

Factory's FileMaker Plug-in for Windows Manual

制作／著作

Factory's FileMaker Plug-in Manual for Windows

ver.1.1 (2004/7)

Copyright 2001~2004

Factory, Inc. All rights reserved.

有限会社ファクトリー

東京都世田谷区三軒茶屋 2-13-20-1004 (154-0024)

Tel.03-5486-7370 Fax.03-5486-6543

support@factory-1987.co.jp

<http://www.factory-1987.co.jp>

商標

本マニュアルに記載されております製品名および社名は、各社の商標あるいは登録商標です。

目次

Factory's FileMaker Plug-in 使用許諾契約書	6
はじめに	8
Factory's FileMaker Plug-in の特徴	8
Factory's FileMaker Plug-in を理解するために	9
プログラムのインストールと設定	10
インストールの手順（オンライン版の場合）	10
インストールの手順（パッケージ版の場合）	10
必要システム構成	11
QuickTime の設定	11
プログラムのユーザ登録とシリアル番号	12
デモ版として使用する場合	12
製品版として使用する場合	12
ユーザ登録	12
ユーザサポート	13
必要事項	13
サイトライセンスパッケージについて	14
ランタイムアプリケーションでの利用について	15
Factory's FileMaker Plug-in の使い方	16
書式について	17
引数について	18
Macintosh と Windows が混在する環境	20
返値について	21
コマンドが正常に動作した場合の実行結果	21
コマンドが正常に動作しなかった場合の実行結果	21
正常に動作した結果として返されたエラー	22
返値からエラー番号を調べる方法	22
値の形式について	23
基本的な値の形式	23

それ以外の値の形式	24
エスケープ・シーケンスについて	27
タグ付きフォーマットについて	28
Image リファレンス	29
Flmg-Version	30
Flmg-DrawImage	31
Flmg-GrabImage	33
Flmg-SaveImage	34
Flmg-DrawImage, Flmg-GrabImage, Flmg-SaveImage 共通のオプション引数	35
Flmg-DrawImage, Flmg-GrabImage, Flmg-SaveImage 複数コマンドの一括指定	37
Flmg-ImageInfo	38
Flmg-PreviewImage	40
Flmg-CompressImage	41
Flmg-CompressImageDialog	42
Flmg-CheckArea	43
Flmg-Effect	44
Flmg-GetMousePos	45
Utility リファレンス	46
FUtl-Version	47
FUtl-Tag	48
FUtl-TagList	49
FUtl-OpenFileDialog	50
FUtl-OpenFolderDialog	51
FUtl-SaveFileDialog	52
FUtl-FindFile (最初の検索)	53
FUtl-FindFile (繰り返し検索)	54
FUtl-VolumeCount	55
FUtl-GetVolume	56
FUtl-FolderItemCount	57
FUtl-GetFolderItem	58
FUtl-FileInfo (基本情報)	59
FUtl-FileInfo (詳細情報)	62
FUtl-SetFileInfo	64
FUtl-CopyFile	66

FUtl-MoveFile	67
FUtl-NewFolder	68
FUtl-DeleteFile	69
FUtl-EjectVolume	70
FUtl-LaunchApp（アプリケーション）	71
FUtl-LaunchApp（書類）	72
FUtl-QuitApp	73
FUtl-PickColor	74
FUtl-CalcColor	75
FUtl-CalcPoint	78
FUtl-CalcRect	80
FUtl-OpenProgress	85
FUtl-UpdateProgress	88
FUtl-CloseProgress	89
FUtl-Answer	90
FUtl-Ask	92
 付録 1：コマンド一覧	 94
Image	94
Utility	94
 付録 2：描画モード一覧	 95

Factory's FileMaker Plug-in 使用許諾契約書

本許諾契約書は、有限会社ファクトリー（以下「乙」という）がFactory's FileMaker Plug-in（以下「本ソフトウェア」という）をソフトウェア使用者（以下「甲」という）に対し、使用許諾を与えるに際しての法的な条項を述べています。

