



F*TRAN+ Ver.6.0

操作説明書／プログラム応用編

第1版

株式会社富士通ビー・エス・シー

は じ め に

F*TRAN+ V6.0をお買い上げいただき、ありがとうございます。

F*TRAN+（エフトラン・プラス）は、汎用機／オフコン／UNIXなどのホストのファイル転送データと、パソコンの標準であるWindowsファイルとのデータ交換をする汎用性の高いファイル変換ユーティリティです。Windowsファイル間のデータ変換もできます。発売以来10年、多くのユーザにささえられている、実績あるソフトウェアです。

F*TRAN+ V6.0には、動作するOS別に2種類の製品があります。

- F*TRAN+ V6.0 Server
Server系、およびDesktop系OSで動作
- F*TRAN+ V6.0 Desktop
Desktop系OSで動作

F*TRAN+ V6.0では、Windows 7上での動作をサポートしています。

F*TRAN+のマニュアルには、導入編、解説編、コマンド編、マルチレコード編、プログラム応用編（本書）があります。

2010年4月
株式会社 富士通ビー・エス・シー

目 次

第1章 OLEオートメーション

1.1 OLEオートメーションとは？	2
--------------------	---

第2章 インターフェース

2.1 インターフェース	4
■ インターフェース	4
■ 全体の流れの参考図	5
■ I F t C m d のメソッド	6
■ I F t S i n k のメソッド (コールバック)	10
■ I F t Q u i t のメソッド	13
■ メッセージボックス、ダイアログボックスの種類	14
■ メッセージボックス、ダイアログボックスのボタンタイプ	14
■ メッセージボックス、ダイアログボックスの戻り値	14
2.2 リターンコード	15

第3章 オートメーションサンプル

3.1 V i s u a l C ++ のサンプル	18
3.2 V i s u a l B a s i c のサンプル	21

本書で用いる表記法

●本文と画面のパラメータ類の表記法

{ A B C }	A、B、またはCのうち、どれか1つを選択します。 省略はできません。
$\left \begin{array}{c} A \\ B \\ C \end{array} \right $	同上。
(A / B / C)	同上。
[A]	Aは省略できます。
[A / <u>B</u> / C]	A、B、またはCのうち、どれか1つを選択します。 省略が可能で、その場合、下線を引いたBを選択した ものとみなします。
$\left[\begin{array}{c} A \\ \underline{B} \\ C \end{array} \right]$	同上。
(A / [B] / C)	同上。ただし、[] でくくったBを選択したものと みなします。
X . . .	X類を A B C のように列挙します。
n、n n、< n >	1 0進数を指定します。 (< > は表記上の記号で、入力はしません)
x x H	1 6進でx xです。Hを省くこともあります。
↓	改行を意味します。リターンキーのシンボルです。
<u>a</u>	下線部を入力します。
<u>a b c</u> ↓	下線部を入力し、リターンキーを押します。
C T R L + A	コントロール (C T R L) キーを押しながら、Aキー を押します。コントロールAと読みます。
^ A	同上。
d :	ドライブ A : や C : など、任意のドライブ指定を表し ます。

◆注意 ----- 実画面と少し差異がある

本書に示す画面と実際の画面には、若干の差異がある場合があります。あらかじめ、ご了承ください。

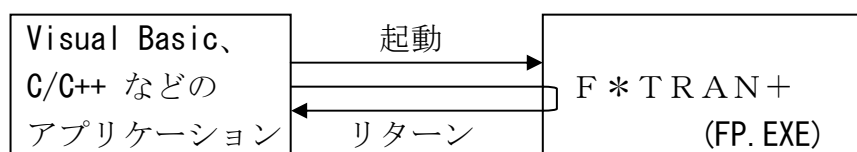
第1章

OLEオートメーション

1. 1 OLEオートメーションとは？

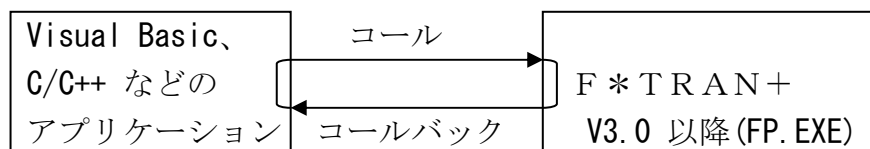
F*TRAN+ V3.0より、OLEオートメーションのインターフェースが提供されています。このインターフェースをアプリケーションから利用することで、F*TRAN+をユーザアプリケーションの「部品」として使用することができ、F*TRAN+の処理を細かにコントロールすることができます。

