの保管場所

デフォルトの状態では、OSセットアップドライブの直下にNPIフォルダが作成されます。
保管場所の変更はレジストリに下記のキーを作成し、その下にDirPathを文字列値として作成して使用されるフォルダを指定してください。
HKEY_LOCAL_MACHINE¥SOFTWARE¥NPI

例.

```
[HKEY_LOCAL_MACHINE¥SOFTWARE¥NPI]
"DirPath"="d:¥¥Printer"
```

64bit環境で32bit版SDKを使用する場合は上記レジストリにアクセスできない恐れがあるため、

```
[HKEY_LOCAL_MACHINE¥SOFTWARE¥WOW6432Node¥NPI]
フォルダにDirPathを記述してください。
```

ログ出力について(出力設定)

当SDKではログファイル指定したフォルダ配下に **SDKLog.txt** という名称で出力します。
ログ出力にはログ設定ファイル(OSドライブ¥NPI¥NLogInf.inf)を作成する必要があります。
(ファイルが存在しない場合はログを出力しません)

```
[LOGSETTING]
OUTPUT=1,2
SIZE=1000000
PATH=~~~
ID=0
SYNC=0
```

以下の形で設定ファイルを記述します。

- | | |
|-----------------|--|
| [LOGSETTING] | ← 固定 |
| OUTPUT=1, 2 | ← 出力したいログ種別をカンマ区切りで指定します
(この場合1, 2 : ERROR、WARNのみ出力) |
| SIZE=1000000 | ← ログの書き込み最大ファイルサイズを指定します |
| PATH=C:¥NPI¥log | ← ログの書き込みフォルダを指定します。 |
| ID=0 | ← プロセスID・スレッドIDの出力設定を行います。
(0 : 出力なし、1 : 出力あり) |
| SYNC=0 | ← ログの書き込みタイミングを指定します。
(0: ログを溜め込み一定間隔での非同期書き込み、1: 同期書き込み) |

ログ種別 (OUTPUT) について

- | | | |
|---------|-----------------------------|---|
| 1 ERROR | : エラー | ※ 「N****Port ~~~~lastError」のログはシステムコールのエラー値(GetLastErrorの値)を返しています。 |
| 2 WARN | : 警告 | |
| 3 FUNC | : 関数呼び出し (ステータス・拡張情報チェック以外) | |
| 4 IN | : ポート受信データ | |
| 5 OUT | : ポート送信データ | |
| 6 CHK | : ステータス・拡張情報チェック時 | |
| 7 PST | : プリントステータス値変化時、拡張情報受信時 | |

※指定した種別に沿ったログが出力されます。

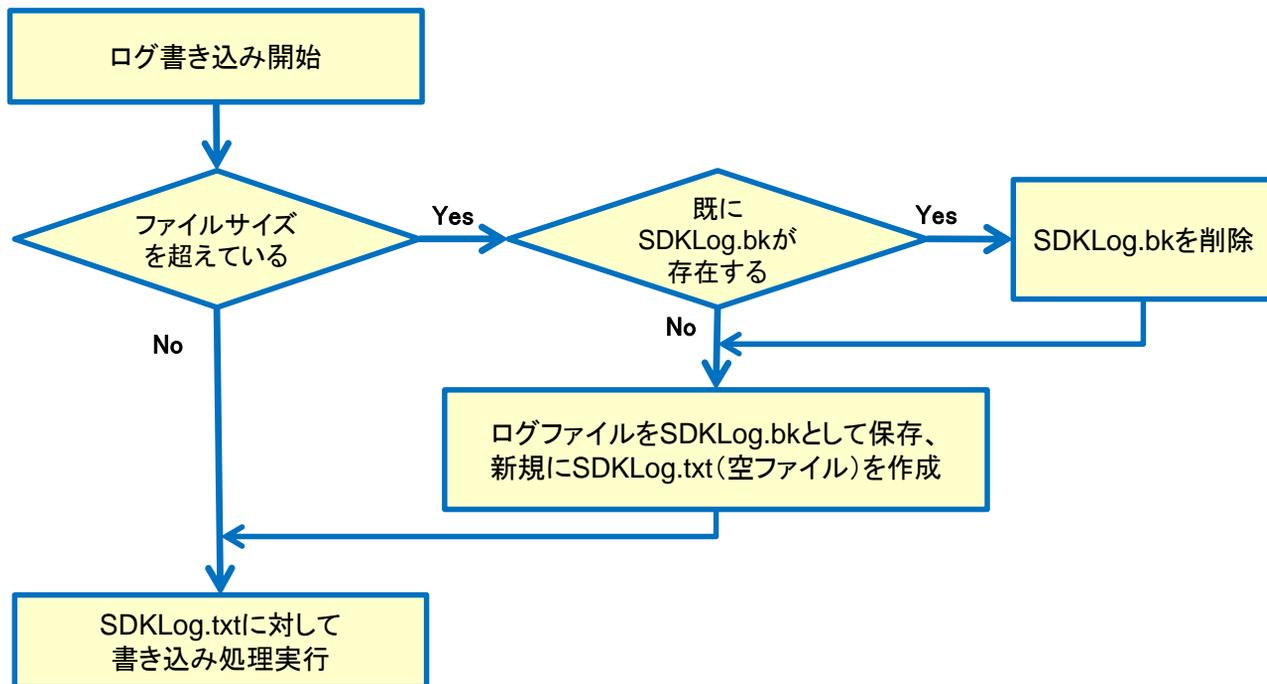
ログ出力について(注意事項)

- ※1 ファイルを編集しても即時には反映されません。
反映されるタイミングはNPrinterLibクラス生成時となります。
- ※2 “=” の前後にスペース(半角・全角問わず)を入れないこと
- ※3 OUTPUTに何も設定しない場合は、ログを出力しません。
- ※4 ログ種別不正値(1~7以外の数値、文字列)を指定した場合は該当値を無視します。
また、ログサイズ不正値(負のサイズ値、文字列)を指定した場合は
デフォルト値1MBを使用します。
- ※5 ログ種別4(IN)、5(OUT)に関して、データの転送量が多い場合に
SDKの動作が遅くなることがあります(ログ出力に時間がかかるため)
- ※6 ファイルサイズを大きく指定すると、ログファイルを開けなくなることがあります。
お使いの端末に応じて、ファイルサイズを変更して下さい。
- ※7 非同期書き込み時のファイルへの出力タイミングについては、ログ出力バッファに
セットされてから約3秒となります。(ログを出力する事によるタイミングずれを抑制する為)
- ※8 同期書き込み時にSDKがログファイルを読み書きできなくなった場合、読み書きが
できるまで関数がログ書き込み処理で止まります。
- ※9 ドライバがセットアップされている環境で同期書き込みを使用する際は、3.2.0.0以降の
ドライバを使用してください。

ログ出力について(出力ファイル)

なお、設定ファイルで指定されたバイト数を越えた場合には、ファイル名をSDKLog.bkに変更してから、新規空ファイルでログを作成します。

既にSDKLog.bkがあった場合は削除します(過去ログは1ファイルのみ保管)



【ログ出力例】

```
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.
```

西暦/月/日 時:分:秒.ミリ秒 で日時を出力します。
括弧で括られた中にはログ種別を表示します。

- 1 ERR : エラー
- 2 WRN : 警告
- 3 FNC : 関数呼び出し
- 4 I N : ポート受信データ
- 5 OUT : ポート送信データ
- 6 CHK : ステータス・拡張情報チェック
- 7 PSK : プリントステータス値変化時
拡張情報受信時

ログ設定ファイルでID=1に設定されている場合は日時の後ろに
<プロセスID,スレッドID>の形で情報を付与します。

バーコード設定ファイルについて

NBarcode関数でバーコードを描画する場合はバーコード設定ファイルの設置が必要です。
NBarcode2関数を使用する場合は設定ファイルではなくNBarcodeSettings関数でバーコードの設定を行う必要があります。

1. バーコード設定ファイルの配置

当SDKで“NBarcode”関数を使用する場合は設定ファイルを作成する必要があります。
このファイルはお客様にて作成して頂く必要があります。

バーコードファイル名 : NPIREG.inf

なお、ファイルは

```
OSドライブ¥NPI¥DriverInf¥プリンタ名¥BFONT¥F0
OSドライブ¥NPI¥DriverInf¥プリンタ名¥BFONT¥F1
OSドライブ¥NPI¥DriverInf¥プリンタ名¥BFONT¥F2
OSドライブ¥NPI¥DriverInf¥プリンタ名¥BFONT¥F3
OSドライブ¥NPI¥DriverInf¥プリンタ名¥BFONT¥F4
OSドライブ¥NPI¥DriverInf¥プリンタ名¥BFONT¥F5
OSドライブ¥NPI¥DriverInf¥プリンタ名¥BFONT¥F6
OSドライブ¥NPI¥DriverInf¥プリンタ名¥BFONT¥F7
OSドライブ¥NPI¥DriverInf¥プリンタ名¥BFONT¥F8
OSドライブ¥NPI¥DriverInf¥プリンタ名¥BFONT¥F9
OSドライブ¥NPI¥DriverInf¥プリンタ名¥2DBFONT¥F0
OSドライブ¥NPI¥DriverInf¥プリンタ名¥2DBFONT¥F1
OSドライブ¥NPI¥DriverInf¥プリンタ名¥2DBFONT¥F2
OSドライブ¥NPI¥DriverInf¥プリンタ名¥2DBFONT¥F3
OSドライブ¥NPI¥DriverInf¥プリンタ名¥2DBFONT¥F4
```

のフォルダを参照します。NPI以下のフォルダはお客様にて作成して下さい。
※前述のレジストリの設定で、OSドライブの部分は変更されます

バーコード設定ファイルについて

2. バーコード設定ファイルの記述方法(1次元バーコード)

以下の仕様にてBarcode1~Barcode10 まで下記形式にて設定して下さい。
なお、各項目は大文字・小文字を区別しますので間違いないように入力して下さい。

"Kind"=dword:00000000	対象のフォルダにNPIREG.infファイルが存在しない場合のデフォルト値
"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 (データ幅の倍率) 0=2dot 1=3dot 2=4dot Height データ高さ(ドット)
StartB, StopB (CODABARのみ) 0:A 1:B 2:C 3:D
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)
Spin 0:回転なし 1:90° 回転 2:180° 回転 3:270° 回転

3. バーコード設定ファイルの記述方法(2次元バーコード)

以下の仕様にて 2D-Barcode1~2D-Barcode5 まで下記形式にて設定して下さい。
なお、各項目は大文字・小文字を区別しますので間違いないように入力して下さい。

"Kind"=dword:00000000	対象のフォルダにNPIREG.infファイルが存在しない場合のデフォルト値
"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 (データ幅の倍率) < QRCode Model2 > 0:2倍 1:3倍 2:4倍 3:5倍 4:6倍 5:7倍 6:8倍 < PDF417 > 0:2倍 1:3倍 2:4倍
HEIGHT (PDF417のみ) 高さ比率を1~6の間で指定する。 HEIGHT = 高さ比率 × WIDTH
ECL (エラーレベル) < 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:回転なし 1:90° 回転 2:180° 回転 3:270° 回転

API一覧(1/2)

以下のAPIが用意されています。

用途	A P I	説明
コールバック関数指定	NSetCallback	コールバック関数を指定します
プリンタ情報取得	NEnumPrinters	当SDKで管理しているプリンタ名の一覧を取得します
プリンタ名取得	NGetPrinterFromID	各種IDからプリンタ名を取得します
プリンタ名削除	NDeletePrinter	当SDKで管理しているプリンタ名を削除します
プリンタ名変更	NRenamePrinter	当SDKで管理しているプリンタ名を変更します
プリンタ情報取得	NGetPrinterInf	プリンタ名から情報を取得します
自動プリンタ設定	NAutoOpen	(Ver.3.0.0.0より機能未対応となります)
プリンタオープン	NOpenPrinter	プリンタ名を指定し、オープンします
プリンタクローズ	NClosePrinter	既にオープンされているプリンタをクローズします
全プリンタクローズ	NClosePrinters	オープンされている全てのプリンタをクローズします
コマンド及びデータ送信	NPrint	指定された16進文字列データを変換してプリンタに送信します
コマンド及びデータ送信	NDPrint	指定されたデータをプリンタに送信します
イメージ送信	NImagePrint	指定されたデバイスコンテキストをコマンド変換してプリンタに送信します
イメージ送信	NImagePrintF	指定されたファイル(bmp)をコマンド変換してプリンタに送信します
ステータスの取得	NGetStatus	指定されたプリンタで取得されているステータスを返します
拡張情報の取得	NGetInformation	対象の種別IDに保存されている情報を取得します
プリンタのリセット	NResetPrinter	プリンタをリセットします(USB、TCP/UDP)
ドキュメント管理	NStartDoc	ドキュメントを開始します
ドキュメント管理	NEndDoc	ドキュメントを終了します
ドキュメント管理	NCancelDoc	ドキュメントをキャンセルします
ドキュメント管理	NEnumDoc	送信待ち状態のドキュメントリストを取得します
ドキュメント管理	NDeleteDoc	送信待ちドキュメントを削除します
バーコード設定	NBarcodeSettings	生成するバーコードイメージの設定をします
バーコード生成	NBarcode	バーコードイメージを生成します
バーコード生成	NBarcode2	バーコードイメージを生成します

