

プリンタステータス取得用モジュール(FTPCtrl.DLL) API 仕様

2010 年 1 月 12 日
富士通コンポーネント(株)

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
FclTP_Search_USBEx	プリンタドライバ名と関連付けられている USB ポートドライバの検索/ハンドル取得
FclTP_GetVendorCommandEx	プリンタドライバ名と関連付けられている USB プリンタへのベンダリクエストコマンド発行/データ取得
FclTP_Search_USB_B	USB ポートドライバの検索/ハンドル取得 (stdcall 呼び出し規約専用)
FclTP_GetVendorCommand_B	ベンダリクエストコマンド発行/データ取得 (stdcall 呼び出し規約専用)
FclTP_Search_USBEx_B	プリンタドライバ名と関連付けられている USB ポートドライバの検索/ハンドル取得 (stdcall 呼び出し規約専用)
FclTP_GetVendorCommandEx_B	プリンタドライバ名と関連付けられている USB プリンタへのベンダリクエストコマンド発行/データ取得 (stdcall 呼び出し規約専用)

- エクスポート関数の仕様 -

ULONG FclTP_Search_USB (void)

USB ポートドライバの検出/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出し、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*, PRECEIVE_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

RECEIVE_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 _RECEIVE_DATA
{
    USHORT    unitLength;           // RECEIVE_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
} RECEIVE_DATA, *PRECEIVE_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 = 4;
```

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。実行方法は本ドキュメントの 7 ページを参照して下さい。

- * 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。

ULONG FclTP_Search_USBEx (LPTSTR *lpPrinterDriverName*)

プリンタドライバ名と関連付けられている USB ポートドライバの検出及びハンドルの取得を行います。

パラメータ

lpPrinterDriverName

プリンタドライバのインストール時に割り当てたプリンタ名を指定します。

戻り値

=1 : 検索/ハンドル取得成功
≠1 : エラー

解説

FclTP_Search_USBEx 関数は、プリンタドライバと関連付けられている USB ポートドライバを検出し、ハンドルを取得します。

取得したハンドルは *lpPrinterDriverName* で指定されたプリンタドライバ名と共に DLL 内に保存します。さらに別のプリンタドライバ名を指定し、USB ポートドライバ検出とハンドル取得が成功した場合は、前データとは別に保存します。したがって DLL 内に複数のプリンタドライバ名とハンドルを保存することが可能です。

DLL 内に保存している全データは FreeLibrary 関数が実行されるまで保持されます。

FclTP_GetVendorCommandEx 関数を実行する前に、必ず FclTP_Search_USBEx 関数を実行し、ハンドルを取得しておく必要があります。本関数が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、もしくは *lpPrinterDriverName* で指定するプリンタドライバ名に誤りがある可能性があります。

ULONG FcITP_GetVendorCommandEx (LPTSTR *lpPrinterDriverName*,
PVENDOR_COMMAND *lpVendorCmd*,
PRECEIVE_DATA *lpRcvData*)

プリンタドライバ名と関連付けられている USB プリンタへのベンダリクエストコマンド発行及びデータ取得を行います。

パラメータ

lpPrinterDriverName

プリンタドライバのインストール時に割り当てたプリンタ名を指定します。

lpVendorCmd

VENDOR_COMMAND 型の構造体変数のポインタを指定します。

lpRcvData

RECEIVE_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_GetVendorCommandEx 関数はプリンタドライバ名と関連付けられている USB プリンタへ *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。

本関数を実行する場合は、FcITP_Search_USBEx 関数を実行し、USB ポートドライバの検索とハンドル取得を行っておく必要があります。FcITP_Search_USB 関数のみを行った場合は、プリンタドライバ名と関連付けられている USB プリンタを判定できないため、本関数はエラーを返します。

lpVendorCmd、*lpRcvData* の構造体の設定方法及び実行方法は、FcITP_GetVendorCommand 関数と同じです。本ドキュメントの 2、8 ページを参照して下さい。

– stdcall呼び出し規約専用エクスポート関数 –

Visual Basic 6.0 以前の Visual Basic 等、エクスポート関数呼び出し時の呼び出し規約に“stdcall”を指定できない環境の場合は以下のエクスポート関数を用いてください。

ULONG FcITP_Search_USB_B (void)

USB ポートドライバの検出/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

機能は FcITP_Search_USB 関数と同じです。本ドキュメントの 1 ページを参照して下さい。

ULONG FcITP_GetVendorCommand_B (PVENDOR_COMMAND lpVendorCmd, PRECEIVE_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

RECEIVE_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

機能は FcITP_GetVendorCommand 関数と同じです。本ドキュメントの 2 ページを参照して下さい。

ULONG FcITP_Search_USBEx_B (LPTSTR *lpPrinterDriverName*)

プリンタドライバ名と関連付けられている USB ポートドライバの検出及びハンドルの取得を行います。

パラメータ

lpPrinterDriverName

プリンタドライバのインストール時に割り当てたプリンタ名を指定します。

戻り値

=1 : 検索/ハンドル取得成功

≠1 : エラー

解説

機能は FcITP_Search_USBEx 関数と同じです。本ドキュメントの 3 ページを参照して下さい。

**ULONG FcITP_GetVendorCommandEx_B (LPTSTR *lpPrinterDriverName*,
PVENDOR_COMMAND *lpVendorCmd*,
PRECEIVE_DATA *lpRcvData*)**

プリンタドライバ名と関連付けられている USB プリンタへのベンダリクエストコマンド発行及びデータ取得を行います。

パラメータ

lpPrinterDriverName

プリンタドライバのインストール時に割り当てたプリンタ名を指定します。

lpVendorCmd

VENDOR_COMMAND 型の構造体変数のポインタを指定します。

lpRcvData

RECEIVE_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

機能は FcITP_GetVendorCommandEx と同じです。本ドキュメントの 4 ページを参照して下さい。

- 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 1) FclTP_GetVendorCommand関数によるステータス取得 -