●OLEオートメーションを使わないF*TRAN+の利用（従来の方法）



F*TRAN+を外部プログラムとして起動し、プロセス終了後のリターン（正常、エラーのどちらか）を取得する。細かいコントロールはできない。非同期の動作となる。

●OLEオートメーションを使ったF*TRAN+の利用



F*TRAN+の処理を自関数としてコールし、メッセージやダイアログのコールバックを受け取ることもできる。詳細なエラーコードの判定ができ、F*TRAN+をアプリケーションの一部として使用することができる。同期した動作となる。

第2章

インターフェース

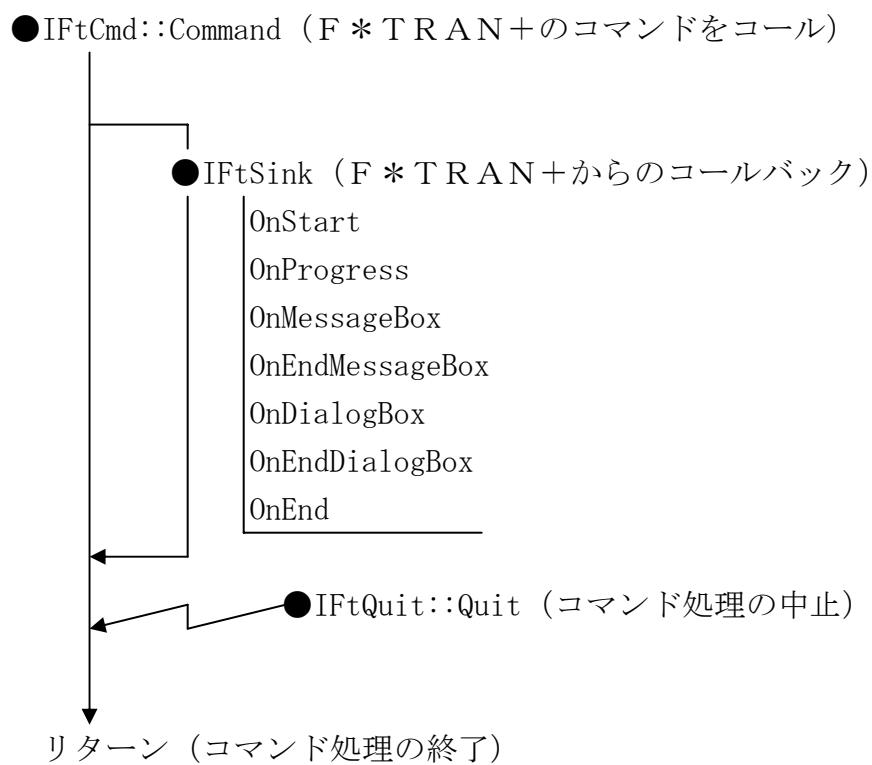
2. 1 インターフェース

F*TRAN+のOLEオートメーションのインターフェースは、つぎのとおりです。

■ インターフェース

● IFtCmd	クライアントとのインタフェース
メソッド	
Command	コマンドラインを実行し、終了後に復帰する。
GetLastFtErrorCode	F*TRAN+ のリターンコードを獲得する。
GetLastWinError	Win32 エラーまたは OLE エラーの最終値を獲得する。
GetLastMessage	実行ウインドウに出力された最終エラーメッセージを獲得する。
SetOwnerWindow	オーナーウインドウを設定する。
MessageBoxValue	OnMessageBox の戻り値を設定する。 OnMessageBox の中で使用する。
DialogBoxValue	OnDialogBox の戻り値を設定する。 OnDialogBox の中で使用する。
SelectWFile	ファイル選択ウインドウから選択したファイルを通知する。
SetCurrentDirectory	カレントディレクトリを設定する。
GetCurrentDirectory	カレントディレクトリを取得する。
GetInstallDirectory	F*TRAN+ のインストールディレクトリを取得する。
● IFtSink	コールバックのインタフェース
メソッド	
OnStart	コマンド開始のタイミングで呼ばれる。
OnEnd	コマンド終了のタイミングで呼ばれる。
OnProgress	処理中の%進行表示のために呼ばれる。
OnMessageBox	メッセージボックス表示の直前に呼ばれる。 MessageBoxValue で戻り値が設定されると、 F*TRAN+ のメッセージボックスは表示されない。
OnEndMessageBox	メッセージボックス終了直後に呼び出される。
OnDialogBox	ダイアログボックス表示の直前に呼ばれる。 DialogBoxValue で戻り値が設定されると、 F*TRAN+ のダイアログボックスは表示されない。
OnEndDialogBox	ダイアログボックス終了直後に呼び出される。
● IFtQuit	中止を指示するインタフェース
メソッド	
Quit	実行中のコマンドの中止を指示する。非同期。