API一覧(2/2)

用 途	A P I	説 明
F/W更新	NFirmwareDL	対象のfwfファイルでファームウェアを更新します
UDPスレッド操作	NInitializeNetwork	UDP受信スレッドを起動 / 停止します
TCP/UDP列挙	NScanPrinters	TCP/UDPプリンタを検出します
TCP通信ロック	NTCPPortLock	他プリンタのTCP通信をロック / アンロックします
TCPバッファクリア	NBufferClear	TCP/UDPプリンタの印刷バッファをクリアします
TCP一括送信	NBlockSendSetting	TCP/UDPプリンタの一括溜込送信を開始 / 終了します

Win32 APIリファレンス

インタフェース名		NCallback	
引数名	IN/OUT	型	説明
i_prt	I	PCHAR	プリンタ名
i_type	I	int	コールバック種別
i_value1	I	int	実行結果1
i_value2	I	int	実行結果2
戻り値	void		

・なし

処理内容

・コールバック通知の関数形式になります。

当宣言と同じ戻り値・引数で関数を宣言し、その関数ポインタをNSetCallback関数の引数に指定します。またNOpenPrinter関数の第3引数に指定し、オープンと同時にコールバックを指定することもできます。

非同期関数を使用される際は、結果をコールバックで取得する形となります。

必ず先にNSetCallback関数を実行して下さい。

コールバック通知の条件は以下の通りです。

コールバック条件	i_prt	i_type	i_value1	i_value2
ステータス値が変化した時 (初回受信時含む)	プリンタ名	1	旧ステータス値	新ステータス値
拡張情報受信時	プリンタ名	2	拡張情報ID	0x00
送信データキューが空になった時	プリンタ名	3	0x00	0x00
NScanPrinters結果通知	NULL	4	実行結果	検出件数
NOpenPrinter結果通知	プリンタ名	5	実行結果	0x00
NResetPrinter結果通知	プリンタ名	6	実行結果	0x00
NFirmwareDL結果通知	プリンタ名	7	実行結果	0x00
NTCPPortLock結果通知	プリンタ名	8	実行結果	0x00
NBufferClear結果通知	プリンタ名	9	実行結果	0x00
NBlockSendSetting結果通知	プリンタ名	10	実行結果	0x00

インタフェース名	NCallback
処理内容	

注意

コールバック関数内で、画面制御などの他のプロセスとの同期制御を必要とする処理や、NOpenPrinter、NSetCallBackは実行しないでください

関数名		NSetCallback		
引数名	IN/OUT	型	説明	
i_callback	I	NCallback	コールバック関数ポインタ	
戻り値	void			
・なし				
処理内容	<p>・コールバック関数を設定します。 コールバック通知を行う前に当該関数を実行して下さい。 なお、設定は“NOpenPrinter”関数でも行うことができます。</p> <p>注意</p> <p>当該関数でnullを指定しない限り、一度設定された関数を呼んでコールバック通知を続けます。 設定したポインタの関数実体がない場合にはSDKから例外が発生されますので、ご注意ください。</p> <p>SDK使用アプリ側にコールバック関数を実装してください。</p> <p>コールバック関数内で、画面制御などの他のプロセスとの同期制御を必要とする処理や、NOpenPrinter、NSetCallBackは実行しないでください</p>			

Win32 APIリファレンス

関数名	NEnumPrinters / NEnumPrintersA		
引数名	IN/OUT	型	説明
o_printers	O	PWCHAR / PCHAR	プリンタ名 (csvカンマ区切り形式で列挙) (NULL指定可)
o_size	O	PINT	o_printersのバイトサイズ
戻り値	INT		
・エラー(-値)、正常終了(0) ※エラーコード表を参照			
処理内容	<p>・システムドライブ内のNPIフォルダ配下にプリンタ情報管理ファイル (NPrinterInf.inf)を作成し、接続可能なプリンタ名の一覧を引数 o_printers に格納します。 例) PRT001,PRT002,AAA</p> <p>利用可能ポートを検出して、“PRTxxx“(xxxは001~999の数値)の名前を割り当て、csv(カンマ区切り)で列挙して返します。 ※アクティブなプリンタのみ列挙されます</p> <p>プリンタ情報管理ファイルが未作成の段階では、プリンタのオープン(NOpenPrinter)を行うことは出来ませんので、必ず当関数を初回使用時に呼出して下さい。 プリンタの追加があった場合にも呼出する必要があります。 一度作成したプリンタ名を使用する場合は、再度当関数を呼出する必要はありません。</p> <p>なお、当関数から得られるプリンタ名一覧は、プリンタ情報管理ファイルを直接参照することでも得られます。ただし、ドライバ名はファイルには記録されません。</p> <p>※Bluetoothで使用される場合は当関数を呼出する前にプリンタとのペアリングを済ませておく必要があります。</p> <p>※一度生成されたプリンタ名はBluetoothのペアリングを解除しても削除されません。 削除したい場合はNDeletePrinter関数を使用して下さい。</p> <p>※プリンタ名割り当てのxxxは最大999まで生成でき、それ以上は生成エラーとなります。 NDeletePrinterで不要なプリンタ名を削除する、またはNRenamePrinterでプリンタ名を書き換える処理を行う必要があります。</p>		

関数名	NEnumPrinters / NEnumPrintersA
処理内容	

注意

・ドライバがセットアップされている時、ドライバのプリンタ名と、直接通信のプリンタ名が o_printers に格納されます。(1台のプリンタがドライバ、直接通信の両方で格納)

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

ドライバ(USB)プリンタAがある場合、太字部はどちらもプリンタAとなります。

※ドライバをセットアップしたポートに対して直接通信では使用しないでください

参考

※本関数はドライバポートを使用する場合には呼び出す必要はありません。

Win32 APIリファレンス

関数名	NGetPrinterFromID / NGetPrinterFromIDA		
引数名	IN/OUT	型	説明
i_ID	I	PWCHAR / PCHAR	ID
o_printer	O	PWCHAR / PCHAR	プリンタ名 (NULL指定可)
戻り値	INT		
・エラー(-値)、正常終了(プリンタ名保存領域サイズ)			
処理内容	<p>・引数の i_ID に以下の各IDを指定することで、引数 o_printer にプリンタ名を格納します。 各種IDはプリンタのセルフ印字を行うことで確認できます。</p> <p><各種インターフェースで指定するID> WLAN:MACアドレス Bluetooth:シリアルID USB:ベンダID4桁(1051)+プロダクトID4桁(1000/1001/1002/1003)</p> <p>注意</p> <p>プリンタ名は予めNEnumPrinters関数にてプリンタ情報管理ファイルに作成されている必要があります。</p>		

関数名	NDeletePrinter / NDeletePrinterA		
引数名	IN/OUT	型	説明
i_prt	I	PWCHAR / PCHAR	削除するプリンタ名
戻り値	INT		
・エラー(-値)、正常終了(0) ※エラーコード表を参照			
処理内容	<p>・プリンタ情報管理ファイル内のプリンタ名を削除する際に使用します。 引数のi_prtに削除したいプリンタ名を指定します。 また空文字を指定することでプリンタ情報管理ファイルそのものを削除します。 (プリンタ名全削除:プリンタドライバは消えません)</p> <p>また、削除したいプリンタがオープンされている場合は、エラーを返します。 該当プリンタとの接続をクローズした状態で当関数を使用してください。</p>		
<div style="border: 1px solid black; border-radius: 10px; padding: 5px; display: inline-block;">注意</div> <p>プリンタドライバを使用しているプリンタ名は消えません。</p>			

関数名		NRenamePrinter / NRenamePrinterA	
引数名	IN/OUT	型	説明
i_beforeprt	I	PWCHAR / PCHAR	変更するプリンタ名
i_afterprt	I	PWCHAR / PCHAR	変更後のプリンタ名
戻り値	INT		
・エラー(-値)、正常終了(0) ※エラーコード表を参照			
処理内容	<p>・プリンタ情報管理ファイル内のプリンタ名を書き換える際に使用します。 デフォルトでは“PRT001”のようにPRT~で始まる形で作成されていますが、この関数を使用することで任意のプリンタ名に変更することが出来ます。 なお、プリンタ名は半角英数字で50文字までとし、以下の文字は使用不可とします。</p> <p>使用禁止文字(半角スペース含む) 「」¥/:?*”<> ’ , . □</p> <p>また、名称を書き換えたいプリンタがオープンされている場合は、エラーを返します。 該当プリンタとの接続をクローズした状態で当関数を使用して下さい。</p> <p>注意</p> <p>ドライバ名は当SDKの管理ではないため、書き換えはできません。</p>		

Win32 APIリファレンス

関数名		NGetPrinterInf / NGetPrinterInfA	
引数名	IN/OUT	型	説明
i_prt	I	PWCHAR / PCHAR	プリンタ名
o_ports	O	PWCHAR / PCHAR	プリンタ情報(csvカンマ区切り形式で列挙) (NULL指定可)
o_size	O	PDWORD	o_portsのバイトサイズ
戻り値	INT		
・エラー(-値)、正常終了(0) ※エラーコード表を参照			
処理内容	<p>・引数で指定されたプリンタ名からプリンタ情報管理ファイル内を検索し、以下の情報をo_portsに格納します。</p> <ul style="list-style-type: none"> ・接続種別 <ul style="list-style-type: none"> USB : 1 Bluetooth : 2 RS232C : 3 TCP/UDP : 4 NPI ドライバ : DRIVER ZEBRA ドライバ : ZEBRADRIVER 他プリンタドライバ : OTHERDRIVER ・接続情報 (USB : デバイスインスタンス <ul style="list-style-type: none"> Bluetooth : デバイスインスタンスID RS232C : シリアルポート TCP/UDP : モデル名、IPアドレス、ポート番号、MACアドレス ドライバ類 : プリンタ名 <p>例) 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 DRIVER:NPI Integration Driver</p>		
<div style="border: 1px solid black; border-radius: 10px; padding: 5px; display: inline-block; margin-bottom: 10px;"> 参考 </div> <p>o_portsにて得られる情報はプリンタ情報管理ファイルを参照することでも得られます。</p>			

関数名	NAutoOpen		
引数名	IN/OUT	型	説明
i_flg	I	INT	—
戻り値	BOOL		
・常にFALSE: 自動オープン無効			
処理内容	<p>・Ver.3.0.0.0より当関数は廃止されています。 前バージョンとの互換性のため、関数自体は残していますが、戻り値は常にFALSE(自動オープン無効)で返されます。</p> <p>NOpenPrinterを実行し、オープン処理を行って下さい。</p> <p>参考</p> <p>※本関数はドライバポートを使用する場合には呼び出す必要はありません。</p>		

関数名	NOpenPrinter / NOpenPrinterA		
引数名	IN/OUT	型	説明
i_prt	I	PWCHAR/PCHAR	オープンするプリンタ名
i_statusFlg	I	BYTE	ステータス受信フラグ
i_callback	I	SDK.NCallback	コールバックインタフェースを実装した関数ポインタ(NULL指定可)

戻り値 INT

- ・エラー(-値)、正常終了(0) ※エラーコード表を参照
コールバック設定時は非同期処理が開始した時点で正常終了(0)を返す

処理内容 **非同期可能関数**

- ・プリンタのオープン処理を行います。
引数 i_prt には NEnumPrinters で取得したプリンタ名を指定します。
- ・送信待ちのドキュメントは削除されます。
- ・i_statusFlg はプリンタからのステータス受信処理方法を指定します。
 - 1: ステータス受信を行い、送信処理を行う。
 - 2: ステータス受信を行い、プリンタの残容量バッファを確認して送信処理を行う。
 - 3: ステータス受信を行わない。(プリンタ言語がXML、ZPLの際はこちらを使用してください)
i_callback の値に関わらず同期処理でプリンタのオープン処理を行います。
ステータス受信を行わないのでコールバック通知も行われません。
※直接通信でのみ使用可能です。
- ・i_callback に関数ポインタを渡した場合、当関数は非同期処理でオープンを行いません。
非同期処理開始前のエラーは当関数の戻り値で返しますが、戻り値が0(正常終了)で返った場合、実行結果はコールバック関数で取得してください。
- ・オープン後はステータスが取得できることを確認の上、データ送信を行なってください。
ステータス取得前に通信を行うとデータ損失の恐れがあります。

