



Brother Windows DLL

ソフトウェア開発者マニュアル

著作権

© 2020 Brother Industries, Ltd. All rights reserved.

本文書の情報は予告無く変更されることがあります。本文書に記載されているソフトウェアは使用許諾契約に従って提供されています。ソフトウェアは、これらの契約条項に従ってのみ使用またはコピーできます。本文書のいかなる部分も、ブラザー工業株式会社の書面による事前の許可なしに、いかなる形式または手段によっても複製することはできません。

商標

ブラザー製品および関連資料等に記載されている社名及び商品名はそれぞれ各社の商標または登録商標です。

重要なお知らせ

本資料は、お客様が直接ラベルプリンター（対応機種については、「[5.1 対応モデル](#)」をご参照ください）を制御される場合に必要な情報を提供するものです。一部の機能は機種によってご利用いただけない場合があります。

お客様が以下の内容にご同意いただいた場合のみ、本資料のご利用が可能です。もしご同意いただけない場合は、お客様は本資料をご利用いただけません。

無保証

- a. 対応モデルおよび本資料のバージョンアップや修正等はブラザーが任意で行うものとします。
- b. ブラザーは本資料に関し、明示または黙示であるかを問わず、瑕疵がないこと、特定の目的に適合することを含め、その他いかなる保証も行いません。
- c. ブラザーは本資料および本資料に基づきお客様が作成したプログラムに起因して発生した直接的または間接的損害について、お客様に対し、いかなる責任も負わないものとします。

お問合せ

モバイルSDKに関する詳細はこちらをご参照ください
<https://support.brother.co.jp/j/s/es/dev/ja/index.html?navi=offall>

目次

1 概要	1
1.1 ラベルを印刷する	1
1.2 プリンターのコマンドを使用して印刷する	1
1.3 FBPLテンプレートを使用して印刷する	2
1.4 画像を印刷する	3
1.5 RFIDタグに書き込む	4
2 API	5
2.1 openport	5
2.2 opennetport	5
2.3 closeport	6
2.4 setup	7
2.5 clearbuffer	7
2.6 barcode	8
2.7 printerfont	10
2.8 sendcommand	12
2.9 printlabel	12
2.10 downloadpcx	13
2.11 formfeed	13
2.12 nobackfeed	13
2.13 windowsfont	14
2.14 about	15
2.15 rfidWrite	16
2.16 rfidRead	17
2.17 rfidGetReadData	18
2.18 rfidGetReadDatavb6	18
2.19 usbportqueryprinter	18
2.20 netportqueryprinter	19
3 各インターフェイスで対応しているAPI	20
3.1 TD-4420TN/TD-4520TN	20
3.2 TD-4650TNWB/TD-4750TNWB/TD-4750TNWBR	20
3.3 RJ-2035B/RJ-3035B	21
3.4 TJ-4021TN/TJ-4121TN/ TJ-4121TNR	22
3.5 TJ-4420TN/ TJ-4520TN/ TJ-4620TN	23
4 サンプルコード	24
4.1 例 (1)	24
4.2 例 (2)	24
5 About	26
5.1 対応モデル	26

