FTP639USL Printer Driver Function Specifications

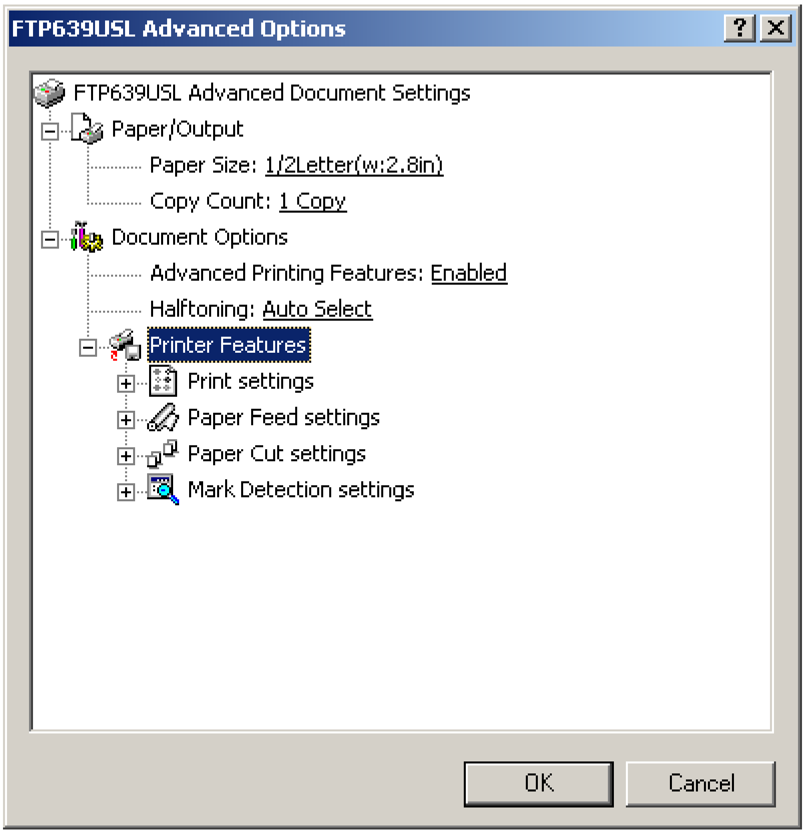
Date 2006/01/16
Driver version 1.03 , I/F USB
for Windows 2000/XP

[General]

This document describes how to make setting printer features, how to use the presenter by using printer driver, driver-API and example of the program.
For information about how to install a printer driver on your system, see "ReadmeE.txt".

[Accessing Printer Features menu]

1. Click **Start**, point to **Settings**, and then click **Printers**.
2. Right-click **FTP639UP**, and then click **Properties**.
3. Click '**Printing Preferences**' button in **General** tab.
4. Click '**Advanced Options**' button, and then '**Advanced Options**' window, as shown below, is displayed.



FTP639USL Advanced Options

FTP639USL Advanced Document Settings
- Paper/Output
  - Paper Size: 1/2Letter(w:2.8in)
  - Copy Count: 1 Copy
- Document Options
  - Advanced Printing Features: Enabled
  - Halftoning: Auto Select
  - Printer Features
    - Print settings
    - Paper Feed settings
    - Paper Cut settings
    - Mark Detection settings

[Printer Features]

・Print setting items

[Quality]	Select the paper that you use. If you select ’**Custom**’, ’**Energy Adjustment**’ item is available. For more information, see ″FTP−639USLXXX Product specifications >G. Control Commands >GS E +n″. If you select ’**Custom**’, ’**TYPE**’ item is available. If you select ‘**Do not use Quality/Energy Adjustment**’, neither Quality nor Energy Adjustment setting are done.
[TYPE]	Select the print mode set by the print quality. For more information about ″Quality″ and ″TYPE″ settings, see ″FTP−639USLXXX Product specifications>Control Commands> GS E +n″.
[Energy Adjustment]	Adjust energy when printing. For more information, see ″FTP−639USLXXX Product specifications >G. Control Commands >FS E +n″.
[Page Mode]	・Off The blank of last page end is deleted, and the following document is printed continuously. ・On Despite document, the paper is always fed according to paper size setting.
[Print Speed]	The printing speed can be set as follows. ・Low The printing speed becomes 30 mm/s. ・Medium The printing speed becomes 80 mm/s. ・High The printing speed becomes 125 mm/s. ・Ultra High The printing speed becomes 200 mm/s.