■全体の流れの参考図



■ IFtCmd のメソッド

● IFtCmd::Command

```
long Command(  
    BSTR pCmd  
);
```

パラメータ	pCmd	コマンドラインと同じコマンド文字列を指定する。
リターン	0	正常
	0 以外	エラー
説明		コマンドの開始点である。 実行終了後に復帰する。 実行中にコールバックが呼び出される。

● IFtCmd::GetLastFtErrorCode

```
long GetLastFtErrorCode(  
);
```

パラメータ	なし。
リターン	F*TRAN+ のリターンコード。
説明	F*TRAN+ のリターンコードが得られる。 リターンコードは Command メソッド終了後、または、OnEnd コールバック時に有効である。

● IFtCmd::GetLastWinError

```
long GetLastWinError(  
);
```

パラメータ	なし。
リターン	Win32 エラーまたは OLE エラー。
説明	Win32 エラーまたは OLE エラーの最終値が得られる。 リターンコードは Command メソッド終了後、または、OnEnd コールバック時に有効である。

● IFtCmd::SetOwnerWindow

```
long SetOwnerWindow(
    long hWnd
);
```

パラメータ	hWnd	オーナーウインドウのハンドル。
リターン	0	正常
	0 以外	エラー
説明	実行ウインドウのオーナーウインドウを設定する。 実行ウインドウが表示されないときには、メッセージボックスやダイアログボックスのオーナーウインドウとなる。オーナーウインドウを設定しないと、デスクトップがオーナーウインドウになる。 Command、SelectWFile の呼び出し前に設定する。 また、hWnd が無効になったときは hWnd を 0 として呼び出す。	

● IFtCmd::GetLastMessage

```
BSTR GetLastMessage(
);
```

パラメータ	なし。
リターン	実行ウインドウに出力された最終エラーメッセージ。
説明	実行ウインドウの最後のメッセージが得られる。 最終的に正常でも途中でエラーがあったときには、そのエラーメッセージになる。 メッセージボックスに表示されたエラーメッセージを、このメソッドで得ることはできない。 最終エラーメッセージは Command メソッド終了後、または、OnEnd コールバック時に有効である。

● IFtCmd::MessageBoxValue

```
long MessageBoxValue(
    long lId
);
```

パラメータ	lId	メッセージボックスの終了コードを指定する。 0 は、このメソッド呼び出しなしと同じである。
リターン	0	
説明		OnMessageBox 内部で使用する。 メッセージボックスの戻り値を F*TRAN+ に知らせる。 F*TRAN+ は、その値に対応する処理に制御を移す。 このメソッドが呼び出されないとき、または lId に 0 が設定されると、 F*TRAN+ は、自身のメッセージボックスを表示する。

● IFtCmd::DialogBoxValue

```
long DialogBoxValue(
    long lId,
    BSTR lpEdit
);
```

パラメータ	lId	ダイアログボックスの戻り値を指定する。 0 は、このメソッド呼び出しなしと同じである。
	lpEdit	"" 固定。
リターン	0	
説明		OnDialogBox 内部で使用する。 ダイアログボックスの戻り値を F*TRAN+ に知らせる。 F*TRAN+ は、その値に対応する処理に制御を移す。 このメソッドが呼び出されないとき、または lId に 0 が設定されると、 F*TRAN+ は、自身のダイアログボックスを表示する。

● IFtCmd::SelectWFile

```
BSTR SelectWFile(
    BSTR lpTitle,
    BSTR lpDrive,
    BSTR lpFilter,
    long lIndex
);
```