本ソフトウェアを使用される場合は、ユーザ登録の有無に関わらず以下の条項のすべてに同意／承諾したものと見なされます。同意／承諾できない場合は、乙よりご返金いたしますので、お買いあげ後7日以内にご購入を証明するもの（領収書やレシートなど）を添え、弊社までご返送願います。また、その際に発生する送料等は甲が負担するものとします。

* オンライン版をご購入の場合は、本ソフトウェアに関するすべてのファイルを削除してください。

第一条（目的）

1. 乙は甲に対し、乙が著作権を有し販売する本ソフトウェアを使用するに当たり、本許諾契約書への同意を確認するものです。

第二条（許可）

1. 本ソフトウェアは、登録されたユーザに限りインストールおよび使用することができます。
2. 本ソフトウェアは、一つのコンピュータ上でのみ使用することができます。

第三条（サポート）

1. 乙はユーザ登録を行った甲に対してのみ、ユーザサポートおよびアップグレードなどのサポート業務を行います。
2. アップグレードなど通知およびアップデートファイル等は、基本的に弊社Webおよびインターネットメールを通じて行います。

第四条（著作権と禁止）

1. 本ソフトウェアおよびマニュアルを含め、パッケージに同梱される一切のドキュメントの著作権は、すべて乙に帰属します。
2. 本ソフトウェアおよびマニュアルの一部または全部を転載、複製する事を禁止します。
3. 本ソフトウェアを逆コンパイル、リバースエンジニア、逆アセンブル、修正など本ソフトウェアに対して不利益となる一切の行為を禁止します。
4. 乙の許可無く本ソフトウェアを、貸与またはリースすることを禁止します。
5. 本ソフトウェアの使用範囲は、日本国内に限定されます。

第五条（譲渡）

1. 乙の許可無く本ソフトウェアを譲渡することを禁止します。
2. 乙の許可を受けて本ソフトウェアを譲り受けた者は、本許諾書の条項を無条件で受け入れたことと見なします。

第六条（免責および責任の限度）

1. 乙は本ソフトウェアを使用することによって直接、あるいは間接的に発生したいかなる損害に対しても一切の責任を負いません。
2. 特例として、損害がどのような種類あるいはどのような規模であった場合でも、乙の賠償責任が本ソフトウェアのご購入金額を超えることはありません。

第七条（契約違反）

1. 甲が本契約およびこれに伴う取り決めに違反した場合、甲は乙が提示する期間内に違反状態を解消しなければなりません。

本契約についてのご質問は、弊社までお問い合わせください。

作成日：2004年7月

製品の内容については万全を期しておりますが、お不審な点や誤り、記載漏れなどがありましたら弊社までご連絡願います。

はじめに

この度はFactory's FileMaker Plug-inをご購入いただき、誠にありがとうございます。

Factory's FileMaker Plug-inは、ファイルメーカー社からリリースされているファイルメーカーProの機能を拡張するプラグインです。

外部にある大量の画像ファイルを、ファイルメーカーProで効率よく管理するためのコマンドが用意されており、ファイルメーカーProに用意されたオブジェクトフィールドを使った画像管理よりも自由度の高いデータベースを構築／運用することが可能となります。

Factory's FileMaker Plug-inの特徴

Factory's FileMaker Plug-in は以下の特徴を持っています。

- ・ファイルのパス情報を利用することにより、実データをファイルメーカーPro内に持つことなく、大量の画像ファイルを効率よく管理することが可能です。

- ・ネットワークボリュームをサポート（ユニバーサルパス）します。

ユニバーサルパスとは、MacintoshとWindowsで共有できるパス形式で、弊社固有の方法で作成されるパス情報です。

- ・QuickTime 4.x を利用することにより、インターネットサーバ内の画像をダイレクトに表示することができます（パスの例：<http://www.xxx.co.jp/graphic/abc.jpg>）。