参考

※本関数はドライバポートを使用する場合には呼び出す必要はありません。

関数名	NClosePrinter / NClosePrinterA		
引数名	IN/OUT	型	説明
i_prt	I	PWCHAR / PCHAR	オープン済みプリンタ名
戻り値	INT		
・エラー(-値)、正常終了(0) ※エラーコード表を参照			
処理内容	<p>・当関数でプリンタのクローズ処理を行います。</p> <p>注意</p> <ul style="list-style-type: none">・印字が終わったことを確認してから当関数を呼び出して下さい。 「印字開始/終了」コマンドを使用し、ステータスを監視することで印字が終了したことを得ることができます。 詳しくは各プリンタの仕様書を参照して下さい。・送信待ちのドキュメントは削除されます。 <p>参考</p> <p>※本関数はドライバポートを使用する場合には呼び出す必要はありません。</p>		

関数名	NClosePrinters		
引数名	IN/OUT	型	説明
戻り値	INT		
・正常終了(0)			
処理内容	<p>・オープン済みのプリンタを全てクローズします。</p> <p>注意</p> <p>当関数はクローズできるプリンタがない場合を除き戻り値として成功のみ返します。 エラー値を取得したい場合はNClosePrinter(1台のみクローズ)関数を使用して下さい。</p>		

関数名		NPrint / NPrintA	
引数名	IN/OUT	型	説明
i_prt	I	PWCHAR / PCHAR	出力プリンタ名
i_dat	I	PCHAR	送信データ(16進文字列)
i_size	I	DWORD	出力バイト数
o_jobid	O	PINT	プリントジョブID(NULL指定可)

戻り値 INT

・エラー(-値)、正常終了(0) ※エラーコード表を参照

処理内容

・指定された16進文字列データをバイナリデータに変換してプリンタに送信します。

なお、“(ダブルコーテーション)、<>(大なり、小なり)、[] (大括弧)で括られたデータと ‘(シングルコーテーション)が先頭にあるデータは、以下の通り変換して送信します

1. “(ダブルコーテーション)で括られた文字列
⇒ 文字列として変換します。(” ABC “ ⇒ 0x41, 0x42, 0x43)
2. <>(大なり、小なり)で括られたファイル名(パス込み)
⇒ ファイル内容(バイナリデータ)を出力します。
3. [] (大括弧)で括られた画像ファイル名(パス込み、bmp形式)
⇒ 画像をラスタビットイメージコマンドに変換して出力します。
4. ‘(シングルコーテーション)が先頭にある文字列
⇒ コメントとして処理されます(出力されない)

注意

・逐次印字と、一括印字について

NPrint及びNImagePrint、NImagePrintFを単独で実行するとその都度プリンタへデータ送信されます。NStartDoc、NEndDocを使用することで、プリンタへのデータ送信を一括送信することが可能となります。

利用目的に合わせて、逐次印字、一括印字をご利用ください。

関数名		NDPrint / NDPrintA	
引数名	IN/OUT	型	説明
i_prt	I	PWCHAR / PCHAR	出力プリンタ名
i_dat	I	PBYTE	送信データ(16進文字列)
i_size	I	DWORD	出力バイト数
o_jobid	O	PINT	プリントジョブID(NULL指定可)
戻り値	INT		
・エラー(-値)、正常終了(0) ※エラーコード表を参照			
処理内容	・指定されたデータをプリンタに送信します(変換なし)		
<div style="border: 1px solid black; border-radius: 10px; padding: 5px; display: inline-block;">注意</div> <p>・逐次印字と、一括印字について NPrint及びNImagePrint、NImagePrintF を単独で実行するとその都度プリンタへデータ送信されます。NStartDoc,NEndDocを使用することで、プリンタへのデータ送信を一括送信することが可能となります。 利用目的に合わせて、逐次印字、一括印字をご利用ください。</p>			

関数名		NImagePrint / NImagePrintA	
引数名	IN/OUT	型	説明
i_prt	I	PWCHAR / PCHAR	出力プリンタ名
i_bmp	I	HDC	デバイスコンテキストハンドル
i_width	I	INT	幅
i_height	I	INT	高さ
i_putType	I	BYTE	出力方式 0x00:ラスタ形式ライン単位 0x01:ラスタ形式ブロック単位 0x02:ラスタ形式ブロック単位階調表現 0x10:ビットイメージ形式
o_jobid	O	PINT	プリントジョブID(NULL指定可)
戻り値	INT		
・エラー(-値)、正常終了(0) ※エラーコード表を参照			
処理内容	<p>・指定されたデバイスコンテキストをプリンタに送信します。</p>		
<div style="border: 1px solid black; border-radius: 10px; padding: 5px; display: inline-block; margin-bottom: 10px;">注意</div> <p>・逐次印字と、一括印字について NPrint及びNImagePrint、NImagePrintF を単独で実行するとその都度プリンタへデータ送信されます。NStartDoc,NEndDocを使用することで、プリンタへのデータ送信を一括送信することが可能となります。 利用目的に合わせて、逐次印字、一括印字をご利用ください。</p>			

関数名		NImagePrintF / NImagePrintFA	
引数名	IN/OUT	型	説明
i_prt	I	PWCHAR / PCHAR	出力プリンタ名
i_bmp	I	PCHAR	BMPファイル名
i_putType	I	BYTE	出力方式 0x00:ラスタ形式ライン単位 0x01:ラスタ形式ブロック単位 0x02:ラスタ形式ブロック単位階調表現 0x10:ビットイメージ形式ライン単位
o_jobid	O	PINT	プリントジョブID(NULL指定可)
戻り値	INT		
・エラー(-値)、正常終了(0) ※エラーコード表を参照			
処理内容	<p>・指定されたBMPファイルを読み込んでイメージをプリンタに送信します。</p>		
<div style="border: 1px solid black; border-radius: 10px; padding: 5px; width: fit-content; margin-bottom: 10px;">注意</div> <p>・逐次印字と、一括印字について NPrint及びNImagePrint、NImagePrintF を単独で実行するとその都度プリンタへデータ送信されます。NStartDoc,NEndDocを使用することで、プリンタへのデータ送信を一括送信することが可能となります。 利用目的に合わせて、逐次印字、一括印字をご利用ください。</p>			

関数名		NGetStatus / NGetStatusA		
引数名	IN/OUT	型	説明	
i_prt	I	PWCHAR / PCHAR	出力プリンタ名	
o_status	O	LPDWORD	ステータス	
戻り値	INT			
・エラー(-値)、正常終了(0) ※エラーコード表を参照				
処理内容	<ul style="list-style-type: none">・指定されたプリンタで取得されているステータスを返します。 ※返される値は、対象プリンタの製品仕様書「エラーの処理」を参照の事。・戻り値が負数であっても、N_ERR_SEND_ERROR (-3) の時はo_statusでプリンタステータスを返しています。(送信エラー原因判別の為)・送信エラー中でオフラインだった場合は、戻り値はN_ERR_SEND_ERROR (-3)でo_statusが0xffffffff(オフライン)を返します。			

Win32 APIリファレンス

関数名		NGetInformation / NGetInformationA	
引数名	IN/OUT	型	説明
i_prt	I	PWCHAR / PCHAR	出力プリンタ名
i_id	I	BYTE	種別ID
o_dat	O	PVOID	拡張情報格納エリア
o_time	O	PDWORD	更新フラグ(システムを起動した後の経過時間) (NULL指定可)
戻り値	INT		
・エラー(-値)、正常終了(0) ※エラーコード表を参照			
処理内容	<p>・拡張情報の対象種別IDに保存されている情報を取得する</p> <p>※予め上位アプリケーションからほしい情報の要求をプリンタに送る必要があります。 (拡張ステータス、転送完了、印字完了情報等要求を必要としないものもあります)</p> <p>※返される値は、対象プリンタの仕様書の《ESC s》コマンド参照の事。</p>		

Win32 APIリファレンス

拡張情報

種別 1	: 4 バイト (固定)	: 更新フラグ (4バイト)	<拡張ステータス> 1Byte : 7~0, 2Byte : 15~8, 3Byte : 23~16, 4Byte : 31~24
種別 2	: 32バイト (デリミタ)	: 更新フラグ (4バイト)	<モデル名>
種別 3	: 8 バイト (固定)	: 更新フラグ (4バイト)	<F/Wバージョン>
種別 4	: 8 バイト (固定)	: 更新フラグ (4バイト)	<Bootバージョン>
種別 5	: 4 バイト (固定)	: 更新フラグ (4バイト)	<予約>
種別 6	: 4 バイト (固定)	: 更新フラグ (4バイト)	<ヘッド通電ドットライン数>
種別 7	: 4 バイト (固定)	: 更新フラグ (4バイト)	<走行ドットライン数>
種別 8	: 4 バイト (固定)	: 更新フラグ (4バイト)	<カット回数>
種別 9	: 16バイト (固定)	: 更新フラグ (4バイト)	<ユーザメンテナンスカウンタ : ヘッド通電ドットライン数, 走行ドットライン数, カット回数, 予備>
種別 10	: 16バイト (固定)	: 更新フラグ (4バイト)	<予約>
種別 11	: 64バイト (デリミタ)	: 更新フラグ (4バイト)	
種別 12	: 32バイト (デリミタ)	: 更新フラグ (4バイト)	
種別 13	: 32バイト (固定)	: 更新フラグ (4バイト)	<NV登録状況>
種別 14	: 32バイト (固定)	: 更新フラグ (4バイト)	<予約>
種別 15	: 16バイト (固定)	: 更新フラグ (4バイト)	
種別 16	: 16バイト (固定)	: 更新フラグ (4バイト)	
種別 17	: 16バイト (固定)	: 更新フラグ (4バイト)	
種別 18	: 16バイト (固定)	: 更新フラグ (4バイト)	
種別 19	: 8 バイト (固定)	: 更新フラグ (4バイト)	<印字完了通知 : 印字開始/終了コマンドの指定により、終了コマンド処理時に任意ID, 終了ステータス記載>
種別 20	: 8 バイト (固定)	: 更新フラグ (4バイト)	<予約>
種別 21	: 8 バイト (固定)	: 更新フラグ (4バイト)	
種別 22	: 8 バイト (固定)	: 更新フラグ (4バイト)	
種別 23	: 8 バイト (固定)	: 更新フラグ (4バイト)	
種別 24	: 4 バイト (固定)	: 更新フラグ (4バイト)	<予約>
種別 25	: 4 バイト (固定)	: 更新フラグ (4バイト)	<転送完了通知 : 転送の完了したジョブID記載>
種別 26	: 4 バイト (固定)	: 更新フラグ (4バイト)	<予約>
種別 27	: 4 バイト (固定)	: 更新フラグ (4バイト)	<予約>
種別 28	: 2 バイト (固定)	: 更新フラグ (4バイト)	<F/Wチェックサム>
種別 29	: 2 バイト (固定)	: 更新フラグ (4バイト)	
種別 30	: 2 バイト (固定)	: 更新フラグ (4バイト)	
種別 31	: 2 バイト (固定)	: 更新フラグ (4バイト)	<通信状態情報 : USB : 0x0000固定 COM : 1バイト目CTS 2バイト目DSR ※更新フラグに最終ステータス取得タイムスタンプ>

※種別25、31を除き、リターにて機能実装していない情報については取得できた内容に妥当性はありません。
※ここで記載した内容が全てのプリンタで使用出来るとは限りません。

種別 32 : 8 バイト16進文字列/文字列
種別 33 : 4 バイト16進文字列/文字列
種別 34 : 2 バイト16進文字列/文字列
種別 35 : 8 バイト16進文字列
種別 36 : 8 バイト16進文字列
種別 37 : 4 バイト16進文字列
種別 38 : 4 バイト16進文字列
種別 39 : 2 バイト16進文字列
種別 40 : 2 バイト16進文字列

※種別11/32/33/34はサブIDによって返信される情報は変更されます

ネットワーク関連拡張情報種別一覧

MACアドレス (6バイト)	種別32-0	応答 : 16進文字列
IPアドレス (4バイト)	種別33-0	応答 : 16進文字列
サブネットマスク (4バイト)	種別33-1	応答 : 16進文字列
デフォルトゲートウェイ (4バイト)	種別33-2	応答 : 16進文字列
DNSサーバアドレス (4バイト×5)	種別33-3	応答 : 16進文字列
印刷タイムアウト (ミリ秒単位 : 4バイト)	種別33-14	応答 : 16進文字列
通信モード (インフラストラクチャー, アドホック)	種別34-4	応答 : 16進文字列
バンド (2.4GHz/5GHz)	種別34-5	応答 : 16進文字列
送信パワーレベル (Low/Medium/High)	種別34-6	応答 : 16進文字列
チャンネル	種別34-7	応答 : 16進文字列
セキュリティタイプ (OPEN/WPA/WPA2/WEP)	種別34-8	応答 : 16進文字列
暗号化タイプ (OPEN/TKIP/AES)	種別34-9	応答 : 16進文字列
IP設定方法 (自動 (DHCP/APIPA) /手動)	種別34-10	応答 : 16進文字列