パラメータ	lpTitle	ウィンドウのタイトルを指定する。
	lpDrive	ファイルのあるドライブのパス名を指定する。
	lpFilter	選択するファイルの種類をファイルフィルタで指定する。 例: "F*TRAN+スクリプトファイル(*.fp5) *.fp5 "
	lIndex	ファイルフィルタの何番目を規定値とするかを指定する。
リターン		選択されたファイル名。 キャンセルまたはエラーのときには、""となる。
説明		ファイル選択ウィンドウが表示される。 オペレータが選択したファイル名が復帰情報となる。 復帰情報として返されたファイルが必ず存在するとは限らないので注意すること。

● IFtCmd::SetCurrentDirectory

```
long SetCurrentDirectory(
    BSTR lpPath
);
```

パラメータ	lpPath	カレントとして設定するパス名
リターン		正常終了の場合は 0 。 エラーのときには、GetLastError の値。
説明		Command メソッド呼び出し前に設定する。実行途中で変更したときの結果は予想できない。

● IFtCmd::GetCurrentDirectory

```
BSTR GetCurrentDirectory(
);
```

パラメータ		なし。
リターン		取得したカレントディレクトリ。 エラーのときには、"" を返す。

● IFtCmd::GetInstallDirectory

```
BSTR GetInstallDirectory(  
);
```

パラメータ なし。

リターン 取得した F*TRAN+ のインストールディレクトリ。

■ IFtSink のメソッド (コールバック)

● IFtSink::OnStart

```
void OnStart(  
);
```

パラメータ なし。

リターン なし。

説明 Command 呼び出し直後に呼び出される。
実行ウィンドウが表示される前のタイミングである。

● IFtSink::OnEnd

```
void OnEnd(  
    long lReturnCode  
);
```

パラメータ lReturnCode Command のリターンコードと同じ。

リターン なし。

説明 Command の終了時に呼び出される。

● IFtSink::OnProgress

```
void OnProgress(  
    long lPercent  
);
```

パラメータ lPercent 1-100 までの進行状況である。

リターン なし。

説明 処理中の%進行表示のために呼ばれる。
呼ばれる率を制御することはできない。

● IFtSink::OnMessageBox

```
void OnMessageBox(
    long lFtFlag,
    long hWnd,
    BSTR lpText,
    BSTR lpCaption,
    long lType
);
```

パラメータ	lFtFlag	メッセージボックスの種類
	hWnd	オーナーウインドウ
	lpText	テキストへのポインタ
	lpCaption	タイトルへのポインタ
	lType	メッセージボックスのボタンタイプ
リターン	なし。	
説明	F*TRAN+ がメッセージボックスを表示する直前に呼び出される。 MessageBoxValue を呼び出し、メッセージボックスの戻り値を設定すると、 F*TRAN+ はメッセージボックスを表示しない。 C 言語のクライアントで Win32 関数::MessageBox() を lFtFlag を除いた パラメータで呼び出せば、F*TRAN+ と同じメッセージボックスになる。	

● IFtSink::OnEndMessageBox

```
void OnEndMessageBox(
    long lId
);
```

パラメータ	lId	メッセージボックスの戻り値である。
リターン	なし。	
説明	メッセージボックス終了直後に呼び出される。	

●IFtSink::OnDialogBox

```
void OnDialogBox(
    long lFtFlag,
    long hWnd,
    BSTR lpText,
    BSTR lpCaption,
    long lType
);
```

パラメータ	lFtFlag	ダイアログボックスの種類
	hWnd	オーナーウインドウ
	lpText	テキストへのポインタ
	pCaption	タイトルへのポインタ
	lType	ダイアログボックスのボタンタイプ
リターン	なし。	
説明	ダイアログボックス表示の直前に呼ばれる。 DialogBoxValue を呼び出し、ダイアログボックスの戻り値を設定すると、 F*TRAN+ はダイアログボックスを表示しない。	

●IFtSink::OnEndDialogBox

```
void OnEndDialogBox(
    long lId
);
```

パラメータ	lId	ダイアログボックスの戻り値である。
リターン	なし。	
説明	ダイアログボックス終了直後に呼び出される。	

■IFtQuit のメソッド

●IFtQuit::Quit

```
long Quit(  
);
```

パラメータ なし。

リターン 0

説明 中止を指示する。
このメソッドは中止を指示するだけなので、実際にコマンドが中止されるのは、このメソッドから復帰したときではない。

■メッセージボックス、ダイアログボックスの種類

FT_MB_QUERY 一般的な問い合わせ。(注：参照)
FT_DLG_GETPUT_SKIP 変換時の問い合わせ・確認。

注) FT_MB_QUERY には、つぎのようなものがある。

- 1) オペレーションの要求、MB_OKCANCEL
- 2) エラーの表示、MB_OK
- 3) 進行確認、MB_OKCANCEL
- 4) 進行確認、MB_YESNO