・Print Feed setting items

[Start of document]	The user can customize the feed lines at start of document. The unit is a dot-line (1mm = 8dot-lines) , the allowable range is from 0 to 255.
[Start of page]	The user can customize the feed lines at start of page. The unit is a dot-line (1mm = 8dot-lines) , the allowable range is from 0 to 255.
[End of page]	The user can customize the feed lines at end of page. The unit is a dot-line (1mm = 8dot-lines) , the allowable range is from 0 to 255.
[End of document]	The user can customize the feed lines at end of document. The unit is a dot-line (1mm = 8dot-lines) , the allowable range is from 0 to 255.

・Paper Cut setting items

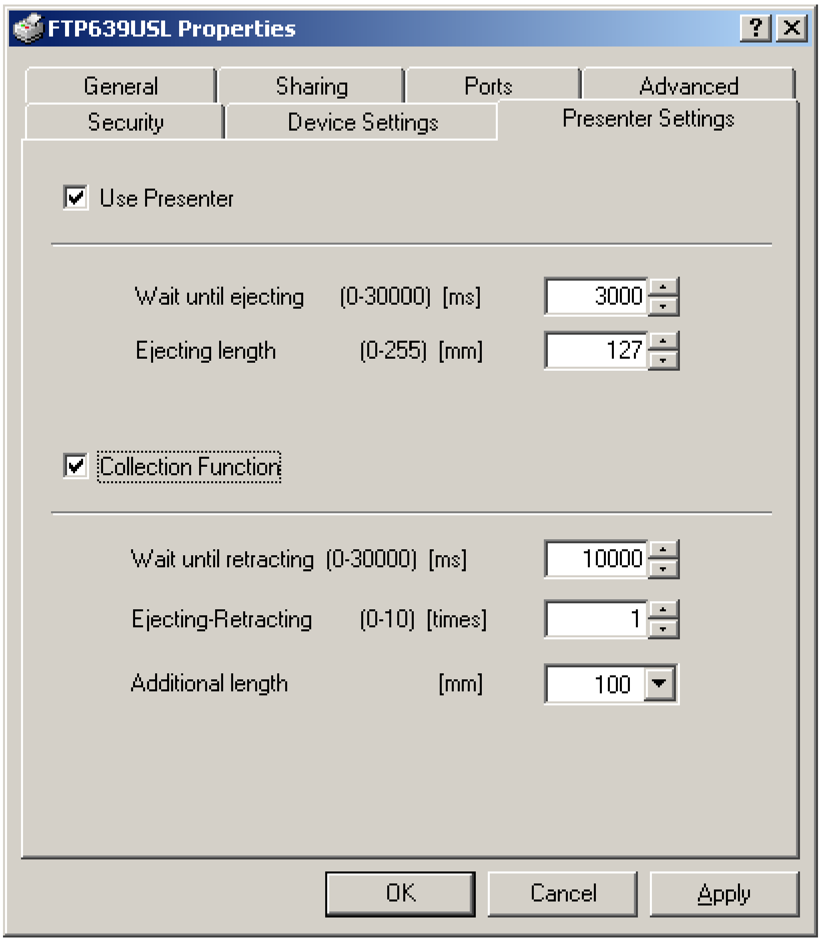
[Paper Cut]	・None The cutter is not used. ・Page Cut The cutter works page by page. ・Document Cut The cutter works only at the end of one document.
[Cutting Method]	・Full Cut Cutting the paper completely. ・Partial Cut Cutting other than center part of the paper.
[Cut Feed lines]	The user can adjust the feed lines before the paper cut. The unit is a dot-line (1mm = 8dot-lines) , the allowable range is from 0 to 255.

