

# FTP639USL プリンタドライバ機能仕様書

2006/01/16

Driver ver. 1.03 , I/F USB

for Windows 2000/XP

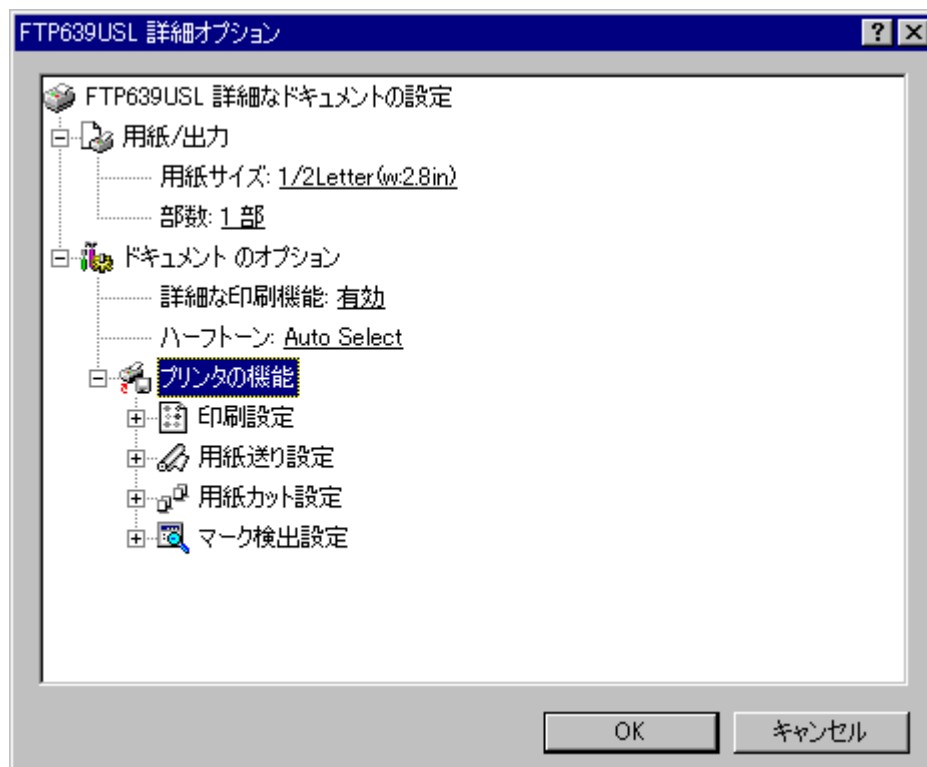
## 概要

本書では、プリンタの機能及び搬送路(プレゼンタ)制御に関する機能説明と設定方法、本ドライバに実装されている API と使用例について記述します。

プリンタドライバのインストール方法等については、ReadmeJ.txt を参照してください。

## プリンタの機能設定画面

1. 「スタートメニュー」から「設定(S)」、「プリンタ(P)」を選択し、「プリンタ」ウィンドウを開きます。
2. プリンタのアイコンをクリックします。
3. 「ファイル(F)」メニューから「プロパティ(R)」を選択し、設定画面を開きます。
4. 「印刷設定(I)」ボタン、「詳細設定(V)」ボタンをクリックし、次の設定画面を開きます。