■メッセージボックス、ダイアログボックスのボタンタイプ

Win32 定義(メッセージボックス、および、ダイアログボックス)

MB_OK	戻り値	IDOK
MB_OKCANCEL	戻り値	IDOK、IDCANCEL
MB_RETRYCANCEL	戻り値	IDRETRY、IDCANCEL
MB_YESNO	戻り値	IDYES、IDNO

F*TRAN+の追加定義(ダイアログボックスのみ)

FT_DLG_ALLCANCEL	戻り値	IDOK、FT_IDALL、IDCANCEL、FT_IDALLCANCEL
------------------	-----	---------------------------------------

注) OnMessageBox の lType は上記の値にメッセージボックスのスタイルのフラグがORされている。上記の値を抜き出すためにはつぎのように Win32 定義の値でマスクする必要がある。(MB_TYPEMASK & lType)

■メッセージボックス、ダイアログボックスの戻り値

Win32 定義(メッセージボックス、および、ダイアログボックス)

IDOK
IDCANCEL
IDABORT
IDRETRY
IDIGNORE
IDYES
IDNO

F*TRAN+の追加定義(ダイアログボックスのみ)

FT_IDALL
FT_IDALLCANCEL

2. 2 リターンコード

● 正常終了コード

0 : 正常終了コード (E_FT_NORMAL)

● 再試行が可能な終了コード

1001 : メディアがセットされていない (E_FT_DEVICE_NOT_READY)

1002 : メディアが書き込み禁止 (E_FT_PROTECTED_DISK)

● 再試行が不可能な終了コード

9001 : 入力パラメータエラー (E_FT_PARAMETER)

9002 : 使用できないときに使用した (E_FT_PROTOCOL)

9003 : オペレータによるキャンセル (E_FT_CANCELED)

9004 : システムのリソース不足、メモリ獲得失敗など (E_FT_INSUFFICIENT_RESOURCES)

9005 : システムのリソース不足以外のエラー発生 (E_FT_WIN_ERROR)

9006 : システムのリソース不足のエラー発生 (E_FT_LIMIT_EXCEEDED)

9007 : 特定できないエラー発生 (E_FT_UNKNOWN)

9008 : 情報通知の事象発生 (E_FT_INFORMATION)

9009 : フロッピーメディアエラー(読み込み時) (E_FT_NO_DATA)

9010 : フロッピーメディアエラー(読み込み時) (E_FT_CRC_ERROR)

9011 : フロッピードライブのエラー(I Oエラー) (E_FT_IO_ERROR)

9012 : フロッピーメディアのエラー(読み込み時) (E_FT_INCORRECT_DRIVE)

9013 : フロッピーに入力ファイルがない (E_FT_NO_SUCH_FILE)

9014 : フロッピーに出力ファイルがすでにある (E_FT_ALREADY_EXISTS)

9015 : ファイルがライトプロテクトである (E_FT_PROTECT_FILE)

9016 : フロッピーのインデックスがすでにいっぱい (E_FT_INDEX_FULL)

9017 : フロッピーの空き領域がない (E_FT_NO_SPACE)

9018 : ファイルが異常な形式である (E_FT_INVALID_FILE)

9019 : F*TRAN でサポートされない形式のファイル (E_FT_UNSUPPORTED_FILE)

9020 : フロッピーが異常な形式である (E_FT_INVALID_DISK)

9021 : F*TRAN でサポートされない形式のフロッピー (E_FT_UNSUPPORTED_DISK)

9022 : ボリューム番号が正しくないフロッピー (E_FT_VOLUME_SEQUENCE)

9023 : Windows ファイルの作成でエラー発生 (E_FT_CREATE_FAILED_WIN)

9024 : Windows ファイルのオープンでエラー発生 (E_FT_OPEN_FAILED_WIN)

9025 : Windows ファイルのクローズでエラー発生 (E_FT_CLOSE_FAILED_WIN)

9026 : Windows ファイルのリードでエラー発生 (E_FT_READ_FAILED_WIN)

9027 : Windows ファイルのライトでエラー発生 (E_FT_WRITE_FAILED_WIN)

