

ポケコン用 RAM ファイルシステム



## Version 4.2

# Users Manual (Rev.8)

本ドキュメントは、  
FX-870P/VX-4 のプログラム(BASIC/機械語)経験者向けの資料です。

ソフトウェア名称	FX-Binary-Filer File Control SubSystem Ver.4.2
ジャンル	バイナリーファイル用 RAM ファイルユーティリティ
著作権者	©CopyRight by JUN AMANO 2003-2024
動作環境	FX-870P / VX-4 RAM 合計 16KB 以上
必須環境	パーソナルコンピュータでシリアルポート経由で通信できる環境
転載可否	可能です。巻末の「12. ご利用に関して」をご覧ください。
問い合わせ先	VYU04421@nifty.com

## 目次

1.	はじめに.....	3
2.	動作環境と特徴.....	4
3.	FBF のインストール.....	5
4.	FBF のアンインストール.....	7
5.	FBF を試してみる.....	7
6.	FBF のしくみ(登録ファイルについて).....	8
7.	FBF の呼び出し方法.....	9
8.	FBF の呼び出し種別.....	10
9.	FBF の呼び出し形態.....	11
10.	機能概要説明.....	12
11.	サンプルについて.....	17
12.	最後に～ 「VX-MENU との違い」と、「FBF が目指すもの」...	19

## 1. はじめに

「FX Binary Filer –File Control Subsystem」(FBF)は、RAM 上のメモリ空間の内容をファイルとして保存し管理できるバイナリーデータ専用の RAM ファイルシステムです。複数の機械語アプリケーションを保存し、必要に応じてメモリにロードして実行したり、グラフィックデータ等のデータをファイルとして保存するなど、従来では難しかった作業が行えるようになりました。

こちらのマニュアルでは、FBF 単体のインストール方法、ワークエリア指定での実行方法など、プログラム作成者向けの利用方法を記載しております。

注意:FBF 単体だけでは何もできません。FBF を利用するプログラム向けとなっております。

本ドキュメントでは以下の名称を略称で記載しています。

名称	本ドキュメントでの略称
FX-870P または VX-4	FX-870P
FX-Binary-Filer File Control SubSystem	FBF
F.COM(改) VX-MENU システム (PB-1000 MENU SIMULATOR)	VX-MENU

## 2. 動作環境と特徴

1. FBF 機能一覧	
(1)	任意のメモリ(アドレス)空間を「ファイル」として登録します。
(2)	登録されたファイルを指定したメモリ空間(アドレス)へロードできます。
(3)	ファイルの名前変更・削除・保護等のファイルに関する管理操作が可能です。
(4)	「RR(Relocate & Run)形式プログラム」機能により、 機械語エリアを確保することなく、メモリ空間のどこでも実行できます。
(5)	FBF を任意のアドレスにインストール出来る「擬似リロケータブル方式」を採用。
(6)	255 個又はメモリの許すまでファイルを登録出来ます。
(7)	BASIC、機械語より FBF を操作可能です。 通常呼び出し(JP)またはサブルーチン(CAL)呼び出し可能。
(8)	FDD バッファ上にパラメータを書き込み CALL を行う事でファイル操作 を行います。
(9)	BASIC から簡単にファイル操作できるコマンドラインインターフェイスを装備。

2. FBF の動作環境	
対応機種	FX-870P または VX-4
必要メモリー	RAM 合計 16KB 以上
通信環境の必要性	必須

### 3. FBF のインストール

FBF は、最上位(最終)アドレスにシステムが参照できないメモリ領域を確保し、常駐します。  
最上位アドレスに機械語領域を確保する方法は、FX-870P では一般的なために、  
FBF が常駐してしまうと、同じアドレスで実行する機械語プログラムが使用できなくなってしまいます。  
そこであらかじめ使用する領域が判明している場合は、その領域より下位のアドレスに  
インストールすることができます。