Win32 APIリファレンス

関数名	NResetPrinter / NResetPrinterA		
引数名	IN/OUT	型	説明
i_prt	I	PWCHAR / PCHAR	出力プリンタ名
戻り値	INT		
・エラー(-値)、正常終了(0) ※エラーコード表を参照 コールバック設定時は非同期処理が開始した時点で正常終了(0)を返す			
処理内容	非同期可能関数		
<ul style="list-style-type: none">・プリンタをリセットします。印字中の印刷ジョブもキャンセルされます。・本API が正常に実行されたか確認するには、本APIの戻り値および、プリンタがリセットされてオンラインになる事を確認してください(NGetStatusで確認)・コールバック関数設定時、当関数は非同期で処理を行いません。 非同期処理開始前のエラーは当関数の戻り値で返しますが、戻り値が0(正常終了)で返った場合、実行結果はコールバック関数で取得してください。・ドライバ・USBのリセット処理は1回あたり最大8秒の完了待ちを行い、8秒以内にリセット処理完了を確認できなかった場合は再度リセット処理を行う為、最大16秒の待ち時間が発生する場合があります。・ドライバ・USBのリセット処理はNResetPrinter.exeを呼び出してリセット処理を実行します。 ドライバセットアップ時は自動的にNResetPrinter.exeが配置されますが、ライブラリ単体での使用時は「ライブラリのみで使用する場合」を参照の元NResetPrinter.exeを配置してください。			
<div style="border: 1px solid black; border-radius: 10px; padding: 5px; display: inline-block;">注意</div> <ul style="list-style-type: none">・リセットが行えるのはドライバ、USB、TCP/UDP接続のみとなります。			

Win32 APIリファレンス

関数名		NStartDoc / NStartDocA	
引数名	IN/OUT	型	説明
i_prt	I	PWCHAR / PCHAR	出力プリンタ名
o_jobid	O	PINT	プリントジョブID (NULL指定可)
戻り値	INT		
・エラー(-値)、正常終了(0) ※エラーコード表を参照			
処理内容	<p>・ドキュメントを開始します。 NStartDoc発行後はNPrint、NDPrint、NImagePrint、NImagePrintFについてはデータを一時バッファ内に溜め込みます。</p> <p>NEndDocを呼出することで溜め込んだデータをプリンタへ出力することができます。 また、NCancelDocを呼出することで溜め込んだデータをクリアします。</p> <p>1プリンタにつき、1ドキュメントとします。</p>		

Win32 APIリファレンス

関数名	NEndDoc / NEndDocA		
引数名	IN/OUT	型	説明
i_prt	I	PWCHAR / PCHAR	出力プリンタ名
戻り値	INT		
・エラー(-値)、正常終了(0) ※エラーコード表を参照			
処理内容	<p>・ドキュメントを終了します NStartDoc呼出後に溜め込んだデータをプリンタに出力します。 溜め込んだデータが存在しない場合は、処理を行いません。</p>		

関数名	NCancelDoc / NCancelDocA		
引数名	IN/OUT	型	説明
i_prt	I	PWCHAR / PCHAR	出力プリンタ名
戻り値	INT		
・エラー(-値)、正常終了(0) ※エラーコード表を参照			
処理内容	<ul style="list-style-type: none">・ドキュメントをキャンセルします溜め込んだデータが存在しない場合には、処理を行いません。		

関数名		NEnumDoc / NEnumDocA	
引数名	IN/OUT	型	説明
i_prt	I	PWCHAR / PCHAR	出力プリンタ名
o_list	O	PWCHAR / PCHAR	ジョブIDリスト(CSV)(NULL指定可)
戻り値	INT		
・エラー(-値)、送信待ち無し(0)、正常終了(リスト保存のバイトサイズ)			
処理内容	<ul style="list-style-type: none">・送信待ち状態のジョブIDリスト(CSV)を返します。※ドキュメントステータス, ジョブID 改行を1ドキュメントの情報とします※ドキュメントステータス 0:送信待ち 1:送信中 2:送信エラー※o_listをNULLで指定する事でリスト保存に必要なサイズの取得のみ行います		

関数名		NDeleteDoc / NDeleteDocA	
引数名	IN/OUT	型	説明
i_prt	I	PWCHAR / PCHAR	出力プリンタ名
ijobID	I	INT	プリントジョブID
戻り値	INT		
・エラー(-値)、正常終了(0) ※エラーコード表を参照			
処理内容	<ul style="list-style-type: none">送信待ちドキュメントを削除します送信待ちドキュメントが存在しない場合には、処理を行いません。		
<div style="border: 1px solid black; border-radius: 10px; padding: 5px; display: inline-block;">注意</div> <p>ドライバポートを使用している場合は本関数は無効です</p>			

Win32 APIリファレンス

関数名		NBarcodeSettings		
引数名	IN/OUT	型	説明	
i_id	I	UINT	バーコード種別	
i_width	I	UINT	データ幅の倍率	
i_height	I	UINT	1モジュールの高さ	
i_option1	I	CHAR	オプション1	
i_option2	I	CHAR	オプション2	
i_option3	I	CHAR	オプション3	
i_option4	I	CHAR	オプション4	
i_hri	I	UINT	HRI文字の位置	
i_spin	I	UINT	回転	
i_code	I	UINT	文字コード種別	
戻り値	INT			
・エラー(-値)、正常終了(0) ※エラーコード表を参照				
処理内容				
<p>・NBarcode2関数を呼び出す前に当関数で出力設定を行います。</p> <p>・使用しない項目の値はNULLでも構いません。その場合デフォルトの値が適用されます。</p> <p>・CODE39、CODE93、CODE128のi_idはオプションが固定されているものと指定したオプションで生成する2種類があります。 オプション固定の場合、CODE39はチェックディジットとスタートコードの自動付与無効、 CODE93はHRI文字の前後へ□の付与有効、 CODE128はスタートコードやSHIFTの手動入力有効となります。</p> <p>・各項目の設定値は次頁のバーコード設定項目一覧を参照ください。</p>				

Win32 APIリファレンス

バーコード設定項目一覧

<p>i_id (バーコード種別) (デフォルト0)</p>	<p>1次元バーコード</p> <p>0 UPC-A 1 UPC-E 2 JAN-13(EAN-13) 3 JAN-8(EAN-8) 4 CODE39(オプション固定) 5 ITF 6 CODABAR(NW-7) 7 CODE128(オプション固定) 8 CODE93(オプション固定) 9 ISBN 10 Industrial 2 of 5 11 Matrix 2 of 5 12 NEC Matrix 13 カスタマバーコード 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(オプション指定有) 21 CODE128(オプション指定有) 22 CODE93(オプション指定有)</p>	<p>2次元バーコード</p> <p>30 QRコード 31 マイクロQRコード 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コード 53 GS1 Aztec Code 54 GS1 DataMatrix 55 連結QRコード</p>
<p>i_width (デフォルト3)</p>	<p>1モジュールあたりの横幅 1~20dot</p>	
<p>i_height (デフォルト162)</p>	<p>バーの高さ(dot) 上限3776dot ※1次元バーコードのみ指定可能</p>	
<p>i_option1~4</p>	<p>option4は1~6の間で設定 (option1 デフォルト0,option2 デフォルト0,option3 デフォルト0,option4 デフォルト3) 他詳細は次頁参照</p>	
<p>i_hri (デフォルト0)</p>	<p>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) 7: Top (Font D) 8: Bottom (Font D) 9: Top & Bottom (Font D)</p> <p>※1次元バーコードと一部2次元バーコードのみ指定可能です。 対応2次元バーコード (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)</p>	
<p>i_spin (デフォルト0)</p>	<p>0: 回転なし 1: 90° 回転 2: 180° 回転 3: 270° 回転</p>	
<p>i_code (デフォルト0)</p>	<p>0: ISO-8859-1 1: Shift-JIS 2: UTF-8 ※Shift-JISとUTF-8はQRコード、PDF417、MicroPDF417、Maxicodeのみ指定可能。</p>	

Win32 APIリファレンス

オプション一覧

記載されていない値を設定した場合はオプションが無視されます。

1次元バーコードオプション

バーコード体系	オプション	
CODE39	Option1	自動チェックディジット付与=有効(1) 自動スタートコード付与=有効(2) 上記同時有効(3)
CODE93	Option1	HRI文字の前後に□を付ける=有効(2)
CODE128	Option1	スタートコードやSHIFTの手動入力=有効(2)
MSI Code	Option1	チェックディジットの有無 無効(0),Mod-10(1),Mod-1010(2),Mod-11(3),Mod-1110(4)

2次元バーコードオプション

バーコード体系	オプション	
QRコード	Option1	誤り訂正レベル(-1,0=自動,1=L, 2=M, 3=Q, 4=H)
	Option2	バージョン指定(0=自動,1~40=指定) 1~40は21×21~177×177のモジュールを指定します。 (型番が1つ増える毎に4モジュール増えます。)
	Option3	マスクパターン(0=自動,1~8=指定)
マイクロQRコード	Option1	誤り訂正レベル(-1,0=自動,1=L, 2=M, 3=Q)
	Option2	バージョン指定(0=自動,1~4=指定) 1~4は11×11~17×17モジュールを指定します。 (型番が1つ増える毎に2モジュール増えます。)
Aztec Code	Option1	誤り訂正レベル(-1,0=自動,1~4=指定)
	Option2	サイズ指定(0=自動,1~36=指定) 1~4は15×15~27×27モジュールのCompact Aztec Code を指定します。 5~36は19×19~151×151モジュールのAztec Code を指定します。 (型番が1つ増える毎に4モジュール増えます。)
	Option3	リーダー初期化データの付与(1=有効) サイズ自動時にCompact Aztecを生成しない(2=有効) 上記同時指定=3

Win32 APIリファレンス

バーコード体系	オプション						
PDF417	Option1	チェックデジット数の指定(-1=自動,0~8=指定) (-1=自動,0=2文字,1=4文字,2=8文字,3=16文字,4=32文字, 5=64文字,6=128文字,7=256文字,8=512文字)					
	Option2	データ部の列数指定(0=自動,1~30=列数指定)					
	Option3	リーダー初期化データの付与(0=無効, 1=有効)					
	Option4	高さ比指定(1~6=指定,未指定時3)					
Micro PDF417	Option2	データ部の列数指定(0=自動,1~4=列数指定) 指定に収まらない場合は自動					
	Option3	リーダー初期化データの付与(0=無効, 1=有効)					
MaxiCode	Option1	モードの選択(-1,0=自動, 2~6=指定) 2 = 1次メッセージを9桁の郵便番号として扱う 3 = 1次メッセージを6桁の郵便番号として扱う 4 = 誤り訂正レベル(1次メッセージEEC,2次メッセージSEC) 5 = 誤り訂正レベル(全文EEC) 6 = 誤り訂正レベル(全文SEC)					
	Option2	1次メッセージ(0=無効, 1=有効) i_datの先頭15桁を1次メッセージとするため16桁以上の入力が必要となる					
DataMatrix	Option1	リーダー初期化データの付与(-1=無効, 1=有効)					
	Option2	サイズ指定(0=自動, 1~39=指定)					
		1	10 x 10	14	48 x 48	27	12 x 26
		2	12 x 12	15	52 x 52	28	12 x 36
		3	14 x 14	16	64 x 64	29	16 x 36
		4	16 x 16	17	72 x 72	30	16 x 48
		5	18 x 18	18	80 x 80	31	8 x 48
		6	20 x 20	19	88 x 88	32	8 x 64
		7	22 x 22	20	96 x 96	33	12 x 64
		8	24 x 24	21	104 x 104	34	16 x 64
		9	26 x 26	22	120 x 120	35	24 x 48
		10	32 x 32	23	132 x 132	36	24 x 64
		11	36 x 36	24	144 x 144	37	26 x 40
		12	40 x 40	25	8 x 18	38	26 x 48
13	44 x 44	26	8 x 32	39	26 x 64		
Option3	サイズ自動時に長方形で生成するか 1=正方形でのみ生成する 2=長方形の方がサイズが小さい場合長方形で生成する						