- ・QuickTime がサポートするファイルタイプの静止画像を表示することができます。

QuickTime 4.0 がインストールされている場合：BMP, FlashPix, GIF, JPEG/JFIF, MacPaint, Photoshop, PICT, Pictures, PNG, QuickTime Image, SGI, Targa, TIFF を表示することができます。

- ・QuickTime がサポートするファイルタイプで画像を保存することができます。

（QuickTime 4.0がインストールされている場合：BMP, JPEG/JFIF, MacPaint, Photoshop, PICT, Picture, PNG, QuickTime Image, SGI, Targa, TIFF）

ファイルメーカーPro内にある画像をファイルとして保存することや外部にある画像ファイルを保存しなおすことができます。また保存時にはQuickTimeのサポートする圧縮方法で圧縮を指定することができ、さらに、ファイルメーカーPro内にある画像を圧縮することも可能です。また、複数の画像を重ね合わせた状態で保存することもできるため、例えば、メインの画像の右下に固有のマークを付けて保存することなども可能です。

- ・指定領域をチェックして再表示のタイミングを判定することができます。

例えば、ウインドウに画像を表示した後で、その表示領域に別のウインドウなどがかぶってしまった場合、表示されていた画像が欠けてしまいます。この場合、もう一度画像を表示する必要がありますが、「指定領域のチェック」機能を使えば、この処理を自動化し再描画することができます。

- ・ファイル情報については、パス情報およびファイル名、種類、サイズ、作成日、修正日などのファイル情報を取得することができます。

- ・指定エリアに対し、拡大／縮小、回転、移動、重ね合わせ、トリミングなど様々な画像表示ができます。

- ・指定エリアに対する背景色が設定可能です。

- ・ファイルやフォルダ操作（作成／コピー／移動／削除／リネームなど）をサポートしています。
- ・フォルダダイアログで選択したフォルダ内の複数画像情報も一括して取得することができます。
- ・元の画像をスケーリングし、ファイルメーカー Pro のデータとして取り込むことができます。
例えば、プレビューイメージとして小さい画像を作成することができますが、もちろん、オリジナル画像データをそのまま取り込むこともできます。
- ・クリックした座標を取得することができるため、クリックした位置を拡大／縮小したり、クリックした位置に画像を描画するといった利用法も可能です。
- ・ファイルメーカー Pro から他のアプリケーションを起動することができます。
- ・マルチプラットフォーム対応。Macintosh や Windows が混在する環境でも利用することができます（それぞれの Plug-in が必要）。

Factory's FileMaker Plug-in を理解するために

Factory's FileMaker Plug-in を利用するには、ファイルメーカー Pro のスクリプトをある程度以上自由に使いこなせる知識が必要になります。

また、同梱のサンプルプログラムには Factory's FileMaker Plug-in を利用して実現できるいくつかの例題が含まれています。本マニュアルとあわせて解析していただければ、仕組みをご理解いただくことができるでしょう。

焦らず、じっくりと取り組んでください。

プログラムのインストールと設定

Factory's FileMaker Plug-in はファイルメーカー Pro 専用のプラグインです。

使用するにはファイルメーカーProの「System」フォルダにプラグインをインストールする必要があります。

インストールの手順（オンライン版の場合）

1. 「Plug-in_Folder」にある「Factory.fmx」を選択し、ファイルメーカー Pro と同階層にある「System」フォルダ（ファイルメーカーPro6まで）あるいは「Extensions」フォルダ（ファイルメーカーPro7から）にコピーします。
2. ファイルメーカー Pro を起動します。
3. プログラムのレジストレーションダイアログが表示されるので、必要事項を記入して「OK」ボタンをクリックします。シリアル番号の入力に関しては後述の「プログラムのユーザ登録とシリアル番号」の項をご覧ください。
4. 「編集」メニューの「環境設定」から「アプリケーション」を選択します。
5. 「アプリケーション環境設定」の設定画面にある「プラグイン」を選択します。
6. 現在ファイルメーカー Pro にインストールされているプラグインの一覧が表示されるので、「Factory's FileMaker Plug-in」の使用チェックボックスをクリックし、「使用」に設定します（場合によっては、既に使用がチェックされていることがあります）。