図1) 既定のインストール

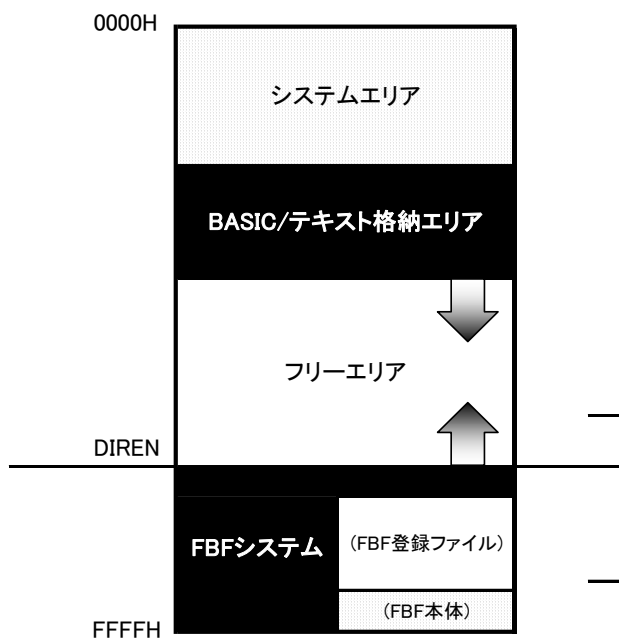
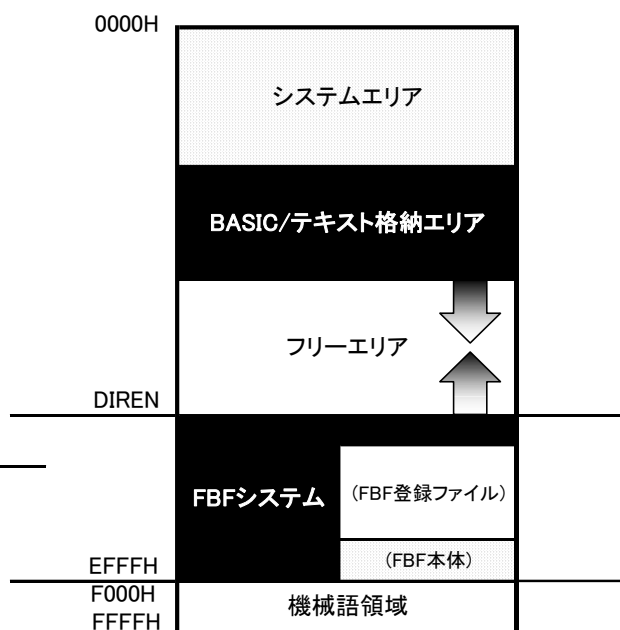


図2) 最上位に機械語領域を確保する場合



※FFFFH は 64KB 増設時の最上位(最終)アドレスとなります。

※FX-870P のシステムはポインタ DIREN(&H18D1,&H18D2)の示すアドレス以降の  
メモリを参照しません。

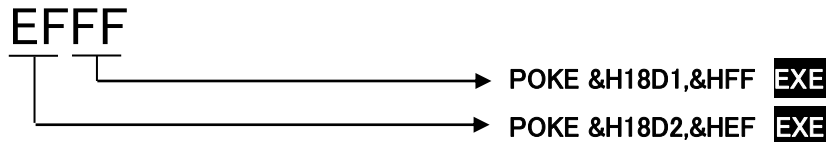
※FBF をインストールした後で、FBF システム領域のアドレス変更はできません。

インストール手順は以下のように行います。

- (1) 前ページの図2のように最上位アドレスに機械語領域を確保する場合は、

確保する **【機械語領域の先頭アドレス - 1】** を DIREN ポインタに設定します。

例) 先頭アドレスが &HF000 の場合 は【&HF000 - 1】 → 【&HEFFF】を書きこみます。



- (2) 「02\_Installer」フォルダにある、「FBFInst.b」を任意のプログラムエリアへ転送して、実行してください。

※インストーラーは 16KB～64KB 共通です。

- (3) 以下の画面が表示されたら、PC から「FBF\_v41rbf」を転送してください。

転送中は矢印が回転します。

CONNECT:



- (4) 転送が終了すると、以下のようにインストールしたアドレス情報を表示して、完了となります。

FBF Installed!  
Start:F9B0H End:FFFFH



## 4. FBF のアンインストール

FBF をアンインストールすると、FBF システムで使用していたメモリを解放して、インストール前の状態に戻ります。登録されたファイルは全て削除されます。

アンインストールするには以下の手順を実施します。

※同じ手順を二度実行すると、暴走しますので注意してください。

- (1) POKE &H1793,127 **EXE**
- (2) MODE110(PEEK(&H18D2)\*256+PEEK(&H18D1)+1) **EXE**

<< FBF-FMxx.BAS を実行する場合 >>

- (1) RUN 10000 **EXE**
- (2) 以下の問い合わせが出ますので YES と半角英大文字で入力して、**EXE**を押します。

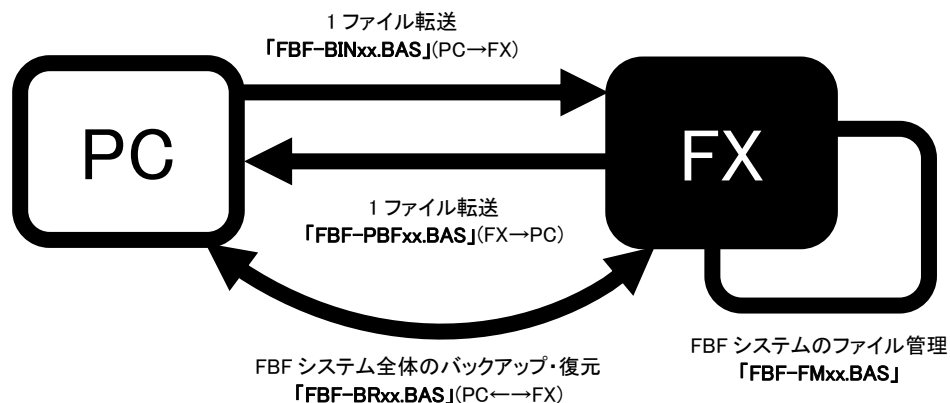
FBF release/file all delete  
OK?(YES=release)

- (3) 「FBF release successfully.」と表示され、アンインストールが完了します。

## 5. FBF を使ってみる

この FBF パッケージにはすぐに活用できる管理ソフトウェアが同梱されております。是非ご利用ください。

(使用方法は各ソフトウェアのマニュアルをご覧ください)



## 6. FBF のしくみ(登録ファイルについて)

FBF では登録されたファイルを「ファイル名」で区別して管理しています。

**【ファイル名】8文字までのキャラクター**

FBF は MS-DOS 等と違い、拡張子の概念がありません。

例えば、ファイル名が「A. EXE」の場合、区切り文字の「.」(ピリオド)も1文字としてカウントされます。

またコントロールコード含む全てのキャラクターを利用可能ですが、

特殊なキャラクターを利用すると、利用プログラム側で表示が崩れたりする場合がありますので、使用しないでください。

尚、以下は特殊な意味を持ちます。

規則	役割	備考
ファイル名に「. R」を含む	<< RR形式プログラム >>	
ファイル名に「. r」を含む	<< RR形式プログラム(CAL形式呼び出し) >>	機械語からのみ利用可能
ファイルの先頭に「\$」がある	<< 削除不可、ファイル位置移動可能 >>	
2文字の名前+「. R」	<< RR形式プログラムの短縮実行形式 >> 例) MODE110(BASE)RR"CL.R", 500 [EXE] は MODE110(BASE)CL,500 [EXE] と同じです。	(1) "PI""OR"等、 BASICと同じ予約語と同じ名前は 使用できません。エラーとなります。
「\$」+2文字の名前+「. R」	<< RR形式プログラムの短縮実行形式 >> (削除不可、ファイル位置移動可能) 例) MODE110(BASE)RR"\$CL.R", 500 [EXE] は MODE110(BASE)\$CL,500 [EXE] と同じです。	(2) "BS"等、内蔵コマンドと同じ場合は 内蔵コマンドが優先されます。

