

## USB プリンタステータス取得用モジュール(FTPCtrl.DLL) API 仕様

富士通コンポーネント(株)  
V1.01

FTPCtrl.DLL および FTPCtrl\_x64.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 はベンダー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。

x86 アプリケーションから呼び出す場合は FTPCtrl.DLL、x64 アプリケーションから呼び出す場合は FTPCtrl\_x64.DLL をご使用下さい。

FTPCtrl.DLL と FTPCtrl\_x64.DLL の機能は同一です。

本 DLL の API はプリンタクラスのみ対応しています。

### 商標

- ・ Microsoft®、Windows®は、米国 Microsoft Corporation の米国およびその他の国における登録商標または商標です。
- ・ Windows の正式名称は、Microsoft Windows Operating System です。その他、本書に記載されている商品・サービス名は、各社の商標または登録商標です。

### 注意事項

- ・ 本書の内容は予告無く変更する場合があります。
- ・ 本書の一部あるいは全部を無断で複写・複製・転載することは、固くお断りします。
- ・ 当社は、本書に記載されている情報の使用に起因するいかなる損害に対しても責任を負うものではありません。
- ・ 本書に記載されている商品・サービス名は、各社の商標または登録商標です。

－ エクスポート関数 －

エクスポート関数名	機能
ULONG _cdecl FclTP_Search_USB	USB ポートドライバの検索/ハンドル取得
ULONG _cdecl FclTP_GetVendorCommand	ベンダーリクエストコマンド発行/データ取得
ULONG _cdecl 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
ULONG _cdecl 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ーリクエストコマンド発行/データ取得
ULONG _stdcall FclTP_Search_USB_B	USB ポートドライバの検索/ハンドル取得
ULONG _stdcall FclTP_GetVendorCommand_B	ベンダーリクエストコマンド発行/データ取得
ULONG _stdcall 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
ULONG _stdcall 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ー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仕様 －

ULONG \_cdecl 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。

パラメーター

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 _cdecl 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。

#### パラメーター

***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 で指定したベンダー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 tagVENDOR_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 tagRECEIVE_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;
```

プリント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; //受信データ数はプリンタによって異なります
```

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 8 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ー、受信データはプリンタによって異なります。詳細はプリンタ製品仕様書を参照して下さい。

ULONG \_cdecl FclTP\_Search\_USBEx ( LPSTR *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。

#### パラメーター

*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 \_cdecl FclTP\_GetVendorCommandEx ( LPSTR *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ーリクエストコマンド発行及びデータ取得を行います。

#### パラメーター

*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

FclTP\_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* で指定したベンダー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、FclTP\_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。FclTP\_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、FclTP\_GetVendorCommand 関数と同じです。本ドキュメントの 3、9 ページを参照して下さい。

ULONG \_\_stdcall FclTP\_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。

#### パラメーター

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 FclTP\_Search\_USB 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 \_\_stdcall FclTP\_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。

#### パラメーター

*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 FclTP\_GetVendorCommand 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 \_\_stdcall FclTP\_Search\_USBEx\_B ( LPSTR *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。

#### パラメーター

*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 FclTP\_Search\_USBEx 関数と同じです。本ドキュメントの 4 ページを参照して下さい。

ULONG \_\_stdcall FclTP\_GetVendorCommandEx\_B ( LPSTR *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ーリクエストコマンド発行及びデータ取得を行います。

#### パラメーター

*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 FclTP\_GetVendorCommandEx と同じです。本ドキュメントの 5 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 フォルダーにある FTPCtrl.DLL または FTPCtrl\_x64.DLL をロードし、モジュールハンドルを取得します。
- 2) FTPCtrl.DLL または FTPCtrl\_x64.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) FcITP\_GetVendorCommand 関数によるステータス取得 -

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

HMODULE hDll;
VENDOR_COMMAND VendorCmd;
RECEIVE_DATA RcvData;

pFcITP_Search_USB fnFcITP_Search_USB;
pFcITP_GetVendorCommand fnFcITP_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
    fnFcITP_Search_USB = (pFcITP_Search_USB)GetProcAddress(hDll, "FcITP_Search_USB");
    fnFcITP_GetVendorCommand = (pFcITP_GetVendorCommand)GetProcAddress(hDll, "FcITP_GetVendorCommand");

    if ((fnFcITP_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; // 受信データ数はプリンタによって異なります
        VendorCmd.unitLength = sizeof(VENDOR_COMMAND);
        RcvData.unitLength = sizeof(RECEIVE_DATA);

        if ((fnFcITP_GetVendorCommand)(&VendorCmd, &RcvData) == 1) { // ベンダーリクエスト発行

            // RcvData.Data[0]~RcvData.Data[3]は受信したステータスデータ
            // 各ビットのステータスはプリンタによって異なります
            if (RcvData.Data[2] & 0x04) {
                // 用紙無し
                ...
            } else if (RcvData.Data[1] & 0x04) {
                // プラテンオープン
                ...
            }
        }
    }
}