1 概要

1.1 ラベルを印刷する

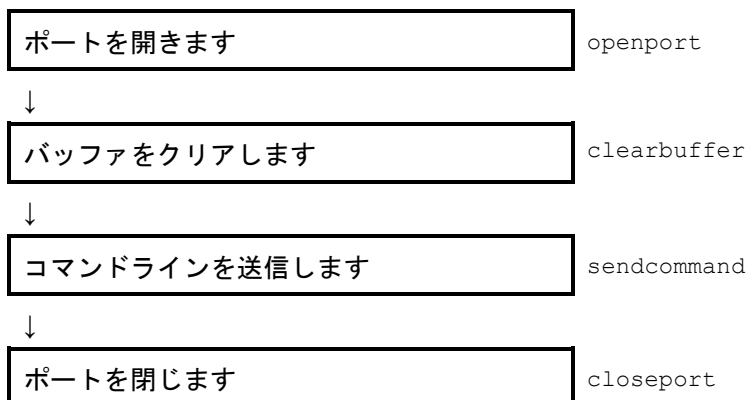
- 例：



1.2 プリンターのコマンドを使用して印刷する

プリンターのコマンド（例：FBPL、ZPL）を送信するには、`sendcommand`関数を使います。

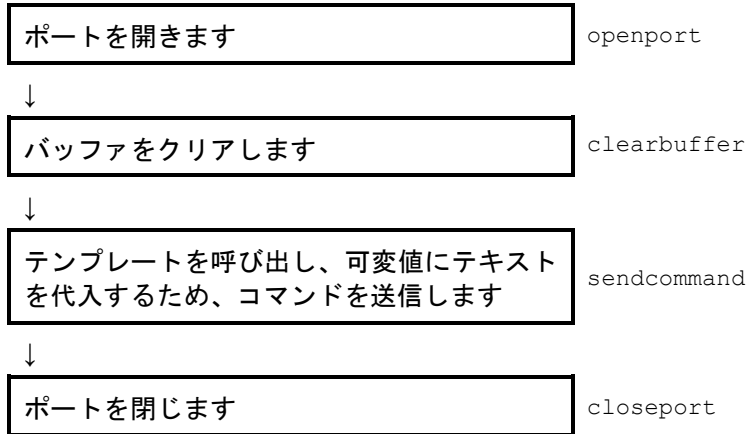
- 例：



1.3 FBPLテンプレートを使用して印刷する

1. FBPLコマンドを使用して、テンプレート（BAS）を作成します。
2. BPM（Brother Printer Management Tool）を使用して、プリンターのFLASHメモリにテンプレートを転送します。
3. `sendcommand`関数を使用して、テンプレートを読み出し、入力したい可変値を送信します。

- 例：



1.4 画像を印刷する

PCX/BMPファイル（例：ロゴ）を呼び出して印刷する場合は、`downloadpcx`関数を使用して、あらかじめプリンターのFLASHメモリにファイルを転送しておきます。その後、FBPLコマンドを送信して画像を呼び出します。



- ビットマップを印刷する場合は、1-bit（モノクロ）または 8-bit（256色）のビットマップ形式のファイルを転送してください。
- **PUTPCX**または**PUTBMP**コマンドを使用して、保存した画像をモノクロまたはグレースケールで印刷します。感熱印刷の場合は、モノクロとグレースケール印刷に対応しています。熱転写印刷の場合は、モノクロ印刷のみに対応しています。詳細は、FBPLコマンドレファレンスの**PUTPCX**または**PUTBMP**コマンドをご参照ください。

1.5 RFIDタグに書き込む

RFIDタグに書き込むには `rfidWrite` 関数を使用します。

- 例：



- 対応モデル
 - TD-4750TNWBR
 - TJ-4121TNR

2 API

2.1 openport

Windowsプリンタースプールを開始します。

宣言

```
int openport (String port);
```

パラメーター

パラメーター	型	必須	説明
port	string	必須	1) プリンタードライバー名 (例: Brother TD-4420TN)
			2) ネットワークプリンターの場合はUNCパスとプリンター名 (例: "\\server\TD-4420TN")
			3) セントロニクスインタフェース LPT1 - LPT4 (例: "LPT1")
			4) USBインタフェース (例: "USB")

戻り値

integer; 成功時: 1、それ以外の場合0

2.2 opennetport

IPアドレスを指定して、Windowsプリンタースプールを開始します。

宣言

```
int opennetport (String IP address, int printer port);
```

パラメーター

パラメーター	型	必須	説明
IP address	string	必須	プリンターIPアドレス (例: "192.168.0.1")
Printer port	integer	必須	プリンター通信ポート (例: 9100)

戻り値

integer; 成功時: 1、それ以外の場合0

2.3 closeport

Windows プリンタースプールを閉じます。

宣言

```
int closeport ();
```

パラメーター

なし

戻り値

integer; 成功時 : 1、それ以外の場合 0

2.4 setup

プリンターの印刷設定をします。

宣言

```
int setup (String width, String height, String speed, String density, String sensorType, String gap, String distance);
```

パラメーター

パラメーター	型	必須	説明
width	string	必須	ラベル幅 (mm)
height	string	必須	ラベル高さ (mm)
speed	string	必須	印字速度 (インチ/秒) 1 / 1.5 / 2 / 3 / 4 / 6 / 8 / 10 / 12 / 13 / 14 設定できる印字速度は、FBPLコマンドレファレンスのSPEEDコマンドをご参照ください。
density	string	必須	印字濃度 0-15 : 0最小濃度、15最大濃度
sensorType	string	必須	用紙センサーの種類 0 : ギャップセンサー、1 : ブラックマークセンサー
gap	string	必須	ラベル間のギャップまたはブラックマークの高さ (mm)
distance	string	必須	ギャップ・ブラックマークのシフト距離(mm); 通常はこのパラメーターを0に設定します。