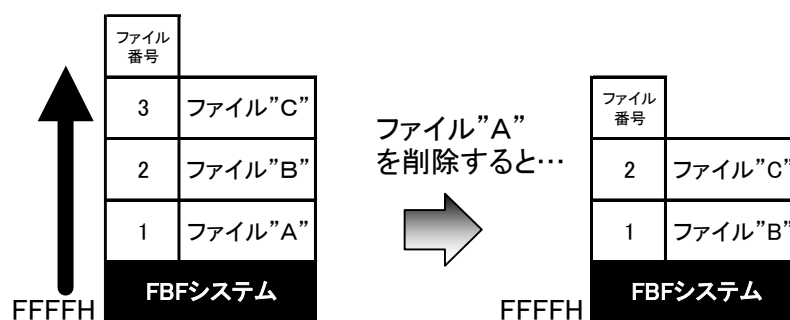
以下のようにファイル名を運用すると分かりやすいでしょう。

拡張子	意味
6文字+「. E」	実行ファイル(EXE)
6文字+「. B」	バイナリーデータ(BIN)
6文字+「. D」	バイナリーデータ(DAT)

FBF のファイルはメモリ最上位(FFFFH)から下位(0000H)に向かって登録されていきます。

ファイル番号は「何個目に登録されているファイル」を示すもので、FBF システムに近い方から順番に振られてきます。ファイル番号で指定する機能があります。

以下の図のようにファイルを削除すると、詰め処理によりファイル番号も変わりますので注意してください。



ファイルはメモリ下位アドレス(先頭)に向かって登録されます。



## 7. FBF の呼び出し方法

FBF は、MODE110 文(機械語実行)にて呼び出します。

呼び出しアドレスは常に「システムエリアの DIREN ポインタの示すアドレス+1」です。

```
A=PEEK(&H18D2)*256+PEEK(&H18D1)+1
MODE110(A)
```

DIREN ポインタの示すアドレスは FBF のファイル登録情報によって変化し、常に計算する必要があるため、以下のようにすると一度だけの計算で FBF を利用できるため、大変便利です。この方法は VX-MENU と互換性があり推奨手順です。

```
A=(PEEK&H18D1+PEEK&H18D2*256)+2:BASE=PEEK(A)+PEEK(A+1)*256
MODE110(BASE)
```

また、FBF には起動プログラムをシステムエリアへコピー(※)する機能があります。この機能を利用すると、固定アドレスでの呼び出しが可能です。

```
MODE110(BASE)JP 実施以後、
MODE110(&H1AD7)
```

※システムスタックポインタの最後尾 3 バイトに常駐します。

この領域は通常使用下では FX-870P システムではまず利用されていません。

※念のため、プログラムの先頭に JP コマンドでコピーを行うようにしてください。

## 8. FBF の呼び出し種別

FBF には、以下の 2 通りの呼び出し種別が存在します。

BASIC からの場合は通常はコマンドライン指定で FBF を利用します。

機械語からの場合はワークエリア指定で FBF を実行します。

ワークエリア指定でしか実行できない機能もあります。

ワークエリア実行の場合は、指定アドレス(&H1793)に「機能番号」(どの機能を利用するか)と必要なパラメーターを指定して、FBF をコールします。

※ファイル名の指定が必要な場合は、ファイル名が8文字満たない場合、残りは「0」で埋める必要があります。

種別	意味	例	BASIC より利用	機械語 より利用
ワークエリア実行	指定アドレスに【機能番号】や必要データを POKE文にて書きこみ、引数無しで実行する。	・POKE&H1793,9 ・MODE110(BASE)	○	○
コマンドライン実行	MODE110文に 引数(コマンド)を記載して実行する。	・MODE110(BASE)BS"TEST",&H100,124	○	×