・Mark Detection setting items

[Mark Detection]	The mark detection can be set before the paper cut. ・Off The printer does not detect the mark.. ・On The printer detects the mark., and feeds the paper up to cutting position. If you select '**On**' , '**Cut Feed lines**' item. is invalid.
[Mark Feed lines]	The user can adjust the feed lines after the mark detection. The unit is a dot–line (1mm = 8dot–lines) , the allowable range is from 0 to 255.

[Accessing Presenter Settings window]

1. Click **Start**, point to **Settings**, and then click **Printers**.
2. Right-click **FTP639UP**, and then click **Properties**.
3. Click '**Printing Preferences**' button in **Presenter Settings** tab.
4. '**Presenter Settings**' window, as shown below, is displayed.



☑ Use Presenter

Wait until ejecting (0-30000) [ms] 3000

Ejecting length (0-255) [mm] 127

☑ Collection Function

Wait until retracting (0-30000) [ms] 10000

Ejecting-Retracting (0-10) [times] 1

Additional length [mm] 100

[Presenter Settings]

Setting items

[Use Presenter]	If you use the presenter, check. ☑・・・Use ☐・・・Not use

< If you use the presenter, the following settings are available. >

[Wait until ejecting]	Specifies the length of time to wait until ejecting the paper. This can be specified from 0 to 30,000 milliseconds. After the specified time passes, the presenter ejects the paper from the front of the mechanism or collects the paper from back of the mechanism
[Ejecting length]	Specifies the ejecting length. This can be specified from 0 to 255 mm. For more information, see ″FTP-639USLXXX Product specifications >G. Control Commands >GS t +n″.
[Collection Function]	If you use the collection function, check. ☑・・・Use ☐・・・Not use

< If you use the collection function, the following settings are available. >

[Wait until retracting]	Specifies the length of time to wait until retracting the paper. This can be specified from 0 to 30,000 milliseconds. If the paper is taken out from outlet before the specified time passes, retracting function does not function.
[Ejecting-Retracting]	Specifies the number of times to repeat ejecting and retracting. This can be specified from 0 to 10 times. If 0, the presenter never ejects the paper, and collects the paper from back of the mechanism.
[Additional length]	Specifies the additional length for collection function. Selects from the drop-down list. For more information, see ″FTP-639USLXXX Product specifications >G. Control Commands >GS x +n″.

[Note]

Note the following things, if you select to use the presenter when printing.

・Automatically Full-Cut the paper after printing a document, regardless of the printer features settings.

・After printing one document, you can't print another document until the paper that the presenter ejects is taken out or the presenter collects the paper.

・If you print a multi-page document, the next page is printed after the paper that presenter ejects is taken out or the presenter collects the paper.

[Printer Driver API]

The enclosed 'FTPCtrl.DLL' provides the exported functions to acquire the status of the USB thermal printer. By calling the exported functions, FTPCtrl.DLL sends the vendor request to printer and notifies response data, the applications can get the printer status easily.

– Exported DLL functions –

Exported DLL function	Effect
FcITP_Search_USB	Search USB port driver, and acquire the port driver handle.
FcITP_GetVendorCommand	Send Vender Commands to printer, and get response data from printer

– Specifications of exported DLL functions –

ULONG FcITP_Search_USB (void)

Search USB port driver, and acquire the port driver handle.

Parameters

Nothing

Return Value

＝1 : Success
≠1 : Failure

Remarks

The port driver handle that acquired by calling FcITP_Search_USB function is saved in DLL until the FreeLibrary function is executed. It is necessary to call FcITP_Search_USB function once and acquire the handle before FcITP_GetVendorCommand function is executed.