戻り値

integer; 成功時 : 1、それ以外の場合0



無定長用紙（長尺テープ）をご使用の場合は、sensorType、gap、distanceのパラメーターは0を指定します。

2.5 clearbuffer

バッファをクリアします。

宣言

```
int clearbuffer();
```

パラメーター

なし

戻り値

integer; 成功時 : 1、それ以外の場合0

2.6 barcode

プリンター内蔵のバーコードフォーマットを使用します。

宣言

```
int barcode(String xDir, String yDir, String barcode, String height,
String interpretation, String rotation, String narrowRatio, String
wideRatio, String content);
```

パラメーター

パラメーター	型	必須	説明
xDir	string	必須	X方向のバーコード開始位置（ドット） (203 dpi: 1 mm=8ドット、300 dpi: 1 mm=12ドット、600 dpi: 1 mm=24ドット)
yDir	string	必須	Y方向のバーコード開始位置（ドット） (203 dpi: 1 mm=8ドット、300 dpi: 1 mm=12ドット、600 dpi: 1 mm=24ドット)
barcode	string	必須	バーコードの種類 詳細は、 バーコードの種類 をご参照ください。
height	string	必須	バーコードの高さ（ドット）
interpretation	string	必須	目視可能文字（ヒューマンリーダブル） 0: 目視可能文字を印刷しない 1: 目視可能文字を印刷する
rotation	string	必須	回転角度 0 : 回転なし 90 : 時計まわりに90度回転 180 : 時計まわりに180度回転 270 : 時計まわりに270度回転
narrowRatio	string	必須	ナローバー比率 詳細は、 <i>FBPL</i> コマンドレファレンスの BARCODE コマンドをご参照ください。
wideRatio	string	必須	ワイドバー比率 詳細は、 <i>FBPL</i> コマンドレファレンスの BARCODE コマンドをご参照ください。
content	string	必須	バーコードコンテンツ

戻り値

integer; 成功時 : 1、それ以外の場合 0

バーコードの種類

一覧からbarcodeの種類を選択してください。

バーコード種類	説明
128	Code 128 コードサブセット自動切替
128M	Code 128 コードサブセット手動切替
EAN128	EAN-128 コードサブセット自動切替
EAN128M	EAN-128 (GS1-128) コードサブセット手動切替
25	Interleaved 2-of-5
25C	Interleaved 2-of-5 チェックディジット付き
25S	Standard 2-of-5
25I	Industrial 2-of-5
39	Code 39 スタンダードとフルアスキーモードの自動切替
39C	Code 39 チェックディジット付き
93	Code 93
EAN13	EAN-13
EAN13+2	2桁付加EAN-13
EAN13+5	5桁付加EAN-13
EAN8	EAN-8
EAN8+2	2桁付加EAN-8
EAN8+5	5桁付加EAN-8
CODA	Codabar (NW-7)
POST	POSTNET

バーコード種類	説明
UPCA	UPC-A
UPCA+2	2桁付加UPC-A
UPA+5	5桁付加UPC-A
UPCE	UPC-E
UPCE+2	2桁付加UPC-E
UPE+5	5桁付加UPC-E
MSI	MSI
MSIC	MSI チェックディジット付き
PLESSEY	Plessey
CPOST	Datalogic 2-of-5 (China post)
ITF14	ITF-14
EAN14	EAN-14
11	Code 11 (USD-8)
TELEPEN	Telepen
TELEPENN	Telepen 数字のみ
PLANET	Planet
CODE49	Code 49
DPI	Deutsche Post Identcode
DPL	Deutsche Post Leitcode
LOGMARS	Code 39 特別仕様



その他のバーコードを印刷するには、`sendcommand`関数を使用してFBPLコマンドを送信してください。

例 : `sendcommand("QRCODE 10,10,H,4,A,0,\"ABCabc123\"\\r\\n")`

2.7 printerfont

プリンター内蔵のフォントを使用して、テキストオブジェクトを定義します。

宣言

```
int printerfont(String xDir, String yDir, String fontType, String rotation,
String xRate, String yRate, String content);
```

パラメーター

パラメーター	型	必須	説明
xDir	string	必須	X方向のテキスト開始位置（ドット） (203 dpi: 1 mm=8ドット、300 dpi: 1 mm=12ドット、600 dpi: 1 mm=24ドット)
yDir	string	必須	Y方向のテキスト開始位置（ドット） (203 dpi: 1 mm=8ドット、300 dpi: 1 mm=12ドット、600 dpi: 1 mm=24ドット)
fontType	string	必須	フォントファイル名 ファイル名はすべて大文字にしてください（例：FILENAME.TTF）
rotation	string	必須	回転角度 0 : 回転なし 90 : 時計まわりに90度回転 180 : 時計まわりに180度回転 270 : 時計まわりに270度回転
xRate	string	必須	X方向のテキストの倍率（範囲：1-8）
yRate	string	必須	Y方向のテキストの倍率（範囲：1-8）
Content	string	必須	印刷するテキストコンテンツ