## 9. FBF の呼び出し形態

FBF は通常実行後に FX-870P のシステムに戻ります。

以下の場合、MODE110 文実行後は続いて、次の命令(PRINT 文)が実行されます。

**10 MODE110(BASE)SI:PRINT"HELLO WORLD!"**

機械語からの実行の場合はファイルを検索して、その結果を利用する場合など、実行後は利用プログラムに処理を戻したい場合があります。

このサブルーチ的な使い方を為す為に FBF では 2 通りの呼び出し形態があります。

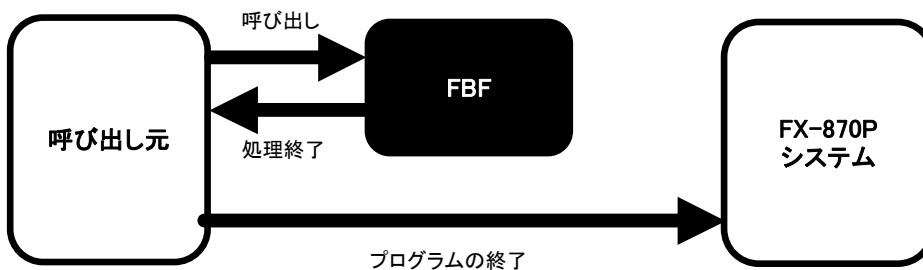
種別	意味	BASIC より利用	機械語 より利用
通常実行	FBF実行後、FX-870Pシステムに制御が戻る	○	○
CAL実行	FBF実行後、呼び出し元のプログラムへ制御が戻る。 【機能番号+128】を機能番号として指定してワークエリア実行する。	×	○

BASIC で例えると、通常実行は「GOTO」、CAL 実行は「GOSUB」になります。

### 通常実行イメージ



### CAL実行イメージ



通常実行と CAL 実行のテンプレートが「Temples」フォルダーに用意されています。

テンプレートの利用が推奨されています。

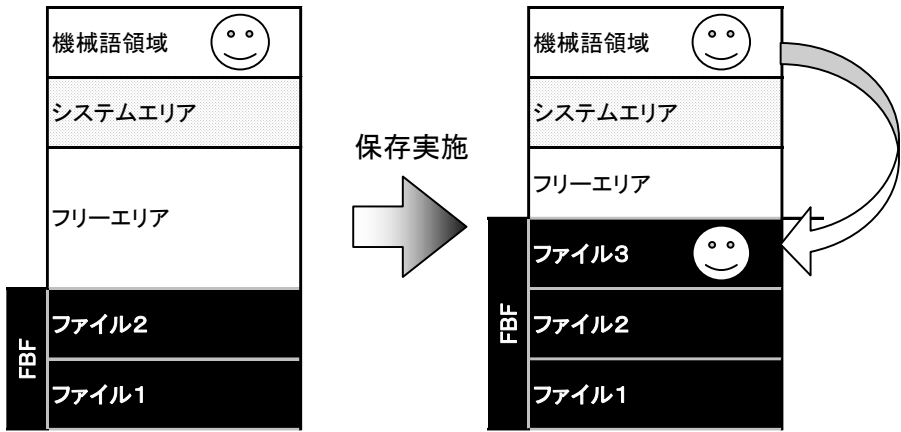
機械語から CAL 実行を行う場合は、以下の条件が必要です。

- ① 機能番号に 128 を足したものを &H1793 に書き込む。
- ② インデックスレジスター(IX)が示すアドレスの内容を 0 にする。
- ③ DIREN が示すアドレス+1 を CAL する。