9028 : HOST	ファイルの作成でエラー発生	(E_FT_CREATE_FAILED_HOST)
9029 : HOST	ファイルのオープンでエラー発生	(E_FT_OPEN_FAILED_HOST)
9030 : HOST	ファイルのクローズでエラー発生	(E_FT_CLOSE_FAILED_HOST)
9031 : HOST	ファイルのリードでエラー発生	(E_FT_READ_FAILED_HOST)
9032 : HOST	ファイルのライトでエラー発生	(E_FT_WRITE_FAILED_HOST)
9033 : 設定ファイルの作成でエラー発生		(E_FT_CREATE_FAILED_SET)
9034 : 設定ファイルのオープンでエラー発生		(E_FT_OPEN_FAILED_SET)
9035 : 設定ファイルのクローズでエラー発生		(E_FT_CLOSE_FAILED_SET)
9036 : 設定ファイルのリードでエラー発生		(E_FT_READ_FAILED_SET)
9037 : 設定ファイルのライトでエラー発生		(E_FT_WRITE_FAILED_SET)
9038 : 設定ファイルまたは MAP 指定エラー		(E_FT_INVALID_DATA)
9039 : 設定ファイルやパラメータ設定に矛盾がある		(E_FT_INCONSISTANT)

●メッセージボックス、ダイアログボックスの類別コード

1 : 一般的な問い合わせ	(FT_MB_QUERY)
2 : リトライの問い合わせ	(FT_MB_RETRY)
3 : 重要障害による処理中断の表示	(FT_MB_FATAL)
4 : メディアの交換指示 (マルチボリューム)	(FT_MB_MV_CHANGE_VOLUME)
5 : メディアの交換指示 (マルチボリューム以外)	(FT_MB_CHANGE_VOLUME)
6 : 複数ファイル変換時のスキップ確認ダイアログ	(FT_DLG_GETPUT_SKIP)
7 : 複数ファイル削除時の確認ダイアログ	(FT_DLG_DELETE_SKIP)
8 : ディスクコピー時のフォーマット確認ダイアログ	(FT_DLG_DCFORMAT_SKIP)
9 : ディスクコピー時のフォーマットのエラーダイアログ	(FT_DLG_DCFORMAT_ERROR)
10 : フォーマットの確認ダイアログ	(FT_DLG_FORMAT_INPUT)
11 : フォーマットの確認ダイアログ	(FT_DLG_FORMAT_OK_CANCEL)
12 : フォーマットの確認ダイアログ (確認のみ)	(FT_DLG_FORMAT_OK)
13 : Windows ファイル 一覧表示ダイアログ (SelectWFile)	(FT_DLG_W_REFERENCE)
14 : IBM ファイル 一覧表示ダイアログ (SelectIFile)	(FT_DLG_I_REFERENCE)

●ダイアログボックスのタイプ

15 : はい、すべて変換、いいえ、キャンセル	(FT_DLG_ALLCANCEL)
16 : OK、すべて削除、いいえ、キャンセル	(FT_DLG_SKIPCANCEL)

●ダイアログボックスの終了コード

1006 : すべて変換、すべて削除	(FT_IDALL)
1008 : キャンセル	(FT_IDALLCANCEL)
1282 : スキップ	(FT_IDSKIP)

第3章

オートメーションサンプル

3. 1 Visual C++のサンプル

Visual C++からF*TRAN+のOLEオートメーション機能を使用したサンプルプログラムについて説明します。

このサンプルプログラムはVisual C++バージョン6以上で動作します。F*TRAN+インストールディレクトリの中にある¥Samples¥Ole¥Vc¥Olesmpl¥のワークスペースを開きます。