戻り値

integer; 成功時：1、それ以外の場合0



内蔵フォントを使用するには、次の一覧からfontTypeを選択してください。

フォント名	説明
0	CG Triumvirate Bold Condense (Monotype Imaging®) フォント幅と高さは伸縮可能です
1	8 x 12 ドット固定ピッチフォント
2	12 x 20 ドット固定ピッチフォント
3	16 x 24ドット固定ピッチフォント
4	24 x 32 ドット固定ピッチフォント
5	32 x 48 ドット固定ピッチフォント
6	14 x 19 ドット固定ピッチフォント OCR-B
7	21 x 27 ドット固定ピッチフォント OCR-B
8	14 x25 ドット固定ピッチフォント OCR-A
ROMAN.TTF	CG Triumvirate Bold Condensed (Monotype Imaging®) フォント幅と高さの比率は固定です
1.EFT	EPL2 font 1
2.EFT	EPL2 font 2
3.EFT	EPL2 font 3
4.EFT	EPL2 font 4
5.EFT	EPL2 font 5
A.FNT	ZPL2 font A

B.FNT	ZPL2 font B
D.FNT	ZPL2 font D
E8.FNT	ZPL2 font E8
F.FNT	ZPL2 font F
G.FNT	ZPL2 font G
H8.FNT	ZPL2 font H8
GS.FNT	ZPL2 font GS
K1.SFT	16×16 ドット漢字
K2.SFT	24×24 ドット漢字
K3.SFT	22×22 ドット漢字
K4.SFT	32×32 ドット漢字
K8.SFT	16×16 ドット半角・全角漢字
K9.SFT	24×24 ドット半角・全角漢字
KA.SFT	22×22 ドット半角・全角漢字
KB.SFT	32×32 ドット半角・全角漢字



- ダウンロードフォントを使用するには、Brother Printer Management Tool (BPM) から使用したいフォントをプリンターに転送してください：
 1. パソコンとプリンターを接続し、プリンターの電源を入れます。
 2. BPMを起動し、「ファイルマネージャー」をクリックします。
 3. 「ファイル形式」を選択します。「ブラウズ」をクリックして、フォントファイルを選択し、「開く」をクリックします。
 4. 「メモリー機器」のドロップダウンリストから「FLASH」を選択します。
 5. 「ダウンロード」をクリックして、ダウンロードを開始します。
- ファイル名はすべて大文字にしてダウンロードしてください（例：ipaexg.ttf→IPAEXG.TTF）
- BPMはWindowsのツールです。Bluetooth®接続で使用できません。USBまたはWi-Fi®を使用してパソコンとプリンターを接続してください。



- 文字化けしたり、一部の文字が印刷されない場合は、プリンターの文字コード設定が印刷データの文字コードと一致していることを確認します。使用するフォントが設定した文字コードに対応していない場合は、文字コード設定を変更するか、フォントを変更して もう一度お試しください。
- プリンターの文字コード設定を変更する場合は、FBPLコマンドレファレンスのCODEPAGEコマンドをご参照ください。
- 内蔵フォント「K1.SFT / K2.SFT / K3.SFT / K4.SFT / K8.SFT / K9.SFT / KA.SFT / KB.SFT」は、シフト JIS に対応しています。使用する場合は、CODEPAGE コマンドで「932 (Shift-JIS)」を指定ください。

2.8 sendcommand

プリンターにコマンドを送信します。

宣言

```
int sendcommand(String command);
```

パラメーター

パラメーター	型	必須	説明
command	string	必須	内蔵プリンターコマンド（例：FBPLコマンド） FBPLコマンドの詳細は、 <i>FBPLコマンドレファレンス</i> をご参照ください。

例： `sendcommand("PRINT 1\r\n")` `sendcommand("CLS\r\nPRINT\r\n")`

戻り値

integer; 成功時：1、それ以外の場合0



お使いのプリンターファームウェアが対応しているFBPL以外のコマンドを送信することも可能です。`sendcommand`関数を使用してFBPL以外のコマンドを送信する場合、この関数と組み合わせて送信できるのは、`openport`・`closeport`関数のみです。