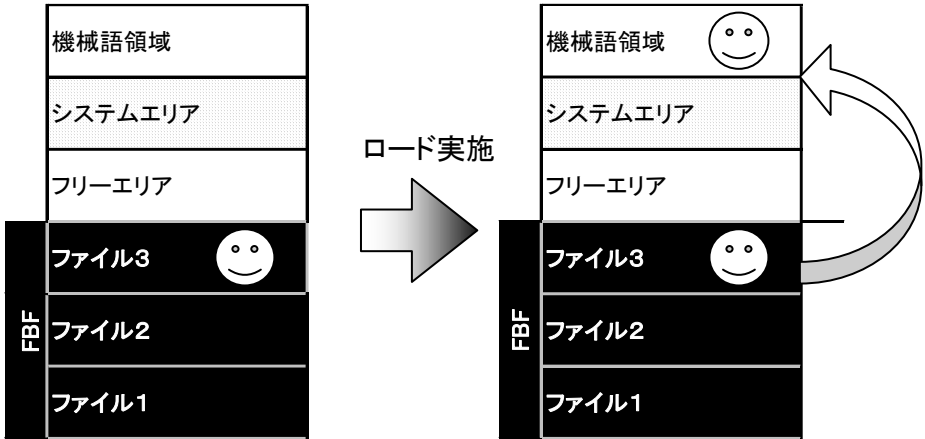
## 10. 機能概要説明

機能の概要を説明します。具体的なパラメーターの指定方法などは別紙をご覧ください。

機能	ファイル番号検索
機能番号	0
コマンド	ありません
説明	<ul style="list-style-type: none"> <li>・ファイル番号を指定してファイル名を検索する機能です。</li> <li>・ファイル一覧を作成したりする場合に利用します。</li> <li>・検索に成功すると、通常実行時に限り、以下の BASIC 変数に値が代入されます。</li> </ul> 【FN\$】 ファイル名 / 【ST】 開始アドレス / 【SZ】 サイズ 【EX】 実行アドレス / 【SO】 収納アドレス

機能	ファイルの保存
機能番号	1
コマンド	BS (“BSAVE”の略)
説明	<ul style="list-style-type: none"> <li>・指定したメモリ空間の内容をファイルとして保存します。</li> <li>・保存ファイル名に既存のファイルを指定すると、そのファイルは消去され、新たに保存されます(上書き保存)。</li> <li>・先頭に「\$マーク」(削除禁止/移動可能)が付いている既存ファイルに対しては上書き保存できません。</li> <li>・ファイルは常に FBF エリアの最後尾に保存されます。</li> </ul> <p>先頭に「\$」が付いたファイルを保存する場合、別途 MV コマンドを実行して FBF エリアの先頭に移動してください。</p> 

機能	ファイルの削除
機能番号	2
コマンド	KL (“KILL”の略)
説明	<ul style="list-style-type: none"> <li>・指定したファイルを削除します。</li> <li>・ファイルを削除すると、メモリの「詰め処理」が行われます。従って、削除後にファイルの復活は出来ません。</li> <li>・先頭に「\$マーク」(削除禁止/移動可能)が付いている既存ファイルに対しては削除できません。</li> </ul>

機能	ファイルのロード
機能番号	3(保存時アドレスへのロード) / 4(指定アドレスへのロード)
コマンド	BL (“BLOAD”の略)
説明	<ul style="list-style-type: none"> <li>・指定したファイルをロードします。</li> <li>・システムエリア・FBF エリア等にロードすると暴走の原因となりますのでご注意ください。</li> </ul> 

機能	ファイルの名称変更
機能番号	5
コマンド	RN (“RENAME”の略)
説明	<ul style="list-style-type: none"> <li>・指定したファイルの名称を変更します(ファイルのリネーム)。</li> <li>・「新しいファイル名」に既存のファイルを指定すると、処理失敗となります。</li> </ul>

機能	ファイル検索
機能番号	6
コマンド	SF (“Search File”の略)
説明	<p>・ファイル名を検索して、ファイルの存在を確認します。</p> <p>・検索に成功すると、通常実行時に限り、以下の BASIC 変数に値が代入されます。</p> <p>【FN\$】 ファイル名 / 【ST】 開始アドレス / 【SZ】 サイズ</p> <p>【EX】 実行アドレス / 【SO】 収納アドレス</p>

機能	情報取得
機能番号	7
コマンド	SI (“System Information”の略)
説明	<p>・FBF システム情報を取得します。</p> <p>・ファイルの最大登録数を取得したい時はこの機能を実行してください。</p>

機能	ジャンプテーブルコピー
機能番号	8
コマンド	JP (“Jump Table”の略)
説明	<p>・FBF の実行プログラムをシステムエリアへコピーします。</p> <p>(システムスタックの最後尾 3 バイトへ常駐)</p> <p>これにより、MODE110(&amp;H1AD7) で実行できるようになります。</p>

機能	ワークエリアクリア
機能番号	9
コマンド	ありません。
説明	<p>・FBF が使用するシステムワークエリア(FDD バッファ)をゼロクリアします。</p> <p>・ワークエリア指定でパラメータを書き込む前に、必ず実行してください。</p>

機能	RR 形式実行①
機能番号	10
コマンド	RR (“Relocate & Run” の略)
説明	<p>・RR 形式のプログラムを実行します。</p> <p>・登録ファイル内でファイルの最後尾に記載されているリロケートデータに基づきプログラムの変更を行い実行します。</p>