インストールの手順（パッケージ版の場合）

1. 圧縮された「Program_and_Samples」ファイルをダブルクリックし、表示されるダイアログ上で任意のディレクトリを指定して「Extract」ボタンをクリックします。
2. 解凍された「Program_and_Samples」フォルダをクリックして開き、フォルダ内にある「Plug-in_Folder」の中にある「Factory（あるいは「Factory」）」を選択します。
3. 「Plug-in_Folder」にある「Factory.fmx」を選択し、ファイルメーカー Pro と同階層にある「System」フォルダ（ファイルメーカーPro6まで）あるいは「Extensions」フォルダ（ファイルメーカーPro7から）にコピーします。
4. ファイルメーカー Pro を起動します。
5. プログラムのレジストレーションダイアログが表示されるので、必要事項を記入して「OK」ボタンをクリックします。シリアル番号の入力に関しては後述の「プログラムのユーザ登録とシリアル番号」の項をご覧ください。
6. 「編集」メニューの「環境設定」から「アプリケーション」を選択します。

7. 「アプリケーション環境設定」の設定画面にある「プラグイン」を選択します。

8. 現在ファイルメーカー Pro にインストールされているプラグインの一覧が表示されるので、「Factory's FileMaker Plug-in」の使用チェックボックスをクリックし、「使用」に設定します（場合によっては、既に使用がチェックされていることがあります）。

確認！！

プログラムの登録画面が表示されない場合は？

ご使用中の OS に、アップル社の QuickTime がインストールされているかを確認してください。

Factory's FileMaker Plug-in は、QuickTime を必須環境として動作します。もしも QuickTime がインストールされていない場合は、下記のアップル社のサイトよりダウンロードの上、インストールして下さい。

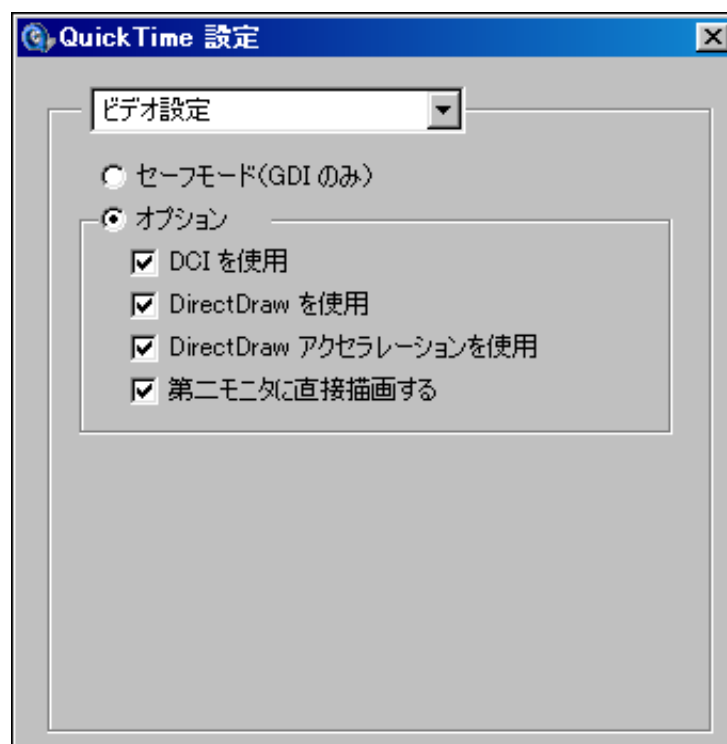
<http://www.apple.co.jp/quicktime/download/index.html>

必要システム構成

- ・ Windows 98, Me, NT, 2000, XP
- ・ 必要メモリ：構築するデータベースに応じて
- ・ ファイルメーカー Pro ver.4.x 以上
- ・ QuickTime 5.x 以上（サポートされるファイルタイプは、QuickTime のバージョンに依存します）
- ・ CD-ROM ドライブ