Win32 APIリファレンス

バーコード体系	オプション	
GS1 DataBar Stacked	オプション無し	
GS1 DataBar Stacked Omnidirectional	オプション無し	
GS1 DataBar Expanded Stacked	Option2	データ部の列数の指定 (1~10=列数の指定、未指定時2列)
EAN-8 Composite	Option1	コンポジットコンポーネントの種類 (-1=自動, 1=CC-A,2=CC-B)
EAN13 Composite	Option1	コンポジットコンポーネントの種類 (-1=自動, 1=CC-A,2=CC-B)
UPC-A Composite	Option1	コンポジットコンポーネントの種類 (-1=自動, 1=CC-A,2=CC-B)
UPC-E Composite	Option1	コンポジットコンポーネントの種類 (-1=自動, 1=CC-A,2=CC-B)
GS1-128 Composite	Option1	コンポジットコンポーネントの種類 (-1=自動, 1=CC-A,2=CC-B,3=CC-C)
GS1 DataBar Omnidirectional Composite	Option1	コンポジットコンポーネントの種類 (-1=自動, 1=CC-A,2=CC-B)
GS1 DataBar Truncated Composite	Option1	コンポジットコンポーネントの種類 (-1=自動, 1=CC-A,2=CC-B)
GS1 DataBar Limited Composite	Option1	コンポジットコンポーネントの種類 (-1=自動, 1=CC-A,2=CC-B)
GS1 DataBar Expanded Composite	Option1	コンポジットコンポーネントの種類 (-1=自動, 1=CC-A,2=CC-B)
GS1 DataBar Stacked Composite	Option1	コンポジットコンポーネントの種類 (-1=自動, 1=CC-A,2=CC-B)
GS1 DataBar Stacked Omnidirectional Composite	Option1	コンポジットコンポーネントの種類 (-1=自動, 1=CC-A,2=CC-B)
GS1 DataBar Expanded Stacked Composite	Option1	コンポジットコンポーネントの種類 (-1=自動, 1=CC-A,2=CC-B)

Win32 APIリファレンス

バーコード体系	オプション	
GS1 QRコード	Option1	誤り訂正レベル(-1=自動,1=L, 2=M, 3=Q, 4=H)
	Option2	バージョン指定(0=自動,1~40=指定)
	Option3	マスクパターン(0=自動,1~8=指定)
GS1 Aztec Code	Option1	誤り訂正レベル(-1=自動,1~4=指定)
	Option2	サイズ指定(0=自動,1~36=指定)
	Option3	サイズ自動時にCompact Aztecを候補に入れない = 2
GS1 DataMatrix	Option2	サイズ指定(0=自動, 1~30=指定)
	Option3	サイズ自動時に長方形で生成するか 1=正方形でのみ生成する 2=長方形の方がサイズが小さい場合長方形で生成する
連結QRコード	Option1	誤り訂正レベル(-1,0=自動,1=L, 2=M, 3=Q, 4=H)
	Option2	バージョン指定(0=通常QRコード,1~40=指定) 1~40は21×21~177×177のモジュールを指定します。 (型番が1つ増える毎に4モジュール増えます。) 0を指定した場合は連結QRコードではなく通常のQRコードが印字されます。
	Option3	マルチバイト文字分割回避(0=無効 1=有効) 連結QRコードにマルチバイト文字が含まれている際に、マルチバイト文字がQRコード間で分割されてしまうことを回避します。 バーコードリーダーによってはマルチバイト文字が分割されてしまうと文字化けしてしまう為、その際は当オプションを有効にして利用ください。

関数名		NBarcode / NBarcodeA	
引数名	IN/OUT	型	説明
i_prt	I	PWCHAR / PCHAR	出カプリンタ名/ドライバ名
i_fontName	I	PWCHAR / PCHAR	フォント名
o_bmp	O	HDC	デバイスコンテキストハンドル
i_x	I	DWORD	レフト
i_y	I	DWORD	トップ
i_width	I	DWORD	幅
i_height	I	DWORD	高さ
i_dat	I	PBYTE	バーコードデータ
i_size	I	DWORD	データサイズ
戻り値	INT		
・エラー(-値)、正常終了(0)			
処理内容	<p>・プリンタのバーコードフォントに指定されている設定のバーコードを o_bmp 上に描画します。なお、iOS SDKでは2次元バーコードはサポートしていません。1次元バーコードのみ使用できます。</p> <p>第2引数のフォント名にはバーコード設定ファイル(NBarcodeInf.inf)に使用されているフォント名を指定して下さい。(Barcode1~Barcode10)</p> <p>NBarcodeInf.infについては当ガイドの「バーコード設定ファイルについて」を参照して下さい。</p>		
<div style="border: 1px solid black; border-radius: 10px; padding: 5px; width: fit-content; margin-bottom: 10px;">注意</div> <ul style="list-style-type: none"> ・Ver.3.2.0.0以降からはNBarcode2関数を使用してバーコードを描画することを推奨します。今後当関数はサポートを終了する恐れがあります。 ・バーコードデータが大きく、作成されたバーコードがデータサイズを超過した場合にもエラーとならず、超過された部分が削除されたバーコードデータが作成されます。この場合、作成されたバーコードを読み込むことは出来ませんので、データサイズの調整を行い、データが全て入りきるように指定して下さい。 			

関数名 NBarcode / NBarcodeA

処理内容 引数の指定イメージ図

UIImageクラス領域(第3引数・灰色枠部分)



※**赤枠部分**のバーコード部分の幅・高さ・HRI文字の有無などについては
バーコード設定ファイル(NBarcodeInf)にて設定します。

※バーコードの回転設定は赤枠部分が回転します。上記図の白い部分は回転しません
ので、回転後の幅・高さで領域設定して下さい。

関数名		NBarcode2	
引数名	IN/OUT	型	説明
o_bmp	IO	HDC	デバイスコンテキストハンドル(NULL指定可)
i_x	I	DWORD	レフト
i_y	I	DWORD	トップ
io_width	IO	PDWORD	幅
io_height	IO	PDWORD	高さ
i_dat	I	PBYTE	バーコードデータ
i_size	I	INT	データサイズ

戻り値 INT

・エラー(-値)、正常終了(0) ※エラーコード表を参照

処理内容

- ・NBarcodeSettingsの設定を元にバーコードを o_bmp 上に描画します。
- ・当関数実行前にNBarcodeSettingsで生成したいバーコードの設定を行う必要があります。
- ・o_bmpをNULLで渡すことでio_widthとio_heightに生成に必要な幅と高さを返します。
- ・o_bmpにデバイスコンテキストハンドルを渡す際は、io_widthとio_heightにデバイスコンテキストハンドルの幅と高さを渡します。
- ・i_x、i_y、io_width、io_heightの上限値は3776です。

注意

・バーコードデータが大きく、作成されたバーコードがデータサイズを超過した場合にもエラーとならず、超過された部分が削除されたバーコードデータが作成されます。この場合、作成されたバーコードを読み込むことは出来ませんので、データサイズの調整を行い、データが全て入りきるように指定して下さい。

関数名	NBarcode2
処理内容	引数の指定イメージ図

HDC領域(第1引数・灰色枠部分)



※**赤枠部分**のバーコード部分の幅・高さ・HRI文字の有無などについては
NBarcodeSettings関数にて設定します。

※バーコードの回転設定は赤枠部分が回転します。上記図の白い部分は回転しません
ので、回転後の幅・高さで領域設定して下さい。

関数名		NFirmwareDL / NFirmwareDLA	
引数名	IN/OUT	型	説明
i_prt	I	PWCHAR / PCHAR	出力プリンタ名
i_file	I	PWCHAR / PCHAR	fwfファイル名 (NULL指定可)
i_errflg	I	BYTE	エラーチェック 0x00: 無効 (強制送信) 0x01: 有効
io_chksum	IO	PWORD	fwfファイルのチェックサム
o_jobid	O	PINT	プリントジョブID (NULL指定可)
戻り値	INT		
<p>・エラー(-値)、正常終了もしくはチェックサム一致(0) ※エラーコード表を参照 コールバック設定時は非同期処理が開始した時点で正常終了(0)を返す</p>			
処理内容	<p>非同期可能関数</p> <ul style="list-style-type: none"> 指定されたFWFファイルをプリンタに送信します。 またFWFファイルをNULLで指定した場合には、プリンタからチェックサムを取得し、指定されたチェックサムと比較します。 コールバック関数設定時、当関数はファームウェア書換処理を非同期で処理を行いません。 非同期処理開始前のエラーは当関数の戻り値で返しますが、戻り値が0(正常終了)で返った場合、実行結果はコールバック関数で取得してください。 ただし、FWFファイルをNULLで指定してチェックサムの確認を行った場合は同期で処理を行う為当関数の戻り値がチェックサム比較結果となります。 <p>※本関数を使用する前に必ずNGetStatusを事前に実施してプリンタが正常動作していることを確認してから続けて本関数を実施してください。</p>		

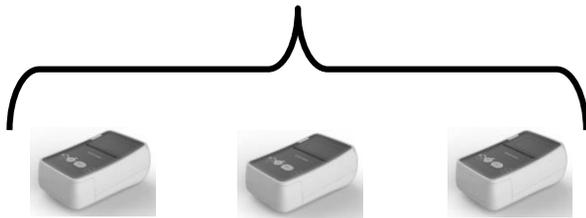
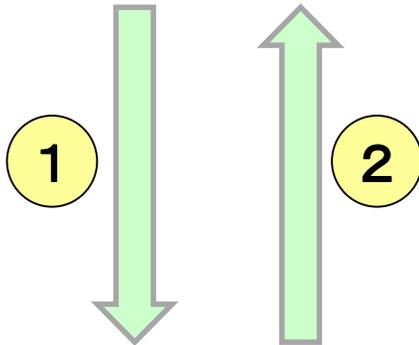
関数名		NInitializeNetwork		
引数名	IN/OUT	型	説明	
i_flg	I	INT	UDP受信サービス起動/停止フラグ 0: 停止 1: 起動	
戻り値	INT			
・エラー(-値)、正常終了(0) ※エラーコード表を参照				
処理内容				
<p>・UDP受信サービスを起動 / 停止します。 初期状態は停止しています。</p> <p>TCP/UDP通信時には処理開始前に当関数でUDP受信サービスを起動して下さい。</p> <p>TCP/UDP通信を利用する時には、利用開始の一番最初に実行してください。 利用しない場合は、当関数は実行しないで下さい。</p> <div style="border: 1px solid black; border-radius: 10px; padding: 5px; width: fit-content; margin: 10px 0;">注意</div> <p>・複数アプリケーションでUDP受信スレッドを使用する場合、UDP受信スレッドの起動/停止は全てのアプリケーションに適用されます。</p> <p>・5秒間経ってもUDP受信サービスの起動/停止を確認できなかった場合タイムアウトします。 NServiceDrvが正しく動作していない恐れがあるためサービス管理ツールよりNServiceDrvが実行されていることを確認してください。</p> <p>・LAN接続のドライバでプリンタ設定画面を開くとNInitializeNetworkが実行されUDP受信サービスが自動で起動します。</p>				

関数名		NScanPrinters		
引数名	IN/OUT	型	説明	
i_waitmsec	I	long	列挙応答・データ作成までの待受ミリ秒数	
戻り値	INT			
・エラー(-値)、正常終了(検出台数) ※エラーコード表を参照 コールバック設定時は非同期処理が開始した時点で正常終了(0)を返す				
処理内容	非同期可能関数 ※TCP/UDPプリンタのみ対応			
<ul style="list-style-type: none">・TCP/UDPプリンタ検出用関数です。TCP/UDPプリンタを使用しない場合は、当関数を実行する必要はありません。 当関数を実行することでプリンタ列挙要求をUDPでブロードキャスト送信します。 <p>引数 i_waitmsec で指定されたミリ秒の間、プリンタ列挙応答の受信を行い、応答データからプリンタ情報の作成を行います。 通常 i_waitmsec は 3000 を指定してください(3秒) (プリンタ情報の取得は別途、NEnumPrinters関数を使用します)</p> <ul style="list-style-type: none">・コールバック関数設定時、当関数は非同期で処理を行いません。 非同期処理開始前のエラーは当関数の戻り値で返しますが、戻り値が0(正常終了)で返った場合、実行結果はコールバック関数で取得してください。				
<div style="border: 1px solid black; border-radius: 10px; padding: 5px; display: inline-block;">注意</div> <p>i_waitmsec に 0 を指定された場合にはプリンタ列挙要求を行わず、プリンタ情報作成済みのプリンタ台数を戻り値で返します。</p>				

TCP/UDP (WLAN) 使用時のプリンタ名取得



PC/タブレット等



192.168.0.1

192.168.0.2

192.168.0.3

※NScanPrinters関数はプリンタ情報を作る際に呼び出します。

- ・新規にTCP/UDPプリンタが追加された
- ・プリンタのIPアドレスが変更になった

等の場合に実行する必要があります。

NEnumPrinters関数を使用する度に必ずしも実行する必要はありません。

1 NScanPrinters呼出
(プリンタ列挙要求:UDP・ブロードキャスト送信)

2 TCP/UDP接続されている各プリンタからプリンタ列挙応答:UDPをSDK内のプリンタ情報として受信します。

3 NEnumPrinters呼出
(受信データを元にTCP/UDPプリンタ情報を取得します)

Win32 APIリファレンス

関数名	NTCPortLock / NTCPortLockA		
引数名	IN/OUT	型	説明
i_prt	I	PWCHAR / PCHAR	出力プリンタ名
i_type	I	BYTE	設定種別
戻り値	INT		
・エラー(-値)、正常終了(0) ※エラーコード表を参照 コールバック設定時は非同期処理が開始した時点で正常終了(0)を返す			
処理内容	非同期可能関数		
<ul style="list-style-type: none">・TCP通信のロック/アンロックを行います。 ロックしたユーザのみTCP通信でデータ送信が行えます。 他ユーザは接続はできませんが送信不可となります。 <p>設定種別</p> <ul style="list-style-type: none">ロック : 0x01アンロック : 0x00 <ul style="list-style-type: none">・コールバック関数設定時、当関数は非同期で処理を行いません。 非同期処理開始前のエラーは当関数の戻り値で返しますが、戻り値が0(正常終了)で返った場合、実行結果はコールバック関数で取得してください。			