## プリンタの機能設定

### ・印刷設定項目

[印字品質]	<p>使用する用紙を選択します。</p> <p>プリンタは選択した用紙に応じたモードで印刷を行います。</p> <p>詳細は「FTP639USLxxx 製品仕様」の「G. 制御コマンド仕様」、 「GS E+n」を参照してください。</p> <p>(Custom)を選択した場合は[エネルギー補正]の設定が有効になります。</p> <p>印字品質/エネルギー補正をしないを選択した場合は、印字品質設定およびエネルギー補正設定を行いません。</p>
[TYPE]	<p>印字品質で指定するモードを選択します。</p> <p>[印字品質]と[TYPE]設定についての詳細は「FTP639USLxxx 製品仕様」の「制御コマンド仕様」-「GS E+n」を参照してください。</p>
[エネルギー補正]	<p>印加エネルギーを補正し、印字濃度を設定します。</p> <p>詳細は「FTP639USLxxx 製品仕様」の「G. 制御コマンド仕様」、 「FS E+n」を参照してください。</p>
[ページモード]	<p>印刷ページのモードを選択します。</p> <ul style="list-style-type: none"><li>・オフ 最終ページ最後尾の空白を削除し、次の印刷とを連続的に 行います。</li><li>・オン 設定された用紙長で印刷を行います。</li></ul>
[印刷速度]	<p>印刷速度を設定します。</p> <ul style="list-style-type: none"><li>・低速 印刷速度が 30 mm/s となります。</li><li>・中速 印刷速度が 80 mm/s となります。</li><li>・高速 印刷速度が 125 mm/s となります。</li><li>・超高速 印刷速度が 200 mm/s となります。</li></ul>

## ・用紙送り設定項目

[ドキュメント開始時]	ドキュメント開始時の用紙送り量を設定します。 単位はドットライン（1 mm = 8ドットライン）で0 ～ 255 の範囲で設定します。
[ページ開始時]	印刷ページ開始時の用紙送り量を設定します。 単位はドットライン（1 mm = 8ドットライン）で0 ～ 255 の範囲で設定します。
[ページ終了時]	印刷ページ終了時の用紙送り量を設定します。 単位はドットライン（1 mm = 8ドットライン）で0 ～ 255 の範囲で設定します。
[ドキュメント終了時]	ドキュメント終了時の用紙送り量を設定します。 単位はドットライン（1 mm = 8ドットライン）で0 ～ 255 の範囲で設定します。

## ・用紙カット設定項目

[用紙カット]	用紙のカットタイミングを選択します。 <ul style="list-style-type: none"><li>・なし 用紙カットを行いません。</li><li>・ページ毎 ページ印刷毎に用紙カットを行います。</li><li>・ドキュメント毎 ドキュメント(全ページ)印刷後に用紙カットを行います。</li></ul> <p>カットなしを選択した場合、[カット方法] [マーク検出]、 [カット前用紙送り量]、マーク検出設定の項目は無効となります。</p>
[カット方法]	カットの方法を選択します。 <ul style="list-style-type: none"><li>・全カット 用紙を全カットします。</li><li>・部分カット 用紙中央部を残してカットします。</li></ul>
[カット前用紙送り量]	カット前の用紙送り量を設定します。 単位はドットライン（1 mm = 8ドットライン）で0 ～ 255 の範囲で設定します。

## ・マーク検出設定

---

[マーク検出]	<p>用紙カット前のマーク検出を設定します。</p> <ul style="list-style-type: none"><li>・オフ 用紙マーク検出を行いません。</li><li>・オン 用紙マーク検出を行ない、カット位置まで用紙を送ります。</li></ul> <p>マーク検出をオンと選択した場合、[用紙送り量]の設定は無効となります。</p>
[マーク頭出し量]	<p>マーク検出後の頭出し量を設定します。</p> <p>単位はドットライン（1 mm = 8 ドットライン）で 0 ～ 255 の範囲で設定します。</p>

---

## 搬送路制御設定画面

1. 「スタートメニュー」から「設定(S)」、「プリンタ(P)」を選択し、「プリンタ」ウィンドウを開きます。
2. プリンタのアイコンをクリックします。
3. 「ファイル(F)」メニューから「プロパティ(R)」を選択し、設定画面を開きます。
4. 設定画面の搬送路設定タブをクリックし、搬送路設定画面に切替えます。

## 搬送路制御設定

### 設定項目の説明

[搬送路を使用する] 搬送路の使用を選択します。

- ☒ …使用する  
☐ …使用しない

＜以下の設定は[搬送路を使用する]と選択した場合において有効となります。＞

[排出待機時間] 搬送路に用紙を格納する時間を 0～30,000 ミリ秒の範囲で指定します。指定した時間が経過すると、用紙の排出もしくは回収動作を行います。

[排出量] 用紙排出口から用紙を排出する量を 0～255 ミリメートルの範囲で指定します。  
詳細は「FTP-639USLxxx 製品仕様書」の「G. 制御コマンド仕様」、「GS t+n」を参照してください。