```
// プロトタイプ宣言
typedef ULONG (*pFclTP_Search_USB)(void);
typedef ULONG (*pFclTP_GetVendorCommand)(PVENDOR_COMMAND pVendorCmd, PRECEIVE_DATA pRcvData);

HMODULE hDll;
VENDOR_COMMAND VendorCmd;
RECEIVE_DATA RcvData;

pFclTP_Search_USB fnFclTP_Search_USB;
pFclTP_GetVendorCommand fnFclTP_GetVendorCommand;

// DLL をロード、ハンドル取得
hDll = LoadLibrary("FTPCtrl.dll");

if (hDll != NULL) {
    // エクスポート関数のアドレスを取得
    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 ポートドライバ検出

        VendorCmd.bRequest = 1;
        VendorCmd.wValueH = 0;      VendorCmd.wValueL = 0;
        VendorCmd.wIndexH = 0;      VendorCmd.wIndexL = 0;
        VendorCmd.wLengthH = 0;      VendorCmd.wLengthL = 4;          // 受信データは 4 バイト
        VendorCmd.unitLength = sizeof(VENDOR_COMMAND);
        RcvData.unitLength = sizeof(RECEIVE_DATA);

        if ( (fnFclTP_GetVendorCommand>(&VendorCmd, &RcvData) == 1 ) { // ベンダリクエスト発行

            // RcvData.Data[0]~RcvData.Data[3]は受信したステータスデータ
            if ( RcvData.Data[2] & 0x04 ) {
                // 用紙無し
                ...
            } else if ( RcvData.Data[1] & 0x04 ) {
                // ヘッドアップ
                ...
            }
        }
    }
}
// DLL アンロード
FreeLibrary(hDll);
}
```

－ プログラム例 2) FclTP_GetVendorCommandEx関数によるステータス取得 －

/*プリンタドライバインストール時にプリンタ名を “FTP63A (640pixels)_1”, “FTP63A (640pixels)_2” とした時のケースです。*/

// プロトタイプ宣言

```
typedef ULONG (*pFclTP_Search_USBEx)(LPTSTR lpPrinterDriverName);
typedef ULONG (*pFclTP_GetVendorCommandEx)(LPTSTR lpPrinterDriverName,
                                           PVENDOR_COMMAND pVendorCmd, PRECEIVE_DATA pRcvData);
```

```
HMODULE hDll;
VENDOR_COMMAND VendorCmd;
RECEIVE_DATA RcvData;
```

```
pFclTP_Search_USBEx fnFclTP_Search_USBEx;
pFclTP_GetVendorCommandEx fnFclTP_GetVendorCommandEx;
```

// DLL をロード、ハンドル取得

```
hDll = LoadLibrary("FTPCtrl.dll");
```

```
if (hDll != NULL) {
```

// エクスポート関数のアドレスを取得

```
fnFclTP_Search_USBEx = (pFclTP_Search_USBEx)GetProcAddress(hDll, "FclTP_Search_USBEx");
```