関数名	NBufferClear / NBufferClearA		
引数名	IN/OUT	型	説明
i_prt	I	PWCHAR / PCHAR	出力プリンタ名
戻り値	INT		
・エラー(-値)、正常終了(0) ※エラーコード表を参照 コールバック設定時は非同期処理が開始した時点で正常終了(0)を返す			
処理内容	非同期可能関数		
・受信バッファのクリアを行います(全バッファ) TCP/UDP接続以外ではエラーを返します。			
・コールバック関数設定時、当関数は非同期で処理を行いません。 非同期処理開始前のエラーは当関数の戻り値で返しますが、戻り値が0(正常終了)で 返った場合、実行結果はコールバック関数で取得してください。			

Win32 APIリファレンス

関数名		NBlockSendSetting / NBlockSendSettingA	
引数名	IN/OUT	型	説明
i_prt	I	PWCHAR / PCHAR	出力プリンタ名
i_type	I	BYTE	設定種別
戻り値	INT		
・エラー(-値)、正常終了(0) ※エラーコード表を参照 コールバック設定時は非同期処理が開始した時点で正常終了(0)を返す			
処理内容	非同期可能関数		
・送信データをTCP/UDPモジュールのバッファへ溜め込み、一括送信が行えます。 当関数は一括送信のON/OFFを切り替えることができます。 (TCP/UDP接続以外ではエラーを返します)			
設定種別			
ON :0x01			
OFF :0x00			
・コールバック関数設定時、当関数は非同期で処理を行いません。 非同期処理開始前のエラーは当関数の戻り値で返しますが、戻り値が0(正常終了)で 返った場合、実行結果はコールバック関数で取得してください。			

エラーコード一覧(1/2)

SDKの使用しているエラーコードの一覧になります。主に関数の戻り値として使用されていますが、SDK内部処理でのみ使用しているもの・他OSとの互換のため存在するものがあります。各関数の戻り値は「[関数戻り値一覧](#)」を参照して下さい。

N_SUCCESS	0	正常終了
N_ERR_HANDLE	-1	ハンドルエラー
N_ERR_PRTOPEN	-2	プリンターオープンエラー
N_ERR_SEND_ERROR	-3	送信エラー発生中
N_ERR_OFFLINE	-5	オフライン
N_ERR_PRTCLOSE	-6	プリンタークローズエラー
N_ERR_FILEOPEN	-10	ファイルオープンエラー
N_ERR_NOT_MAPPING	-11	拡張情報取得エラー(MapViewOfFile失敗)
N_ERR_NOT_OPEN_MAPFILE	-12	マップファイルオープンエラー
N_ERR_PRTOUTPUT	-13	プリンター出力エラー
N_ERR_NONE_PRTLIST	-21	使用できるプリンターが存在しません
N_ERR_NOHANDLE	-22	プリンターがオープンされていません
N_ERR_LACKRESOURCE	-31	リソース不足
N_ERR_NOTSUPPORTED	-40	サポートされていない関数です
N_ERR_LOADFROMFILE	-50	画像ファイルの読み込みに失敗しました
N_ERR_IMAGESIZE	-51	イメージサイズ不正
N_ERR_RESETPRINTER	-60	リセット失敗
N_ERR_ACCESSDENIED	-61	アクセス拒否
N_ERR_RESETTIMEOUT	-62	リセット処理タイムアウト
N_ERR_STARTDOC	-70	StartDoc関数エラー
N_ERR_DOCNOTSTARTED	-71	ドキュメント開始状態でない
N_ERR_ALREADYSTARTDOC	-72	既にドキュメント開始状態となっている
N_ERR_FWFFILE	-80	fwfファイルエラー
N_ERR_FWF_CHECKSUM	-81	引数で入力したチェックサムと、 プリンターから取得したチェックサムが一致しない
N_ERR_FWDL_TIMEOUT	-82	ファームウェアダウンロードタイムアウト (印字開始コマンドのチェックがタイムアウト)
N_ERR_FWCHK_TIMEOUT	-83	ファームウェアチェックサム確認タイムアウト
N_ERR_FWDL_FOUNDERERROR	-84	ファームウェアダウンロード時のステータス確認でエラーを検出
N_ERR_ARGUMENT	-90	引数不正値エラー
N_ERR_ARGUMENT_01	-91	第1引数エラー
N_ERR_ARGUMENT_02	-92	第2引数エラー
N_ERR_ARGUMENT_03	-93	第3引数エラー
N_ERR_ARGUMENT_04	-94	第4引数エラー
N_ERR_ARGUMENT_05	-95	第5引数エラー

エラーコード一覧(2/2)

N_ERR_ARGUMENT_06	-96	第6引数エラー
N_ERR_ARGUMENT_07	-97	第7引数エラー
N_ERR_ARGUMENT_08	-98	第8引数エラー
N_ERR_ARGUMENT_09	-99	第9引数エラー
N_ERR_SOCKET_RECV	-108	受信エラー
N_ERR_UDPTHREAD_STARTED	-111	UDPスレッドは既に起動しています
N_ERR_UDPTHREAD_STOPPED	-112	UDPスレッドが停止しています / 起動できませんでした
N_ERR_UDPTHREAD_STOP	-113	UDPスレッドを停止できませんでした
N_ERR_PRTINFO_READ	-131	プリンター情報ファイルの読み込みに失敗しました
N_ERR_PRTINFO_WRITE	-132	プリンター情報ファイルの書き込みに失敗しました
N_ERR_PRTNAME_ALLOC	-133	プリンター名の割り当てに失敗しました
N_ERR_PRTRENAME_BEFORE	-134	変更前のプリンター名が存在しません
N_ERR_PRTRENAME_AFTER	-135	変更後のプリンター名が既に使用されています
N_ERR_PRTINFO_ILLEGAL	-137	プリンター情報ファイルが不正です
N_ERR_PRTINFO_DELETE	-138	プリンター名の削除に失敗しました
N_ERR_PRTINFO_NOTFOUND	-139	プリンター名が存在しませんでした
N_ERR_DEVICE_NOT_SUPPORT	-150	接続種別がサポートされていません
N_ERR_CREATEBCDDATA	-162	バーコードデータの生成に失敗しました
N_ERR_BARCODE_DATA_LONG	-163	バーコード文字列が長すぎます
N_ERR_BARCODE_DATA_INVALID	-164	バーコードに対応していない文字列が含まれています
N_ERR_BARCODE_OPTION	-165	オプションの組み合わせが正しくありません
N_ERR_BARCODE_BUFFER	-166	バーコード生成用のバッファが足りませんでした
N_ERR_MNT_HEADER	-200	メンテナンス応答のヘッダーが不正です
N_ERR_FLAGSQR	-201	メンテナンス応答(Flags)のQRフラグが不正です
N_ERR_FLAGSFORMAT	-202	メンテナンス応答(Flags)でフォーマットエラーを検出しました
N_ERR_FLAGSBUSY	-203	メンテナンス応答(Flags)でビジーを検出しました
N_ERR_FLAGSUNDEFINED	-204	メンテナンス応答(Flags)で未実装を検出しました
N_ERR_FLAGSREJECT	-205	メンテナンス応答(Flags)で拒否を検出しました
N_ERR_FLAGSOTHER	-206	メンテナンス応答(Flags)でその他エラーを検出しました
N_ERR_MNT_QID	-207	メンテナンス応答のクエリIDが不正です
N_ERR_MNT_QR	-208	メンテナンス応答の予備フラグが不正です
N_ERR_MNT_QPARAM	-209	メンテナンス応答のクエリパラメータが不正です
N_ERR_MNT_OTHER	-210	メンテナンス応答その他のエラー
N_ERR_NO_LIBRARY	-900	DLLが読み込めません
N_ERR_OTHER	-999	その他エラー
N_WRN_PRTALREADYOPEN	10	プリンターが既にオープンされています
N_WRN_NOTEXISTDOC	11	ドキュメントが存在しませんでした

API別エラーコード一覧(1/5)

関数名	定義名	定義値	備考
NSetCallback	戻り値なし	—	
NEnumPrinters	N_SUCCESS	0	関数戻り値
	N_ERR_NONE_PRTLIST	-21	関数戻り値
	N_ERR_ARGUMENT_02	-92	関数戻り値
	N_ERR_PRTNAME_ALLOC	-133	関数戻り値
NDeletePrinter	N_SUCCESS	0	関数戻り値
	N_WRN_PRTALREADYOPEN	10	関数戻り値
	N_ERR_PRTINFO_READ	-131	関数戻り値
	N_ERR_PRTINFO_NOTFOUND	-139	関数戻り値
	N_ERR_DEVICE_NOTSUPPORT	-150	関数戻り値
NRenamePrinter	N_SUCCESS	0	関数戻り値
	N_WRN_PRTALREADYOPEN	10	関数戻り値
	N_ERR_ARGUMENT_01	-91	関数戻り値
	N_ERR_ARGUMENT_02	-92	関数戻り値
	N_ERR_PRTRENAME_BEFORE	-134	関数戻り値
	N_ERR_PRTRENAME_AFTER	-135	関数戻り値
NGetPrinterInf	N_SUCCESS	0	関数戻り値
	N_ERR_ARGUMENT_01	-91	関数戻り値
	N_ERR_ARGUMENT_03	-93	関数戻り値
	N_ERR_PRTINFO_ILLEGAL	-137	関数戻り値
	N_ERR_PRTINFO_NOTFOUND	-139	関数戻り値
NGetPrinterFromID	プリンター名サイズ	+値	関数戻り値
	N_ERR_ARGUMENT_01	-91	関数戻り値
	N_ERR_PRTINFO_NOTFOUND	-139	関数戻り値
NAutoOpen	Ver.3.0.0.0にて廃止	FALSE	使用しないで下さい
NOpenPrinter	N_SUCCESS	0	関数戻り値・コールバック
	N_WRN_PRTALREADYOPEN	10	関数戻り値・コールバック
	N_ERR_HANDLE	-1	関数戻り値・コールバック
	N_ERR_PRTOPEN	-2	関数戻り値・コールバック
	N_ERR_FILEOPEN	-10	関数戻り値・コールバック
	N_ERR_NONE_PRTLIST	-21	関数戻り値・コールバック
	N_ERR_ARGUMENT_01	-91	関数戻り値
	N_ERR_UDPTHREADSTOPPED	-112	関数戻り値・コールバック
	N_ERR_PRTINFO_NOTFOUND	-139	関数戻り値・コールバック
N_ERR_DEVICE_NOTSUPPORT	-150	関数戻り値・コールバック	
NClosePrinter	N_SUCCESS	0	関数戻り値
	N_ERR_PRTCLOSE	-6	関数戻り値
	N_ERR_NONE_PRTLIST	-21	関数戻り値
	N_ERR_NOHANDLE	-22	関数戻り値
	N_ERR_ARGUMENT_01	-91	関数戻り値
	N_ERR_DEVICE_NOTSUPPORT	-150	関数戻り値
NClosePrinters	N_SUCCESS	0	関数戻り値
	N_ERR_NONE_PRTLIST	-21	関数戻り値

API別エラーコード一覧(2/5)