2.9 printlabel

ラベルを印刷します。

宣言

```
int printlabel(String set, String copy);
```

パラメーター

パラメーター	型	必須	説明
set	string	必須	ラベルセット数
copy	string	必須	コピー枚数

戻り値

integer; 成功時：1、それ以外の場合0

2.10 downloadpcx

モノクロPCX／BMPグラフィックファイルをプリンターに転送します。

宣言

```
int downloadpcx(String fileName, String name);
```

パラメーター

パラメーター	型	必須	説明
fileName	string	必須	拡張子を含むファイル名
name	string	必須	ファイル名（プリンターのFLASHメモリに保存するファイル名） ファイル名はすべて大文字にしてください（例：FILENAME.PCX、 FILENAME.BMP）

戻り値

integer; 成功時：1、それ以外の場合0



詳細は、[1.4 画像を印刷する](#) をご参照ください。

2.11 formfeed

用紙を1枚搬送します。`setup`関数を呼んだあとに使用してください。

宣言

```
int formfeed();
```

パラメーター

なし

戻り値

integer; 成功時：1、それ以外の場合0

2.12 nobackfeed

バックフィードを無効にします。

宣言

```
int nobackfeed();
```

パラメーター

なし

戻り値

integer; 成功時：1、それ以外の場合0

2.13 windowsfont

Windowsフォントを使用して、テキストオブジェクトを定義します。

宣言

```
int windowsfont(int xDir, int yDir, int height, int rotation, int
fontStyle, int underline, String fileType, String text);
```

パラメーター

パラメーター	型	必須	説明
xDir	integer	必須	X方向のテキスト開始位置（ドット） (203 dpi: 1 mm=8ドット、300 dpi: 1 mm=12ドット、600 dpi: 1 mm=24ドット)
yDir	integer	必須	Y方向のテキスト開始位置（ドット） (203 dpi: 1 mm=8ドット、300 dpi: 1 mm=12ドット、600 dpi: 1 mm=24ドット)
height	integer	必須	フォントの高さ（ドット）
rotation	integer	必須	回転角度 0 : 回転なし 90 : 反時計まわりに90度回転 180 : 反時計まわりに180度回転 270 : 反時計まわりに270度回転
fontStyle	integer	必須	フォントのスタイル 0 : 装飾なし 1 : 斜体 2 : 太字 3 : 太字と斜体
underline	integer	必須	下線 0 : 下線なし 1 : 下線あり
fileType	string	必須	フォント名
text	string	必須	印刷するテキストコンテンツ

戻り値

integer; 成功時 : 1、それ以外の場合0

2.14 about

画面にDLLバージョンを表示します。

宣言

```
void about();
```

パラメーター

なし

戻り値

integer; 成功時 : 1

2.15 rfidWrite

RFIDタグにデータを書き込みます。

宣言

```
int rfidWrite(String lock, String format, String blockNumber, String dataSize, String memoryBank, String data);
```

パラメーター

パラメーター	型	必須	説明
lock	string	必須	データブロックに書き込んで、上書きされないようにロックします。 0 : ロックせずに書き込みます 1 - FFFFFFFF (16進数) : 書き込んでロックします
format	string	必須	データフィールドの形式 A = ASCII H = Hex (16進数)
blockNumber	string	必須	0を指定します
dataSize	string	必須	10進数で1～nのデータサイズを書き込みます
memoryBank	string	必須	メモリ領域 EPC – EPC領域（最大496 bits） USR – ユーザー領域（最大8 kilobits） ACS – アクセスコード KIL – キルコード PC – Protocol Control (PC) 領域
data	string	必須	書き込むデータ

戻り値

integer; 成功時：1、それ以外の場合0



- 「戻り値 integer; 成功時：1」はプリンターにコマンドを送ることに成功したかどうかを表します。rfidWritが成功したかどうかではありません。
- RFIDに関するAPIは「rfidWrite → rfidRead → rfidGetReadData」の順に記載してください。
- 対応モデル：TD-4750TNWBR、TJ-4121TNR
- PC領域への書き込みは、次のファームウェアバージョンからご使用いただけます。

機種	ファームウェアバージョン
TD-4750TNWBR	V1.04.S21 以降
TJ-4121TNR	B2.12.S70 以降

2.16 rfidRead

RFIDタグのデータを読み取り、`printlnlabel`関数と呼んだあとに読み取った値を返します。

宣言

```
int rfidRead(String unlock, String format, String blockNumber, String dataSize, String memoryBank);
```