[回収機能] 用紙の回収機能を選択します。

- ☒ …用紙の回収を行う  
☐ …用紙の回収を行わない

＜以下の設定は[用紙の回収を行う]と選択した場合において有効となります。＞

[取り込み待機時間] 排出された用紙を取り込むまでの時間を 0～30,000 ミリ秒の範囲で指定します。指定した時間が経過する前に用紙が排出口から取り外された場合、取り込み動作は行われません。

[排出回数] 用紙の排出および取り込み動作を繰り返す回数を 0～10 回の範囲で指定します。0 を指定した場合、用紙の排出は行われず、用紙の回収のみ行われます。  
2 回以上を指定する場合は搬送路内で用紙が詰まる恐れがありますので十分ご注意ください。

[用紙回収量] 用紙回収動作において、回収センサにて用紙ありから用紙なしを検知した後の搬送量を選択します。  
詳細は「FTP639USLxxx 製品仕様」の「G. 制御コマンド仕様」、「GS x+n」を参照してください。

## 搬送路使用時の注意事項

- ・プリンタの機能設定に関わらず、ページモードオン、ページ毎の全カットで動作します。
- ・印刷後、用紙排出口から用紙を抜き取るか用紙が回収されるまで、次の印刷は受け付けません。
- ・複数のページを印刷する場合、次のページは用紙排出口から用紙が抜き取られるか用紙が回収されるまで印刷されません。

## プリンタドライバ API

ドライバセットに含まれる FTPCtrl.DLL は、アプリケーションから USB インターフェースを介してサーマルプリンタのステータス取得を行うことを目的とした DLL です。この DLL 内のエクスポート関数を呼び出すことで、FTPCtrl.DLL はベンダリクエストコマンドを発行し、その受信データをアプリケーションに通知するため、容易にステータスを取得を行うことが可能です。

### - エクスポート関数 -

エクスポート関数名	機能
FclTP_Search_USB	USB ポートドライバの検索/ハンドル取得
FclTP_GetVendorCommand	ベンダリクエストコマンド発行/データ取得

### - エクスポート関数の詳細 -

#### ULONG FclTP\_Search\_USB ( void )

USB ポートドライバの検索/ハンドルの取得を行います。

#### パラメータ

なし

#### 戻り値

=1 : 検索/ハンドル取得成功  
≠1 : エラー

#### 解説

FclTP\_Search\_USB 関数は、USB ポートドライバを検索し、ハンドルを取得します。  
取得したハンドルは DLL 内に保存され、FreeLibrary 関数が実行されるまで保持されます。  
FclTP\_GetVendorCommand 関数を実行する前に、必ず FclTP\_Search\_USB 関数を実行し、ハンドル取得をする必要があります。エラーを返す場合は、プリンタドライバが正常にインストールされていない可能性があります。

**ULONG FcITP\_GetVendorCommand ( PVENDOR\_COMMAND *lpVendorCmd*, RECIEVE\_DATA *lpRcvData* )**  
ベンダコマンドを発行し、データを取得します。

### パラメータ

#### ***lpVendorCmd***

VENDOR\_COMMAND 型の構造体変数のポインタを指定します。

#### ***lpRcvData***

RECIEVE\_DATA 型の構造体変数のポインタを指定します。

### 戻り値

=1 : 正常終了  
≠1 : エラー

### 解説

FcITP\_GetVendorCommand 関数は、*lpVendorCmd* のメンバー変数 *bRequest* で指定したベンダリクエストを発行し、その受信データを *lpRcvData* に格納します。