関数名	定義名	定義値	備考
NPrint	N_SUCCESS	0	関数戻り値
	N_ERR_HANDLE	-1	関数戻り値
	N_ERR_FILEOPEN	-10	関数戻り値
	N_ERR_PRTOUTPUT	-13	関数戻り値
	N_ERR_NONE_PRTLIST	-21	関数戻り値
	N_ERR_NOHANDLE	-22	関数戻り値
	N_ERR_LACKRESOURCE	-31	関数戻り値
	N_ERR_LOADFROMFILE	-50	関数戻り値
	N_ERR_IMAGESIZE	-51	関数戻り値
	N_ERR_ARGUMENT	-90	関数戻り値
	N_ERR_ARGUMENT_01	-91	関数戻り値
	N_ERR_ARGUMENT_02	-92	関数戻り値
	N_ERR_ARGUMENT_03	-93	関数戻り値
NDPrint	N_SUCCESS	0	関数戻り値
	N_ERR_PRTOUTPUT	-13	関数戻り値
	N_ERR_NONE_PRTLIST	-21	関数戻り値
	N_ERR_NOHANDLE	-22	関数戻り値
	N_ERR_ARGUMENT_01	-91	関数戻り値
	N_ERR_ARGUMENT_02	-92	関数戻り値
NImagePrint	N_SUCCESS	0	関数戻り値
	N_ERR_PRTOUTPUT	-13	関数戻り値
	N_ERR_NONE_PRTLIST	-21	関数戻り値
	N_ERR_NOHANDLE	-22	関数戻り値
	N_ERR_LACKRESOURCE	-31	関数戻り値
	N_ERR_IMAGESIZE	-51	関数戻り値
	N_ERR_ARGUMENT_01	-91	関数戻り値
	N_ERR_ARGUMENT_02	-92	関数戻り値
	N_ERR_ARGUMENT_03	-93	関数戻り値
	N_ERR_ARGUMENT_04	-94	関数戻り値
N_ERR_ARGUMENT_05	-95	関数戻り値	
NImagePrintF	N_SUCCESS	0	関数戻り値
	N_ERR_PRTOUTPUT	-13	関数戻り値
	N_ERR_NONE_PRTLIST	-21	関数戻り値
	N_ERR_NOHANDLE	-22	関数戻り値
	N_ERR_LACKRESOURCE	-31	関数戻り値
	N_ERR_LOADFROMFILE	-50	関数戻り値
	N_ERR_IMAGESIZE	-51	関数戻り値
	N_ERR_ARGUMENT_01	-91	関数戻り値
	N_ERR_ARGUMENT_02	-92	関数戻り値
	N_ERR_ARGUMENT_03	-93	関数戻り値
NGetStatus	N_SUCCESS	0	関数戻り値
	N_ERR_SEND_ERROR	-3	関数戻り値
	N_ERR_OFFLINE	-5	関数戻り値
	N_ERR_NOT_MAPPING	-11	関数戻り値
	N_ERR_NOT_OPEN_MAPFILE	-12	関数戻り値

API別エラーコード一覧(3/5)

関数名	定義名	定義値	備考
NGetStatus	N_ERR_PRTOUTPUT	-13	関数戻り値
	N_ERR_LACKRESOURCE	-31	関数戻り値
	N_ERR_ARGUMENT_01	-91	関数戻り値
	N_ERR_ARGUMENT_02	-92	関数戻り値
NGetInformation	N_SUCCESS	0	関数戻り値
	N_ERR_NOT_MAPPING	-11	関数戻り値
	N_ERR_NOT_OPEN_MAPFILE	-12	関数戻り値
	N_ERR_PRTOUTPUT	-13	関数戻り値
	N_ERR_NONE_PRTLIST	-21	関数戻り値
	N_ERR_NOHANDLE	-22	関数戻り値
	N_ERR_ARGUMENT_01	-91	関数戻り値
	N_ERR_ARGUMENT_02	-92	関数戻り値
NResetPrinter	N_SUCCESS	0	関数戻り値・コールバック
	N_ERR_NONE_PRTLIST	-21	関数戻り値
	N_ERR_LACKRESOURCE	-31	関数戻り値・コールバック
	N_ERR_RESETPRINTER	-60	関数戻り値・コールバック
	N_ERR_RESETTIMEOUT	-62	関数戻り値・コールバック
	N_ERR_ARGUMENT_01	-91	関数戻り値
	N_ERR_DEVICE_NOTSUPPORT	-150	関数戻り値・コールバック
	N_ERR_NO_LIBRARY	-900	関数戻り値・コールバック
NStartDoc	N_SUCCESS	0	関数戻り値
	N_ERR_PRTOUTPUT	-13	関数戻り値
	N_ERR_NONE_PRTLIST	-21	関数戻り値
	N_ERR_ALREADYSTARTDOC	-72	関数戻り値
	N_ERR_ARGUMENT_01	-91	関数戻り値
NEndDoc	N_SUCCESS	0	関数戻り値
	N_ERR_NONE_PRTLIST	-21	関数戻り値
	N_ERR_LACKRESOURCE	-31	関数戻り値
	N_ERR_DOCNOTSTARTED	-71	関数戻り値
	N_ERR_ARGUMENT_01	-91	関数戻り値
NCancelDoc	N_SUCCESS	0	関数戻り値
	N_ERR_NONE_PRTLIST	-21	関数戻り値
	N_ERR_DOCNOTSTARTED	-71	関数戻り値
	N_ERR_ARGUMENT_01	-91	関数戻り値
NEnumDoc	リスト保存サイズ	+ 値	関数戻り値
	N_ERR_NONE_PRTLIST	-21	関数戻り値
	N_ERR_ARGUMENT_01	-91	関数戻り値
NDeleteDoc	N_SUCCESS	0	関数戻り値
	N_WRN_NOTEXISTDOC	11	関数戻り値
	N_ERR_NONE_PRTLIST	-21	関数戻り値
	N_ERR_ARGUMENT_01	-91	関数戻り値
	N_ERR_DEVICE_NOTSUPPORT	-150	関数戻り値

API別エラーコード一覧(4/5)

関数名	定義名	定義値	備考
NBarcodeSettings	N_SUCCESS	0	関数戻り値
	N_ERR_ARGUMENT	-90	関数戻り値(第10引数エラー)
	N_ERR_ARGUMENT_01	-91	関数戻り値
	N_ERR_ARGUMENT_02	-92	関数戻り値
NBarcodeSettings	N_ERR_ARGUMENT_07	-97	関数戻り値
	N_ERR_ARGUMENT_08	-98	関数戻り値
	N_ERR_ARGUMENT_09	-99	関数戻り値
	N_ERR_BARCODE_OPTION	-165	関数戻り値
NBarcode	N_SUCCESS	0	関数戻り値
	N_ERR_NONE_PRTLIST	-21	関数戻り値
	N_ERR_LACKRESOURCE	-31	関数戻り値
	N_ERR_ARGUMENT_01	-91	関数戻り値
	N_ERR_ARGUMENT_02	-92	関数戻り値
	N_ERR_ARGUMENT_03	-93	関数戻り値
	N_ERR_ARGUMENT_04	-94	関数戻り値
	N_ERR_ARGUMENT_05	-95	関数戻り値
	N_ERR_ARGUMENT_06	-96	関数戻り値
	N_ERR_ARGUMENT_07	-97	関数戻り値
	N_ERR_ARGUMENT_08	-98	関数戻り値
N_ERR_ARGUMENT_09	-99	関数戻り値	
NBarcode2	N_SUCCESS	0	関数戻り値
	N_ERR_LACKRESOURCE	-31	関数戻り値
	N_ERR_ARGUMENT_01	-91	関数戻り値
	N_ERR_ARGUMENT_04	-94	関数戻り値
	N_ERR_ARGUMENT_05	-95	関数戻り値
	N_ERR_ARGUMENT_06	-96	関数戻り値
	N_ERR_ARGUMENT_07	-97	関数戻り値
	N_ERR_CREATEBCDDATA	-162	関数戻り値
	N_ERR_BARCODE_DATA_LONG	-163	関数戻り値
	N_ERR_BARCODE_DATA_INVALID	-164	関数戻り値
	N_ERR_BARCODE_OPTION	-165	関数戻り値
	N_ERR_BARCODE_BUFFER	-166	関数戻り値
	N_ERR_NO_LIBRARY	-900	関数戻り値
NFirmwareDL (ファームウェアチェック) (ファームウェアダウンロード)	N_SUCCESS	0	関数戻り値・コールバック
	N_ERR_OFFLINE	-5	関数戻り値・コールバック
	N_ERR_NONE_PRTLIST	-21	関数戻り値
	N_ERR_ARGUMENT_01	-91	関数戻り値
	N_ERR_ARGUMENT_03	-93	関数戻り値
	N_ERR_FWF_CHECKSUM	-81	関数戻り値
	N_ERR_FWCHK_TIMEOUT	-83	関数戻り値
	N_ERR_SEND_ERROR	-3	関数戻り値
	N_ERR_PRTOUTPUT	-13	関数戻り値・コールバック
	N_ERR_NOHANDLE	-22	関数戻り値・コールバック
	N_ERR_LACKRESOURCE	-31	関数戻り値・コールバック

API別エラーコード一覧(5/5)

関数名	定義名	定義値	備考
NFirmwareDL	N_ERR_DOCNOTSTARTED	-71	関数戻り値・コールバック
	N_ERR_ALREADYSTARTDOC	-72	関数戻り値
	N_ERR_FWFFILE	-80	関数戻り値・コールバック
	N_ERR_FWDL_TIMEOUT	-82	関数戻り値・コールバック
	N_ERR_FOUNDERORROR	-84	関数戻り値
NInitializeNetwork	N_SUCCESS	0	関数戻り値
	N_ERR_ARGUMENT_01	-91	関数戻り値
	N_ERR_UDPTHREADSTARTED	-111	関数戻り値
	N_ERR_UDPTHREADSTOPPED	-112	関数戻り値
	N_ERR_UDPTHREADSTOP	-113	関数戻り値
NScanPrinters	N_SUCCESS	0	関数戻り値・コールバック
	N_ERR_PRTOUTPUT	-13	関数戻り値・コールバック
	N_ERR_ARGUMENT_01	-91	関数戻り値
	N_ERR_UDPTHREADSTOPPED	-112	関数戻り値・コールバック
	N_ERR_PRTNAME_ALLOC	-133	関数戻り値・コールバック
	N_ERR_PRTINFO_NOTFOUND	-139	関数戻り値・コールバック
	LANプリンタ数	+ 値	関数戻り値・コールバック
NtcpPortLock	N_SUCCESS	0	関数戻り値・コールバック
	N_ERR_PRTOUTPUT	-13	関数戻り値・コールバック
	N_ERR_NONE_PRTLST	-21	関数戻り値・コールバック
	N_ERR_NOHANDLE	-22	関数戻り値・コールバック
	N_ERR_ARGUMENT_01	-91	関数戻り値・コールバック
	N_ERR_ARGUMENT_02	-92	関数戻り値・コールバック
	N_ERR_UDPTHREADSTOPPED	-112	関数戻り値・コールバック
	N_ERR_DEVICE_NOTSUPPORT	-150	関数戻り値・コールバック
NBufferClear	N_SUCCESS	0	関数戻り値・コールバック
	N_ERR_PRTOUTPUT	-13	関数戻り値・コールバック
	N_ERR_NONE_PRTLST	-21	関数戻り値・コールバック
	N_ERR_NOHANDLE	-22	関数戻り値・コールバック
	N_ERR_LACKRESOURCE	-31	関数戻り値・コールバック
	N_ERR_ARGUMENT_01	-91	関数戻り値・コールバック
	N_ERR_ARGUMENT_02	-92	関数戻り値・コールバック
	N_ERR_UDPTHREADSTOPPED	-112	関数戻り値・コールバック
	N_ERR_DEVICE_NOTSUPPORT	-150	関数戻り値・コールバック
NBlockSendSetting	N_SUCCESS	0	関数戻り値・コールバック
	N_ERR_PRTOUTPUT	-13	関数戻り値・コールバック
	N_ERR_NONE_PRTLST	-21	関数戻り値・コールバック
	N_ERR_NOHANDLE	-22	関数戻り値・コールバック
	N_ERR_ARGUMENT_01	-91	関数戻り値・コールバック
	N_ERR_ARGUMENT_02	-92	関数戻り値・コールバック
	N_ERR_UDPTHREADSTOPPED	-112	関数戻り値・コールバック
	N_ERR_DEVICE_NOTSUPPORT	-150	関数戻り値・コールバック

※NPrintA等、名称末尾にAが付く関数は上記に加えてN_ERR_LACKRESOURCE(-31)が戻り値となる

クラス説明

ネームスペース : NPrinterCLib

クラス名 : NClassLib

種別	名称	備考
メソッド	NEnumPrinters	Win32 APIラッパ
メソッド	NGetPrinterFromID	Win32 APIラッパ
メソッド	NDeletePrinter	Win32 APIラッパ
メソッド	NRenamePrinter	Win32 APIラッパ
メソッド	NGetPrinterInf	Win32 APIラッパ
メソッド	NAutoOpen	Win32 APIラッパ
メソッド	NOpenPrinter	Win32 APIラッパ
メソッド	NClosePrinter	Win32 APIラッパ
メソッド	NClosePrinters	Win32 APIラッパ
メソッド	NPrint	Win32 APIラッパ
メソッド	NDPrint	Win32 APIラッパ
メソッド	NImagePrint	Win32 APIラッパ
メソッド	NImagePrintF	Win32 APIラッパ
メソッド	NGetStatus	Win32 APIラッパ
メソッド	NGetInformation	Win32 APIラッパ
メソッド	NResetPrinter	Win32 APIラッパ
メソッド	NStartDoc	Win32 APIラッパ
メソッド	NEndDoc	Win32 APIラッパ
メソッド	NCancelDoc	Win32 APIラッパ
メソッド	NEnumDoc	Win32 APIラッパ
メソッド	NDeleteDoc	Win32 APIラッパ
メソッド	NBarcodeSettings	Win32 APIラッパ
メソッド	NBarcode	Win32 APIラッパ
メソッド	NBarcode2	Win32 APIラッパ