パラメーター

パラメーター	型	必須	説明
unlock	string	必須	あとで上書きできるようにロックを解除します。 0 : ロック解除せずに読み取ります 1 - FFFFFFFF (16進数) : 読み取り、ロックを解除します
format	string	必須	データフィールドの形式 A = ASCII H = Hex (16進数)
blockNumber	string	必須	0を指定します
dataSize	string	必須	10進数で1～nのデータサイズを読み取ります
memoryBank	string	必須	メモリ領域 EPC – EPC領域 (最大496 bits) TID – TID領域 USR – ユーザー領域 (最大8 kilobits) ACS – アクセスコード KIL – キルコード PC – Protocol Control (PC) 領域

戻り値

integer; 成功時 : 1、それ以外の場合0



- 対応モデル : TD-4750TNWBR、TJ-4121TNR
- 読み取ったデータを表示するためには異なるAPI「rfidGetReadData」を使用する必要があります。詳細は本マニュアルの該当セッションをご参照ください。
- RFIDに関するAPIは「rfidWrite → rfidRead → rfidGetReadData」の順に記載してください。

2.17 rfidGetReadData

`rfidRead`関数で読み取ったRFIDタグのデータを返します。
この関数は、`printlabel`関数を呼んだ直後に使用してください。
下記Interfaceに対応しています。

- openport: USB接続のみ
- opennetport: 有線LAN、無線LAN

宣言

```
string rfidGetReadData();
```

パラメーター

なし

戻り値

string; タグのデータ



- 対応モデル：TD-4750TNWBR、TJ-4121TNR
- RFIDに関するAPIは「rfidWrite → rfidRead → rfidGetReadData」の順に記載してください。

2.18 rfidGetReadDatavb6

VBA向けの関数で、`rfidRead`関数で読み取ったRFIDタグのデータを返します。
この関数は、`printlabel`関数を呼んだ直後に使用してください。

宣言

```
string rfidGetReadDatavb6();
```

パラメーター

なし

戻り値

string; タグのデータ



対応モデル：TD-4750TNWBR、TJ-4121TNR

2.19 usbportqueryprinter

プリンターのステータスを取得します（USB接続のopenport("USB")使用時のみ）
詳細は、FBPLコマンドレファレンスの<ESC>!?コマンドをご参照ください。

宣言

```
string usbportqueryprinter();
```

パラメーター

なし

戻り値

string; 失敗時：-1、それ以外の場合：ステータスコード

ステータスコード (HEX)	プリンタステータス
00	通常
01	ヘッドオープン
02	用紙詰まり
03	用紙詰まりとヘッドオープン
04	用紙切れ
05	用紙切れとヘッドオープン
08	リボン切れ
09	リボン切れとヘッドオープン
0A	リボン切れと用紙詰まり
0B	リボン切れ、用紙詰まりとヘッドオープン
0C	リボン切れと用紙切れ
0D	リボン切れ、用紙切れとヘッドオープン
10	一時停止
20	印刷
80	その他のエラー

2.20 netportqueryprinter

プリンタのステータスを取得しますopennetport（有線LANもしくは無線LAN経由）でIPアドレス指定した場合に使用できます。

詳細は、FBPLコマンドレファレンスの<ESC>!?コマンドをご参照ください。

宣言

```
string netportqueryprinter();
```

パラメーター

なし

戻り値

string; 失敗時：-1、それ以外の場合：ステータスコード

ステータスコード (HEX)	プリンタステータス
00	通常
01	ヘッドオープン
02	用紙詰まり
03	用紙詰まりとヘッドオープン
04	用紙切れ
05	用紙切れとヘッドオープン
08	リボン切れ
09	リボン切れとヘッドオープン
0A	リボン切れと用紙詰まり
0B	リボン切れ、用紙詰まりとヘッドオープン
0C	リボン切れと用紙切れ
0D	リボン切れ、用紙切れとヘッドオープン
10	一時停止
20	印刷
80	その他のエラー