```
fnFclTP_GetVendorCommandEx = (pFclTP_GetVendorCommandEx)GetProcAddress(hDll, "FclTP_GetVendorCommandEx");
```

```
if ( (fnFclTP_Search_USBEx)("FTP63A (640pixels)_1") == 1 ) { // USB ポートドライバ検出
```

```
    VendorCmd.bRequest = 1;
    VendorCmd.wValueH = 0;    VendorCmd.wValueL = 0;
    VendorCmd.wIndexH = 0;    VendorCmd.wIndexL = 0;
    VendorCmd.wLengthH = 0;    VendorCmd.wLengthL = 4; // 受信データは 4 バイト
    VendorCmd.unitLength = sizeof(VENDOR_COMMAND);
    RcvData.unitLength = sizeof(RECEIVE_DATA);
```

```
    if ( (fnFclTP_GetVendorCommandEx)("FTP63A (640pixels)_1", &VendorCmd, &RcvData) == 1 ) { // ベンダリクエスト発行
```

// RcvData.Data[0]～RcvData.Data[3]は受信したステータスデータ

```
    if ( RcvData.Data[2] & 0x04 ) {
```

// 用紙無し

...

```
    } else if ( RcvData.Data[1] & 0x04 ) {
```

// ヘッドアップ

...

```
    }
```

```
    }
```

```
}
```

```
if ( (fnFclTP_Search_USBEx)("FTP63A (640pixels)_2") == 1 ) { // 次のプリンタ
```

```
    VendorCmd.bRequest = 1;
    VendorCmd.wValueH = 0;    VendorCmd.wValueL = 0;
    VendorCmd.wIndexH = 0;    VendorCmd.wIndexL = 0;
    VendorCmd.wLengthH = 0;    VendorCmd.wLengthL = 4; // 受信データは 4 バイト
    VendorCmd.unitLength = sizeof(VENDOR_COMMAND);
    RcvData.unitLength = sizeof(RECEIVE_DATA);
```

```
    if ( (fnFclTP_GetVendorCommandEx)("FTP63A (640pixels)_2", &VendorCmd, &RcvData) == 1 ) { // ベンダリクエスト発行
```

// RcvData.Data[0]～RcvData.Data[3]は受信したステータスデータ

```
    if ( RcvData.Data[2] & 0x04 ) {
```

// 用紙無し

...

```
    } else if ( RcvData.Data[1] & 0x04 ) {
```

// ヘッドアップ

...

```
    }
```

```
    }
```

```
}
```

// DLL アンロード

```
FreeLibrary(hDll);
```

```
}
```


－ プログラム例 3) FclTP_GetVendorCommand関数による印刷終了確認方法 －

/*イメージデータ送信後に FS r コマンド(0x1C 0x72 0x02)を送信したときのケースです。*/

```
// プロトタイプ宣言
typedef ULONG (*pFclTP_Search_USB)(void);
typedef ULONG (*pFclTP_GetVendorCommand)(PVENDOR_COMMAND pVendorCmd, PRECEIVE_DATA pRcvData);

HMODULE hDll;
VENDOR_COMMAND VendorCmd;
RECEIVE_DATA RcvData;

pFclTP_Search_USB fnFclTP_Search_USB;
pFclTP_GetVendorCommand fnFclTP_GetVendorCommand;

// DLL をロード、ハンドル取得
hDll = LoadLibrary("FclTPCtrl.dll");

if (hDll != NULL) {
    // エクスポート関数のアドレスを取得
    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 ポートドライバ検出

        VendorCmd.bRequest = 1;
        VendorCmd.wValueH = 0;    VendorCmd.wValueL = 0;
        VendorCmd.wIndexH = 0;    VendorCmd.wIndexL = 0;
        VendorCmd.wLengthH = 0;    VendorCmd.wLengthL = 4; // 受信データは 4 バイト
        VendorCmd.unitLength = sizeof(VENDOR_COMMAND);
        RcvData.unitLength = sizeof(RECEIVE_DATA);

        while(1){
            //ベンダリクエストの4バイト目に変化するまでループする
            if ((fnFclTP_GetVendorCommand>(&VendorCmd, &RcvData) == 1) { // ベンダリクエスト発行

                // RcvData.Data[0]～RcvData.Data[3]は受信したステータスデータ
                if (RcvData.Data[3] == 0x02) {
                    // 印刷終了
                    ...
                    break; //ループを抜ける
                }
            }
        }
    }
}
// DLL アンロード
FreeLibrary(hDll);
}
```