以下、サンプルソース重要部分の抜粋です。

```
// このサンプルプロジェクトの作成手順 (Visual C++ 6.0)
```

```
// このプロジェクトでは、F*TRAN+ のマンマシンインターフェイスを完全に隠した形  
// で変換処理を行うプログラムを作成しています。エラーメッセージも F*TRAN+ のも  
// のをマスクして表示しているため、ユーザーは F*TRAN+ の存在を意識することがあ  
// りません。
```

```
// このプロジェクトは以下の手順で作成されました。
```

```
// ① Microsoft Developer Studio を起動する。
```

```
// ②「ファイル」メニューから「新規作成」をクリックする。
```

```
// ③「プロジェクト」タブから「MFC AppWizard (exe)」を選択する。
```

```
// ④「プロジェクト名」に「OLESAMPL」と入力し「OK」を押す。
```

```
// ⑤「ダイアログベース」ラジオボタンをクリックし「次へ」を押す。
```

```
// ⑥「オートメーション」チェックボックスを有効にし「次へ」を押す。
```

```
// ⑦「次へ」「終了」「OK」の順に押し、プロジェクトを新規作成する。
```

```
// ⑧「表示」メニューから「ClassWizard」をクリックする。
```

```
// ⑨「メッセージマップ」タブから「クラスの追加」を押す。
```

```
// ⑩「タイプライブラリから…」をクリックし、提供テーブルファイル(FTWIN.TLB)
```

```
// をインポートする。
```

```
// ⑪「クラスの確認」ダイアログで、生成したいクラスを選択し「OK」を押す(ここ  
// では「IFtCmd」のみ)。
```

```
// ⑫「OK」を押し MFC ClassWizard を閉じる。
```

```
// ⑬ ClassWizard によって自動生成されたヘッダファイル(Ftwin.h)をインクルード  
// する。
```

```
#include "ftwin.h"
```

```
// ⑭ OnOK() 内に以下のようにコードを実装する。
```

```
void COLESAMPLDlg::OnOK()
```

```
{  
    // ----- ここから -----  
  
    IftCmd FtCmd;  
    CString strCode;  
    int     retCode;  
  
    // サーバーオブジェクトを作成する  
    if (FtCmd.CreateDispatch("Ftplus.FtCmd", NULL) == 0)  
    {  
        // エラー  
        MessageBox("FtCmd.CreateDispatch にてエラーが発生しました。");  
        return;  
    }  
  
    // 変換コマンド(ここでは"getdata")において、変換元・先ファイルを絶対パスで  
    // 指定しない場合は、カレントディレクトリをあらかじめ指定しておく必要がある  
    if ((retCode = FtCmd.SetCurrentDirectory("C:¥¥FTRANP")) != 0)  
    {  
        // サーバーオブジェクトへの接続を開放する  
        FtCmd.ReleaseDispatch();  
  
        // エラーコードを表示して戻る  
        strCode.Format("FtCmd.SetCurrentDirectory にてエラーが発生しました。 ¥n エラー  
番号: %d", retCode);  
        MessageBox(strCode);  
        return;  
    }  
}
```

```
// ここではパラメータファイルのパスが省略されているので、パラメータファイルを
// F*TRAN+ と同じフォルダに格納する必要がある
if (FtCmd.Command("/nwd /wc/ getdata PLANET PLANET.GET ++PNGETPRN.P") != 0)
{
    // エラー：提供ファイル FtDef.h に定義されている終了コードを取得する
    retCode = FtCmd.GetLastFtErrorCode();

    // サーバーオブジェクトへの接続を開放する
    FtCmd.ReleaseDispatch();

    // エラーコードを表示して戻る
    strCode.Format("FtCmd.Command にてエラーが発生しました。 ¥n エラー番号： %d",
retCode);
    MessageBox(strCode);
    return;
}

// サーバーオブジェクトへの接続を開放する
FtCmd.ReleaseDispatch();

// 変換結果をメモ帳で表示し、自分は終了する
SetCurrentDirectory("C:¥¥FTRANP");
WinExec("NOTEPAD.EXE PLANET.GET", SW_SHOW);

// ----- ここまで ----- //

if (CanExit())
    CDialog::OnOK();
}

// ⑮ 必要に応じてリソースを編集する(ここではメインダイアログに F*TRAN+ のアイ
//     コンを貼り付けている)。
// ⑯ 「ビルド」メニューから「リビルド」をクリックし EXE を作成する。
```

3. 2 Visual Basicのサンプル

Visual BasicからF*TRAN+のOLEオートメーション機能を使用したサンプルプログラムについて説明します。

このサンプルプログラムはVisual Basicバージョン6以上で動作します。F*TRAN+をインストールした¥Samples¥Ole¥Vb¥ディレクトリの中につきファイルがありますので、プロジェクトを開いてください。