3 各インターフェイスで対応しているAPI

3.1 TD-4420TN/TD-4520TN

API	COM	Ethernet	Wi-Fi	USB	Driver
1. openport()	OK	N/A	N/A	OK	OK
2. opennetport(a, b)	N/A	OK	N/A	N/A	N/A
3. closeport()	OK	OK	N/A	OK	OK
4. setup(a,b,c,d,e,f,g)	N/A	OK	N/A	OK	OK
5. clearbuffer()	N/A	OK	N/A	OK	OK
6. barcode(a,b,c,d,e,f,g,h,i)	N/A	OK	N/A	OK	OK
7. printerfont(a,b,c,d,e,f,g)	N/A	OK	N/A	OK	OK
8. sendcommand(command)	N/A	OK	N/A	OK	OK
9. printlabel(a,b)	N/A	OK	N/A	OK	OK
10. downloadpcx(a,b)	N/A	OK	N/A	OK	OK
11. formfeed()	N/A	OK	N/A	OK	OK
12. nobackfeed()	N/A	OK	N/A	OK	OK
13. windowsfont(a,b,c,d,e,f,g,h)	N/A	OK	N/A	OK	OK
14. about()	OK	OK	OK	OK	OK
15. rfidWrite(a,b,c,d,e,f)	N/A	N/A	N/A	N/A	N/A
16. rfidRead(a,b,c,d,e)	N/A	N/A	N/A	N/A	N/A
17. rfidGetReadData()	N/A	N/A	N/A	N/A	N/A
18. printerstatus()	OK	N/A	N/A	N/A	N/A
19. usbportqueryprinter()	N/A	N/A	N/A	OK	N/A
20. netportqueryprinter()	N/A	OK	N/A	N/A	N/A

3.2 TD-4650TNWB/TD-4750TNWB/TD-4750TNWBR

API	COM	Ethernet	Wi-Fi	USB	Driver
1. openport()	OK	N/A	N/A	OK	OK
2. opennetport(a, b)	N/A	OK	OK	N/A	N/A
3. closeport()	OK	OK	OK	OK	OK
4. setup(a,b,c,d,e,f,g)	N/A	OK	OK	OK	OK
5. clearbuffer()	N/A	OK	OK	OK	OK
6. barcode(a,b,c,d,e,f,g,h,i)	N/A	OK	OK	OK	OK
7. printerfont(a,b,c,d,e,f,g)	N/A	OK	OK	OK	OK
8. sendcommand(command)	N/A	OK	OK	OK	OK
9. printlabel(a,b)	N/A	OK	OK	OK	OK
10. downloadpcx(a,b)	N/A	OK	OK	OK	OK
11. formfeed()	N/A	OK	OK	OK	OK
12. nobackfeed()	N/A	OK	OK	OK	OK
13. windowsfont(a,b,c,d,e,f,g,h)	N/A	OK	OK	OK	OK
14. about()	OK	OK	OK	OK	OK
15. rfidWrite(a,b,c,d,e,f)	N/A	OK	OK	OK	OK
16. rfidRead(a,b,c,d,e)	N/A	OK	OK	OK	OK
17. rfidGetReadData()	N/A	OK	OK	OK	N/A
18. printerstatus()	OK	N/A	N/A	N/A	N/A
19. usbportqueryprinter()	N/A	N/A	N/A	OK	N/A
20. netportqueryprinter()	N/A	OK	OK	N/A	N/A

3.3 RJ-2035B/RJ-3035B

API	COM	Ethernet	Wi-Fi	USB	Driver
1. openport()	OK	N/A	N/A	OK	OK
2. opennetport(a, b)	N/A	N/A	OK	N/A	N/A
3. closeport()	OK	N/A	OK	OK	OK
4. setup(a,b,c,d,e,f,g)	N/A	N/A	OK	OK	OK
5. clearbuffer()	N/A	N/A	OK	OK	OK
6. barcode(a,b,c,d,e,f,g,h,i)	N/A	N/A	OK	OK	OK
7. printerfont(a,b,c,d,e,f,g)	N/A	N/A	OK	OK	OK
8. sendcommand(command)	N/A	N/A	OK	OK	OK
9. printlabel(a,b)	N/A	N/A	OK	OK	OK
10. downloadpcx(a,b)	N/A	N/A	OK	OK	OK
11. formfeed()	N/A	N/A	OK	OK	OK
12. nobackfeed()	N/A	N/A	OK	OK	OK
13. windowsfont(a,b,c,d,e,f,g,h)	N/A	N/A	OK	OK	OK
14. about()	OK	OK	OK	OK	OK
15. rfidWrite(a,b,c,d,e,f)	N/A	N/A	N/A	N/A	N/A
16. rfidRead(a,b,c,d,e)	N/A	N/A	N/A	N/A	N/A
17. rfidGetReadData()	N/A	N/A	N/A	N/A	N/A
18. printerstatus()	OK	N/A	N/A	N/A	N/A
19. usbportqueryprinter()	N/A	N/A	N/A	OK	N/A
20. netportqueryprinter()	N/A	N/A	OK	N/A	N/A