機能	RR 形式実行②
機能番号	11
コマンド	RL (“Relocate Load & Run” の略)
説明	<p>・RR 形式のプログラムを実行します。</p> <p>・指定アドレスへロードした後に、RR 形式実行①と同様にプログラムの書き換えを行い、実行します。</p> <p>・RR 形式実行①の場合、実行のたびにプログラム書き換え処理が発生しますが、RL の場合は一度実行してまえば、以後指定アドレスでの実行が可能となります。</p> <p>例) 1 回目 MODE110(BASE)RL”TEST”,&amp;H0 ←0 番地にロードして実行 2 回目以降 MODE110(0)</p> <p>・RR 形式実行②の場合、元のファイルの内容が保たれます。</p>

機能	ファイル移動処理
機能番号	12
コマンド	MV (“Move”の略)
説明	<p>・指定したファイルを FBF のファイル登録エリアの先頭に移動します。</p> <p>・先頭に「\$マーク」(削除禁止/移動可能)が付いているファイルのみ移動できます。</p> <p>・<u>サイズが大きいファイルの場合、移動に時間が掛かります(5分等)。</u></p> <p><u>このような場合であっても、リセットしないでお待ちください。</u></p> <p>・FBF ではファイルを削除した場合は「メモリの詰め処理」が行われます。プログラムから別なプログラムを呼び出し、そのプログラムからファイル削除を実施すると、詰め処理により呼び出し元のプログラムのアドレスが変化してしまい、暴走の原因になる場合があります。</p> <p>このような場合のために本機能が必要となります。</p> <div style="text-align: center;"> </div>

機能	FBF 切り離し
機能番号	127
コマンド	OF (“OFF”の略)
説明	<p>・FBF を切り離して、使用していたメモリをフリーエリアに解放します。</p> <p>・本機能を実行すると、全ファイルが消去されます。</p> <p>・本機能を実行した後は、FBF 対応プログラムを実行すると、暴走しますので注意してください。</p>



## 11. サンプルについて

このパッケージには以下のフォルダにサンプルプログラムを収録しています。

**“FBF\_Samples”フォルダ**

- ・FILEFIND.BAS --- ファイル名を入力して検索を行うプログラム
- ・FILELIST.BAS --- ファイル名一覧を表示するプログラム
- ・GRFLOAD.ASM / GRAPHIC.PBF
  - 機械語で CAL 呼び出しを利用したサンプルプログラム  
(GRFLOAD.ASM をアセンブルし、GRAPHIC.PBF と共に  
ポケコンに転送してから実行してください)

以下のフォルダに機械語から利用するためのテンプレートがありますので、  
利用してください。

**“FBF\_Templates”フォルダ**

- ・FBFJP.ASM --- 通常呼び出しを行う際に使用するテンプレート
- ・FBFCAL.ASM --- CAL 呼び出しを行う際に使用するテンプレート

## 12. 最後に～

### 「VX-MENU との違い」と、「FBF が目指すもの」

FX-870P にはファイルシステムにして「FBF」「VX-MENU」の 2 種類のファイルシステムが登場しました。この2種類の違いは何でしょうか。

「VX-MENU」は、従来の FX-870P のシステム(P0～9,F0～9)を「置き換える」意味で登場しました。そのため、ファイル管理機能や通信・BASIC・テキスト編集機能など全て含んだ「統合環境」となっています。

「FBF」は、コマンドラインや RR 形式実行の採用など FX-870P の環境をパワーアップする目的で作成されました。また、インストールするアドレスを変更してメモリ最上位アドレスを機械語領域として確保できるなど、従来資産の互換性も考慮されています。

また、ファイルシステムのための最小限の構成で、ゲームなどに組み込んで利用するケースも想定されています。

NEC PC98 時代にフロッピーディスクのゲームを FD ドライブのに入れて起動すると「MEG-DOS」が起動して、ゲームが始まる・・・の MEG-DOS みたいな存在が理想(笑)です。

ぜひ、作成されたプログラムに是非 FBF をご活用いただければと思います。