Project1. vbp

以下、サンプルソースです。

```
'
'
' このサンプルプログラムは Visual Basic 6.0 以上で動作します。
'
' ----- このサンプルプロジェクトの作成手順 -----
'
' ① Visual Basic を起動する。
' ② 「ファイル」メニューの「新しいプロジェクト」をクリックして、
'   「新規作成」タブの「ActiveX EXE」をダブルクリックする。
' ③ 「プロジェクト」メニューから「標準モジュールの追加」をクリックして、
'   「新規作成」タブの「標準モジュール」を開く。
' ④ 「プロジェクト」メニューから「フォームモジュールの追加」をクリックして、
'   「新規作成」タブの「フォームモジュール」を開く。
' ⑤ 「プロジェクト」メニューから「Project1のプロパティ」をクリックして、
'   「プロジェクトプロパティ」ダイアログボックスの「全般」タブの中の
'   「スタートアップの設定」を「Sub Main」にする。
' ⑥ 次に「コンポーネント」タブをクリックして、「スタートモード」を
'   「独立型」にして、「OK」を押す。
' ⑦ 「プロジェクト」メニューから「参照設定」をクリックして「参照」ボタンを押し、
'   F*TRAN の提供テーブルファイル(Ftwin.tlb)を開く。
'   リストボックスに「Ftwin」が表示され、チェックが着いていることを確かめて、
'   「OK」ボタンを押す。
' ⑧ クラスモジュールのプロパティの「Instancing」が「5-MultiUse」に
'   なっているか確かめる。
' ⑨ 必要に応じてフォームを編集する(ここではForm1 にボタンを2つ追加している)。
' ⑩ 以下のようにコードを実装し、「ファイル」メニューの「Project1.exe の作成」を
'   クリックする。
```

```
' ----- 標準モジュール -----
'
```

```
Option Explicit
Sub Main()
    Form1.Show
End Sub
```

```
' ----- フォームモジュール -----
'
```

```
Option Explicit
Dim FtCmd As FtwIn.FtCmd
' 変換が正常に終了したら、その内容を表示する
'
```

```
Private Sub Command1_Click()
    Dim ret As Long
    Dim command As String
    Dim msg As String
```

```
' 変換コマンド(ここでは"getdata")において、変換元・先ファイルを絶対パスで
' 指定しない場合は、カレントディレクトリをあらかじめ指定しておく必要がある
```

```
ChDrive "c:"
ChDir "c:¥FTRANP"
ret = FtCmd.SetCurrentDirectory("c:¥FTRANP")
```

```
' エラー処理
If ret <> 0 Then
    msg = "FtCmd.SetCurrentDirectoryにてエラーが発生しました。" _
        & Chr(13) & "エラー番号:" & ret
    MsgBox msg, vbOKOnly, "F*TRAN+"
Exit Sub
End If
```

```
' ここではパラメータファイルのパスが省略されているので、パラメータファイルを
' F*TRANと同じフォルダに格納する必要がある
```

```
command = "/nwd /wc/ getdata Planet *.txt ++pngetprn.p"
ret = FtCmd.command(command)
```

’ エラー処理

If ret <> 0 Then

’ 提供ファイル FtDef.h に定義されている終了コードを取得する

ret = FtCmd.GetLastFtErrorCode

msg = "FtCmd. command にてエラーが発生しました。" _

& Chr(13) & "エラー番号：" & ret

MsgBox msg, vbOKOnly, "F*TRAN+"

End If

’ 変換結果をメモ帳で表示

Shell "notepad.exe Planet.txt", vbNormalFocus

End Sub

Private Sub Command2_Click()

’ サーバーオブジェクトへの接続を開放する

Set FtCmd = Nothing

End

End Sub

Private Sub Form_Load()

’ サーバーオブジェクトを作成する

Set FtCmd = New FtwIn.FtCmd

End Sub

Private Sub Form_Unload(Cancel As Integer)

’ サーバーオブジェクトへの接続を開放する

Set FtCmd = Nothing

End Sub

F * T R A N + V 6 . 0 操作説明書／プログラム応用編

2 0 1 0 年 4 月 第 1 版発行

編集・著作 株式会社 富士通ビー・エス・シー

所 在 地 〒135-8300 東京都港区台場 2-3-1 トレードピアお台場

- ・ Windows、MS-DOS、Visual Basic、Access、Visual C++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。
- ・ Unicode は Unicode コンソーシアムの商標です。
- ・ F*TRAN は富士通ビー・エス・シーの登録商標です。
- ・ 会社名および製品名はそれぞれ各社の商標または登録商標です。
- ・ 本書およびシステムは、改善のため事前連絡なしに変更することがあります。
- ・ 無断複製、および転載を禁じます。