クラス説明

ネームスペース : NPrinterCLib

クラス名 : NClassLib

種 別	名 称	備 考
メソッド	NFirmwareDL	Win32 APIラッパ
メソッド	NScanPrinters	Win32 APIラッパ
メソッド	NTCPPortLock	Win32 APIラッパ
メソッド	NBufferClear	Win32 APIラッパ
メソッド	NBlockSendSetting	Win32 APIラッパ
メソッド	NSetCallback	Win32 APIラッパ
メソッド	NInitializeNetwork	Win32 APIラッパ

関数名	NEnumPrinters		
引数名	IN/OUT	型	説明
o_printers	O	[MarshalAs(UnmanagedType.LPWStr)] StringBuilder	プリンタ名 (NULL指定可)
o_size	O	out long	o_printersのバイトサイズ
戻り値	int		
処理内容			
備考			

戻り値、処理内容については「Win32 APIリファレンス」参照

関数名	NGetPrinterFromID		
引数名	IN/OUT	型	説明
i_ID	I	[MarshalAs(UnmanagedType.LPWStr)] string	BluetoothシリアルID
o_printer	O	[MarshalAs(UnmanagedType.LPWStr)] StringBuilder	プリンタ名 (NULL指定可)
戻り値	int		
処理内容			
備考			

戻り値、処理内容については「Win32 APIリファレンス」参照

関数名	NDeletePrinter		
引数名	IN/OUT	型	説明
o_printers	0	[MarshalAs(UnmanagedType.LPWStr)] string	プリンタ名
戻り値	int		
処理内容			
備考			

戻り値、処理内容については「Win32 APIリファレンス」参照

関数名	NRenamePrinter		
引数名	IN/OUT	型	説明
i_beforeprt	I	[MarshalAs(UnmanagedType.LPWStr)] string	変更前のプリンタ名
i_afterprt	I	[MarshalAs(UnmanagedType.LPWStr)] string	変更後のプリンタ名
戻り値	int		
処理内容			
備考			

戻り値、処理内容については「Win32 APIリファレンス」参照

関数名	NGetPrinterInf		
引数名	IN/OUT	型	説明
i_prt	I	[MarshalAs(UnmanagedType.LPWStr)] string	プリンタ名
o_ports	O	MarshalAs(UnmanagedType.LPWStr)] StringBuilder	ポート情報
o_size	O	out long	o_portsのバイトサイズ
戻り値	int		
処理内容			
備考			

戻り値、処理内容については「Win32 APIリファレンス」参照

関数名	NAutoOpen		
引数名	IN/OUT	型	説明
i_flg	I	int	自動プリンタオープンフラグ 0 : 自動オープンしない(手動オープン) 1 : 自動オープンする その他 : 状態取得のみ(設定は行わない)
戻り値	bool		
処理内容			
備考			
<p>Ver.3.0.0.0より当関数は機能廃止となりました。 戻り値は必ずfalseとなります。</p>			

戻り値、処理内容については「Win32 APIリファレンス」参照

関数名	NOpenPrinter		
引数名	IN/OUT	型	説明
i_prt	I	[MarshalAs(UnmanagedType.LPWStr)] string	オープンするプリンタ名 受信スレッド起動フラグ コールバック関数デリゲート
i_statusFlg	I	byte	
i_callback	I	NCALLBACK	
戻り値	int		
処理内容			
備考			

戻り値、処理内容については「Win32 APIリファレンス」参照

関数名	NClosePrinter		
引数名	IN/OUT	型	説明
i_prt	I	[MarshalAs(UnmanagedType.LPWStr)] string	プリンタ名
戻り値	int		
処理内容			
備考			

戻り値、処理内容については「Win32 APIリファレンス」参照

関数名	NClosePrinters		
引数名	IN/OUT	型	説明
戻り値	int		
処理内容			
備考			

戻り値、処理内容については「Win32 APIリファレンス」参照

関数名	NPrint		
引数名	IN/OUT	型	説明
iprt	I	[MarshalAs(UnmanagedType.LPWStr)] string	出力プリンタ名
i_dat	I	byte[]	送信データ(16進文字列)
i_size	I	uint	出力バイト数
io_jobid	IO	out int	プリントジョブID(NULL指定可)
戻り値	int		
処理内容			
備考			

戻り値、処理内容については「Win32 APIリファレンス」参照

関数名	NDPrint		
引数名	IN/OUT	型	説明
i_prt	I	[MarshalAs(UnmanagedType.LPWStr)] string	出力プリンタ名
i_dat	I	byte[]	送信データ(16進文字列)
i_size	I	uint	出力バイト数
io_jobid	IO	out int	プリントジョブID(NULL指定可)
戻り値	int		
処理内容			
備考			

戻り値、処理内容については「Win32 APIリファレンス」参照

関数名	NImagePrint		
引数名	IN/OUT	型	説明
i_prt	I	[MarshalAs(~LPWStr)] string	出力プリンタ名
i_bmp	I	IntPtr	デバイスコンテキストハンドル
i_width	I	int	幅
i_height	I	int	高さ
i_putType	I	byte	出力方式 0x00: ラスター形式ライン単位 0x01: ラスター形式ブロック単位 0x02: ラスター形式ブロック単位階調表現 0x10: ビットイメージ形式
io_jobid	IO	out int	プリントジョブID(NULL指定可)
戻り値	int		
処理内容			
備考			

戻り値、処理内容については「Win32 APIリファレンス」参照

関数名	NImagePrintF		
引数名	IN/OUT	型	説明
i_prt	I	[MarshalAs(~LPWStr)] string	出力プリンタ名
i_bmp	I	[MarshalAs(~LPWStr)] string	BMPファイル名
i_putType	I	byte	出力方式 0x00:ラスター形式ライン単位 0x01:ラスター形式ブロック単位 0x02:ラスター形式ブロック単位階調表現 0x10:ビットイメージ形式
io_jobid	IO	out int	プリントジョブID (NULL指定可)
戻り値	int		
処理内容			
備考			

戻り値、処理内容については「Win32 APIリファレンス」参照

関数名	NGetStatus		
引数名	IN/OUT	型	説明
i_prt	I	[MarshalAs(UnmanagedType.LPWStr)] string	プリンタ名
o_status	O	out long	ステータス
戻り値	int		
処理内容			
備考	<ul style="list-style-type: none"> PrinterNameプロパティ設定後使用のこと 		

戻り値、処理内容については「Win32 APIリファレンス」参照

関数名	NGetInformation		
引数名	IN/OUT	型	説明
i_prt	I	[MarshalAs(UnmanagedType.LPWStr)] string	プリンタ名
i_id	I	byte	種別ID
o_dat	O	byte[]	拡張情報格納エリア
o_time	O	out long	更新フラグ(システムを起動した後の経過時間)(NULL指定可)
戻り値	int		
処理内容			
備考			

戻り値、処理内容については「Win32 APIリファレンス」参照

関数名	NResetPrinter		
引数名	IN/OUT	型	説明
i_prt	I	[MarshalAs(UnmanagedType.LPWStr)] string	プリンタ名
戻り値	int		
処理内容			
備考			

戻り値、処理内容については「Win32 APIリファレンス」参照

関数名	NStartDoc		
引数名	IN/OUT	型	説明
i_prt	I	[MarshalAs(UnmanagedType.LPWStr)] string	プリンタ名
o_jobid	O	out int	ジョブID (NULL指定可)
戻り値	int		
処理内容			
備考			

戻り値、処理内容については「Win32 APIリファレンス」参照

関数名	NEndDoc		
引数名	IN/OUT	型	説明
i_prt	I	[MarshalAs(UnmanagedType.LPWStr)] string	プリンタ名
戻り値	int		
処理内容			
備考			

戻り値、処理内容については「Win32 APIリファレンス」参照

関数名	NCancelDoc		
引数名	IN/OUT	型	説明
i_prt	I	[MarshalAs(UnmanagedType.LPWStr)] string	プリンタ名
戻り値	int		
処理内容			
備考			

戻り値、処理内容については「Win32 APIリファレンス」参照

関数名	NEnumDoc		
引数名	IN/OUT	型	説明
i_prt	I	[MarshalAs(UnmanagedType.LPWStr)] string	プリンタ名
o_docnolist	O	[MarshalAs(UnmanagedType.LPWStr)] StringBuilder	ドキュメント番号リスト(NULL指定可)
戻り値	int		
処理内容			
備考			

戻り値、処理内容については「Win32 APIリファレンス」参照

関数名	NDeleteDoc		
引数名	IN/OUT	型	説明
i_prt i_docno	I	[MarshalAs(UnmanagedType.LPWStr)] string int	プリンタ名 ドキュメント番号
戻り値	int		
処理内容			
備考			

戻り値、処理内容については「Win32 APIリファレンス」参照

関数名	NBarcodeSettings		
引数名	IN/OUT	型	説明
i_id i_width i_height i_option1 i_option2 i_option3 i_option4 i_hri i_spin i_code	I	uint uint uint char char char char uint uint uint	バーコード種別 データ幅の倍率 1モジュールの高さ オプション1 オプション2 オプション3 オプション4 HRI文字の位置 回転 文字コード種別
戻り値	int		
処理内容			
備考			

戻り値、処理内容については「Win32 APIリファレンス」参照

関数名	NBarcode		
引数名	IN/OUT	型	説明
i_prt	I	string	プリンタ名/ドライバ名
i_fontName	I	string	フォント名
i_bmp	IO	IntPrt	デバイスコンテキストハンドル
i_x	I	uint	レフト
i_y	I	uint	トップ
i_width	I	uint	幅
i_height	I	uint	高さ
i_dat	I	byte[]	バーコードデータ
i_size	I	uint	データサイズ
戻り値	int		
処理内容			
備考	戻り値、処理内容については「Win32 APIリファレンス」参照		

関数名	NBarcode2		
引数名	IN/OUT	型	説明
i_bmp	IO	IntPrt	デバイスコンテキストハンドル
i_x	I	uint	レフト
i_y	I	uint	トップ
io_width	IO	out uint	幅
io_height	IO	out uint	高さ
i_dat	I	byte[]	バーコードデータ
i_size	I	int	データサイズ
戻り値	int		
処理内容			
備考	戻り値、処理内容については「Win32 APIリファレンス」参照		

関数名	NFirmwareDL		
引数名	IN/OUT	型	説明
i_prt	I	[MarshalAs(UnmanagedType.LPStr)] string	出力プリンタ名 fwfファイル名 (NULL指定可) エラーチェック 0x00:無効(強制送信) 0x01:有効
i_file	I	[MarshalAs(UnmanagedType.LPStr)] string	
i_errflg	I	byte	
io_chksum	O	out short	fwfファイルのチェックサム プリントジョブID (NULL指定可)
io_jobid	O	out int	
戻り値	int		
処理内容			
備考			

戻り値、処理内容については「Win32 APIリファレンス」参照

関数名	NScanPrinters		
引数名	IN/OUT	型	説明
i_waitmsec	I	uint	プリンタ列挙要求から列挙応答・データ作成までの待機ミリ秒数
戻り値	int		
処理内容			
備考			

戻り値、処理内容については「Win32 APIリファレンス」参照

関数名	NTCPortLock		
引数名	IN/OUT	型	説明
i_prt i_type	I I	[MarshalAs(UnmanagedType.LPWStr)] string byte	プリンタ名 接続種別
戻り値	int		
処理内容			
備考			

戻り値、処理内容については「Win32 APIリファレンス」参照

関数名	NBufferClear		
引数名	IN/OUT	型	説明
i_prt	I	[MarshalAs(UnmanagedType.LPWStr)] string	プリンタ名
戻り値	int		
処理内容			
備考			

戻り値、処理内容については「Win32 APIリファレンス」参照

関数名	NBlockSendSetting		
引数名	IN/OUT	型	説明
i_prt i_type	I I	[MarshalAs(UnmanagedType.LPWStr)] string byte	プリンタ名 設定種別
戻り値	int		
処理内容			
備考			

戻り値、処理内容については「Win32 APIリファレンス」参照

関数名	NSetCallback		
引数名	IN/OUT	型	説明
i_callback	I	NClassLib.NCALLBACK	コールバック関数ポインタ
戻り値	int		
処理内容			
備考			

戻り値、処理内容については「Win32 APIリファレンス」参照

関数名	NInitializeNetwork		
引数名	IN/OUT	型	説明
i_flg	I	int	設定種別
戻り値	int		
処理内容			
備考			

戻り値、処理内容については「Win32 APIリファレンス」参照