// 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“FTP639U (576pixels)\_1”, “FTP639U (576pixels)\_2” とした時のケースです。\*/

```
// プロトタイプ宣言
typedef ULONG (_cdecl *pFclTP_Search_USBEEx)(LPSTR lpPrinterDriverName);
typedef ULONG (_cdecl *pFclTP_GetVendorCommandEx)(LPSTR lpPrinterDriverName,
                                                    PVENDOR_COMMAND pVendorCmd, PRECEIVE_DATA pRcvData);

HMODULE hDll;
VENDOR_COMMAND VendorCmd;
RECEIVE_DATA RcvData;

pFclTP_Search_USBEEx fnFclTP_Search_USBEEx;
pFclTP_GetVendorCommandEx fnFclTP_GetVendorCommandEx;

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

if (hDll != NULL) {
    // エクスポート関数のアドレスを取得
    fnFclTP_Search_USBEEx = (pFclTP_Search_USBEEx)GetProcAddress(hDll, "FclTP_Search_USBEEx");
    fnFclTP_GetVendorCommandEx = (pFclTP_GetVendorCommandEx)GetProcAddress(hDll, "FclTP_GetVendorCommandEx");

    if ( (fnFclTP_Search_USBEEx)("FTP639U (576pixels)_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; // 受信データ数はプリンタによって異なります
        VendorCmd.unitLength = sizeof(VENDOR_COMMAND);
        RcvData.unitLength = sizeof(RECEIVE_DATA);

        if ( (fnFclTP_GetVendorCommandEx)("FTP639U (576pixels)_1", &VendorCmd, &RcvData) == 1 ) { // ベンダーリクエスト発行

            // RcvData.Data[0]～RcvData.Data[3]は受信したステータスデータ
            // 各ビットのステータスはプリンタによって異なります
            if ( RcvData.Data[2] & 0x04 ) {
                // 用紙無し
                ...
            } else if ( RcvData.Data[1] & 0x04 ) {
                // プラテンオープン
                ...
            }
        }
    }

    if ( (fnFclTP_Search_USBEEx)("FTP639U (576pixels)_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; // 受信データ数はプリンタによって異なります
        VendorCmd.unitLength = sizeof(VENDOR_COMMAND);
        RcvData.unitLength = sizeof(RECEIVE_DATA);

        if ( (fnFclTP_GetVendorCommandEx)("FTP639U (576pixels)_2", &VendorCmd, &RcvData) == 1 ) { // ベンダーリクエスト発行

            // RcvData.Data[0]～RcvData.Data[3]は受信したステータスデータ
            // 各ビットのステータスはプリンタによって異なります
            if ( RcvData.Data[2] & 0x04 ) {
                // 用紙無し
                ...
            } else if ( RcvData.Data[1] & 0x04 ) {
                // プラテンオープン
                ...
            }
        }
    }
}

FreeLibrary(hDll); // DLL アンロード
}
```

### － 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返信パラメーター設定コマンド(FS r n(=2))を送信したときのケースです。\*/

```
// プロトタイプ宣言
typedef ULONG (_cdecl *pFclTP_Search_USB)(void);
typedef ULONG (_cdecl *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;           //受信データ数はプリンタによって異なります
        VendorCmd.unitLength = sizeof(VENDOR_COMMAND);
        RcvData.unitLength = sizeof(RECEIVE_DATA);

        while(1){
            //ベンダーリクエストの 4 バイト目が変わるまでループする
            if ((fnFclTP_GetVendorCommand>(&VendorCmd, &RcvData) == 1) { // ベンダーリクエスト発行

                // RcvData.Data[0]～RcvData.Data[3]は受信したステータスデータ
                // 各ビットのステータスはプリンタによって異なります
                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);
}
```

## － プログラム例 4) FclTP\_GetVendorCommand 関数によるコマンドの応答取得方法 －

/\*メモリスイッチの送信コマンド(GS ( E pL pH fn a (fn=4))を送信したときのケースです。\*/

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

HMODULE hDll;
VENDOR_COMMAND VendorCmd;
RECEIVE_DATA RcvData;
int i = 0;
char text[256] = "";
char format[256] = "";

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 = 9;
        VendorCmd.wValueH = 0;        VendorCmd.wValueL = 0;
        VendorCmd.wIndexH = 0;        VendorCmd.wIndexL = 0;
        VendorCmd.wLengthH = 0;        VendorCmd.wLengthL = 10; //受信データ数はコマンドによって異なります
        VendorCmd.unitLength = sizeof(VENDOR_COMMAND);
        RcvData.unitLength = sizeof(RECEIVE_DATA);

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

            for (i = 0; i < 8; i++) {
                strcat_s(format, 256, "0x%02X ");
            }
            sprintf_s(text, 256, format, RcvData.Data[1], RcvData.Data[2], RcvData.Data[3], RcvData.Data[4],
                RcvData.Data[5], RcvData.Data[6], RcvData.Data[7], RcvData.Data[8]);
            MessageBox(NULL, text, "Command response", MB_OK);
        }
    }
    // DLL アンロード
    FreeLibrary(hDll);
}
```