If FcITP_Search_USB function fails, there is a possibility that printer driver is not normally installed.

ULONG FcITP_GetVendorCommand (PVENDOR_COMMAND *lpVendorCmd*, PRECIEVE_DATA *lpRcvData*)

Send Vender Command, and acquire the response data from printer.

Parameters

lpVendorCmd

Pointer to the VENDOR_COMMAND structure.

lpRcvData

Pointer to the RECIEVE_DATA structure.

Return Value

＝1 : Success
≠1 : Failure

Remarks

FcITP_GetVendorCommand function send vendor request that is specified bRequest member of lpVendorCmd, and store response data to lpRcvData.

```
// VENDOR_COMMAND structure
typedef struct _VENDOR_COMMAND
{
   USHORT    unitLength;               // VENDOR_COMMAND structure size, in bytes
   UCHAR     bRequest;                 // specifies vendor request
   UCHAR     wValueH;                  // high-order byte of wValue
   UCHAR     wValueL;                  // low-order byte of wValue
   UCHAR     wIndexH;                  // high-order byte of wIndex
   UCHAR     wIndexL;                  // low-order byte of wIndex
   UCHAR     wLengthH;                 // high-order byte of wLength
   UCHAR     wLengthL;                 // low-order byte of wLength
} VENDOR_COMMAND, *PVENDOR_COMMAND;

// RECIEVE_DATA structure
typedef struct _RECIEVE_DATA
{
   USHORT    unitLength;               // RECIEVE_DATA structure size, in bytes
   BOOLEAN   DataValid;                // data valid (TRUE) / data invalid(FALSE)
   ULONG     DataLength;               // response data size, in bytes
   UCHAR     Data[256];                // buffer to storage response data
} RECIEVE_DATA, *PRECIEVE_DATA;
```

For example, get the printer status, specifies the value for the members of *lpVendorCmd*, as follows

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

if FcITP_GetVendorCommand function fails, return value is 1, and DataValid becomes FALSE.

＊ For more information about the kind of vendor request, specifies the value for the members of *lpVendorCmd* and response data, see ″FTP-639USLXXX Product specifications″.

– Use of exported DLL functions –

1) Use LoadLibrary function to load the 'FTPCtrl.DLL' in the Windows folder.
2) After the 'FTPCtrl.DLL' is loaded, call the GetProcAddress function to acquire the address of the exported DLL functions.
3) Call the exported DLL functions using the function pointers returned by GetProcAddress function.

– Example) Get the printer status by calling FcITP_GetVendorCommand function –

```
//Define the function prototype
typedef ULONG (*pFcITP_Search_USB)(void);
typedef ULONG (*pFcITP_GetVendorCommand)(PVENDOR_COMMAND pVendorCmd, PRECIEVE_DATA pRcvData);

HMODULE   hDll;
VENDOR_COMMAND VendorCmd;
RECIEVE_DATA RcvData;

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

 if ( (fnFcITP_Search_USB)() == 1 ) {                                  // Search USB port driver
   VendorCmd.bRequest = 1;
   VendorCmd.wValueH = 0;        VendorCmd.wValueL = 0;
   VendorCmd.wIndexH = 0;        VendorCmd.wIndexL = 0;
   VendorCmd.wLengthH = 0;       VendorCmd.wLengthL = 6;                //Response data size = 6bytes
   VendorCmd.unitLength = sizeof(VENDOR_COMMAND);
   RcvData.unitLength = sizeof(RECIEVE_DATA);

   if ( (fnFcITP_GetVendorCommand)(&VendorCmd, &RcvData) == 1 ) {      // vendor request
       if ( RcvData.Data[2] & 0x04 ) {                  // RcvData.Data[0]〜RcvData.Data[5] store the status data
          // Out of paper
       } else if ( RcvData.Data[1] & 0x04 ) {
          // Cover is opened
       }
   }
 }
 //Free the library
 FreeLibrary(hDll);
}
```