各構造体の型は以下の通りです。

typedef struct \_VENDOR\_COMMAND

```
{
    USHORT    unitLength;           // VENDOR_COMMAND 構造体の長さを指定
    UCHAR     bRequest;            // リクエストを指定
    UCHAR     wValueH;             // wValue の High バイト
    UCHAR     wValueL;            // wValue の Low バイト
    UCHAR     wIndexH;            // wIndex の High バイト
    UCHAR     wIndexL;            // wIndex の Low バイト
    UCHAR     wLengthH;            // wLength の High バイト
    UCHAR     wLengthL;           // wLength の Low バイト
} VENDOR_COMMAND, *PVENDOR_COMMAND;
```

typedef struct \_RECIEVE\_DATA

```
{
    USHORT    unitLength;           // RECIEVE_DATA 構造体の長さを指定
    BOOLEAN   DataValid;            // 受信データ有効(TRUE)/無効(FALSE)
    ULONG     DataLength;           // 受信データ長(バイト)
    UCHAR     Data[256];            // 受信データ格納バッファ
} RECIEVE_DATA, *PRECIEVE_DATA;
```

プリンタステータス取得を行う場合は、*lpVendorCmd* のメンバー変数に以下の値を設定します。

```
bRequest = 1
wValueH = 0 wValueL = 0
wIndexH = 0 wIndexL = 0
wLengthH = 0 wLengthL = 6
```

FcITP\_GetVendorCommand 関数が失敗した場合、戻り値は≠1 となり、DataValid は FALSE となります。

- \* リクエストの種類や設定パラメータ、その受信データについてはプリンタ製品仕様書を参照して下さい。



## - エクスポート関数の使用方法 -

- 1) LoadLibrary 関数を使用して、Windows フォルダにある FTPCtrl.DLL をロードし、モジュールハンドルを取得します。
- 2) FTPCtrl.DLL をロード後、GetProcAddress 関数を実行し、DLL 内のエクスポート関数のアドレスを取得します。
- 3) GetProcAddress 関数の戻り値のエクスポート関数のポインタを使用して、DLL 内のエクスポート関数を呼び出します。

## - プログラム例) FclTP\_GetVendorCommand 関数によるステータス取得 -

```
//Define the function prototype
typedef ULONG (*pFclTP_Search_USB)(void);
typedef ULONG (*pFclTP_GetVendorCommand)(PVENDOR_COMMAND pVendorCmd, RECIEVE_DATA pRcvData);

HMODULE hDll;
VENDOR_COMMAND VendorCmd;
RECIEVE_DATA RcvData;

pFclTP_Search_USB fnFclTP_Search_USB;
pFclTP_GetVendorCommand fnFclTP_GetVendorCommand;

//Load the DLL and keep the handle to it
hDll = LoadLibrary("FTPCtrl.DLL");

// If the handle is valid, try to get the function address
if (NULL != hDll) {
    //Get pointer to our function using GetProcAddress
    fnFclTP_Search_USB = (pFclTP_Search_USB)GetProcAddress(hDll,"FclTP_Search_USB");
    fnFclTP_GetVendorCommand = (pFclTP_GetVendorCommand)GetProcAddress(hDll,"FclTP_GetVendorCommand");

    if ( (fnFclTP_Search_USB)() == 1 ) { // USB ポートドライバ検索及び判定
        VendorCmd.bRequest = 1;
        VendorCmd.wValueH = 0; VendorCmd.wValueL = 0;
        VendorCmd.wIndexH = 0; VendorCmd.wIndexL = 0;
        VendorCmd.wLengthH = 0; VendorCmd.wLengthL = 6; //受信データは 6 バイト
        VendorCmd.unitLength = sizeof(VENDOR_COMMAND);
        RcvData.unitLength = sizeof(RECIEVE_DATA);

        if ( (fnFclTP_GetVendorCommand)(&VendorCmd, &RcvData) == 1 ) { // ベンダリクエスト実行
            if ( RcvData.Data[2] & 0x04 ) { // RcvData.Data[0]~RcvData.Data[5]はステータスデータ
                // 用紙無し
            } else if ( RcvData.Data[1] & 0x04 ) {
                // ヘッドアップ
            }
        }
    }
}

//Free the library
FreeLibrary(hDll);
}
```