3.4 TJ-4021TN/TJ-4121TN/ TJ-4121TNR

API	COM	Ethernet	Wi-Fi	USB	Driver
1. openport()	OK	N/A	N/A	OK	OK
2. opennetport(a, b)	N/A	OK	OK	N/A	N/A
3. closeport()	OK	OK	OK	OK	OK
4. setup(a,b,c,d,e,f,g)	N/A	OK	OK	OK	OK
5. clearbuffer()	N/A	OK	OK	OK	OK
6. barcode(a,b,c,d,e,f,g,h,i)	N/A	OK	OK	OK	OK
7. printerfont(a,b,c,d,e,f,g)	N/A	OK	OK	OK	OK
8. sendcommand(command)	N/A	OK	OK	OK	OK
9. printlabel(a,b)	N/A	OK	OK	OK	OK
10. downloadpcx(a,b)	N/A	OK	OK	OK	OK
11. formfeed()	N/A	OK	OK	OK	OK
12. nobackfeed()	N/A	OK	OK	OK	OK
13. windowsfont(a,b,c,d,e,f,g,h)	N/A	OK	OK	OK	OK
14. about()	OK	OK	OK	OK	OK
15. rfidWrite(a,b,c,d,e,f)	N/A	OK	OK	OK	OK
16. rfidRead(a,b,c,d,e)	N/A	OK	OK	OK	OK
17. rfidGetReadData()	N/A	OK	OK	OK	N/A
18. printerstatus()	OK	N/A	N/A	N/A	N/A
19. usbportqueryprinter()	N/A	N/A	N/A	OK	N/A
20. netportqueryprinter()	N/A	OK	OK	N/A	N/A

3.5 TJ-4420TN/ TJ-4520TN/ TJ-4620TN

API	COM	Ethernet	Wi-Fi	USB	Driver
1. openport()	OK	N/A	N/A	OK	OK
2. opennetport(a, b)	N/A	OK	OK	N/A	N/A
3. closeport()	OK	OK	OK	OK	OK
4. setup(a,b,c,d,e,f,g)	N/A	OK	OK	OK	OK
5. clearbuffer()	N/A	OK	OK	OK	OK
6. barcode(a,b,c,d,e,f,g,h,i)	N/A	OK	OK	OK	OK
7. printerfont(a,b,c,d,e,f,g)	N/A	OK	OK	OK	OK
8. sendcommand(command)	N/A	OK	OK	OK	OK
9. printlabel(a,b)	N/A	OK	OK	OK	OK
10. downloadpcx(a,b)	N/A	OK	OK	OK	OK
11. formfeed()	N/A	OK	OK	OK	OK
12. nobackfeed()	N/A	OK	OK	OK	OK
13. windowsfont(a,b,c,d,e,f,g,h)	N/A	OK	OK	OK	OK
14. about()	OK	OK	OK	OK	OK
15. rfidWrite(a,b,c,d,e,f)	N/A	N/A	N/A	N/A	N/A
16. rfidRead(a,b,c,d,e)	N/A	N/A	N/A	N/A	N/A
17. rfidGetReadData()	N/A	N/A	N/A	N/A	N/A
18. printerstatus()	OK	N/A	N/A	N/A	N/A
19. usbportqueryprinter()	N/A	N/A	N/A	OK	N/A
20. netportqueryprinter()	N/A	OK	OK	N/A	N/A

4 サンプルコード

4.1 例 (1)

```

BROLIB_DLL.openport("Brother TD-4750TN");
BROLIB_DLL.setup("100", "63.5", "4", "8", "0", "0", "0");
BROLIB_DLL.clearbuffer();
BROLIB_DLL.barcode("100", "100", "128", "100", "1", "0", "2", "2", "Barcode Test");
BROLIB_DLL.printerfont("100", "250", "3", "0", "1", "1", "Print Font Test");
BROLIB_DLL.windowsfont(100, 300, 24, 0, 0, 0, "ARIAL", "Windows Arial Font Test");
BROLIB_DLL.downloadpcx("UL.PCX", "UL.PCX");
BROLIB_DLL.sendcommand("PUTPCX 100,400,\"UL.PCX\"");
BROLIB_DLL.printlabel("1", "1");
BROLIB_DLL.closeport();

```

4.2 例 (2)

```

BROLIB_DLL.setup("100", "19.5", "4", "8", "0", "0", "0");
BROLIB_DLL.openport("Brother TD-4650TNWBR");
BROLIB_DLL.clearbuffer();
BROLIB_DLL.rfidWrite("0", "H", "0", "12", "EPC", "123456789012");
BROLIB_DLL.rfidRead("0", "H", "0", "12", "EPC");
BROLIB_DLL.printlabel("1", "1");
label2.Text = BROLIB_DLL.rfidGetReadData();
BROLIB_DLL.closeport();

```

5 About

5.1 対応モデル

> [Brother Windows DLLの対応モデル一覧](#)

brother