QuickTime の設定

Factory's FileMaker Plug-in を使用する場合は、下図の通り「ビデオ設定」で必ず「オプション」を選択してください。他のモードが設定されている場合、Factory's FileMaker Plug-in は正常に機能しません。



プログラムのユーザ登録とシリアル番号

Factory's FileMaker Plug-in がインストールされたファイルメーカー Pro を初めて起動すると、ユーザ登録を行うダイアログが表示されます。

デモ版として使用する場合

デモ版として使用される場合は、同梱の「デモ版用シリアル番号」ドキュメントに記載されているシリアル番号をご利用ください。

「デモ版用シリアル番号」を入力された場合は、シリアル番号を入力してから30日間、製品版と変わることなく使用することができます。また、デモ版の使用時、ファクトリーのオンラインショッピングコーナーで Factory's FileMaker Plug-in のシリアル番号を購入することによって得られる製品版シリアル番号を入力することで、改めて製品版として使用し続けることができます。

製品版として使用する場合

パッケージ版の場合：

「CD-ROM ケース」に貼られている「シリアル番号」を入力してください。

オンライン版の場合：

ファクトリーのオンラインショッピングコーナーで Factory's FileMaker Plug-in のシリアル番号を購入いただくことによって返送されるメールに記載されています。

ユーザ登録

製品のサポートを行う上で、ユーザ登録は非常に重要です。お手数ですが、Web 上のユーザ登録専用ページ、または同梱されます「ユーザ登録 Fax フォーム（PDF ファイル）」を利用して登録を行ってください。

ユーザ登録専用ページ

https://www.factory-1987.co.jp/SSL/Online_Regist/regist.html

ユーザサポート

本製品へのご質問およびユーザサポートは、基本的にWeb上に設けられた専用のページからのみ受けつけてます（ユーザサポートは、ユーザ登録をされたご本人以外からのご質問には回答いたしません）。

事情により電子メールあるいはファックスにてご質問を送られるお客様の場合は、必ず次の必要事項をお書き添えのいただき、製品サポート専用のメールアドレスまたはファックス番号までお送りください。

いずれの場合も回答には通常、3営業日以内を原則としておりますが、場合によっては多少日数がかかることを予めご了承ください。

必要事項

- ・製品シリアル番号
- ・製品名
- ・氏名
- ・会社名
- ・部署名
- ・電話番号
- ・ファックス番号
- ・コンピュータの機種
- ・メモリ
- ・OSバージョン
- ・ファイルメーカーのバージョン
- ・問題点

製品サポート専用メールアドレス : support@factory-1987.co.jp

ファックス番号 : 03-5486-6543

サイトライセンスパッケージについて

企業内で使用される場合にお得なサイトライセンスパッケージをご用意しております。

サイトライセンスパッケージとは、Factory's FileMaker Plug-inをまとめて複数本購入を希望されるユーザーのためパッケージで、通常のパッケージを必要本数分購入される場合と比べ、価格および管理の面で下記のようなメリットがあります。

- 単体のパッケージを同数購入した場合と比べ価格が安い。
- 所有する本数分のシリアル番号を管理する必要がない（一つのシリアル番号で全体を管理することができる）。
- ユーザー登録を含め、アップグレードなどの処理もサイトライセンスパッケージに対し、一括して受けることができるため、個別処理の手間がかからない。

具体的な価格および購入方法に関しましては、パッケージに同梱される「Site License Program について」書類をお読みいただくか、下記の Factory's FileMaker Plug-in サポートページをご覧ください。

http://www.factory-1987.co.jp/Factory/FileMaker_Plug-in_folder/FFPISupport.html

ランタイムアプリケーションでの利用について

Factory's FileMaker Plug-in を利用したデータベースを、ファイルメーカー Pro Developer 版を用いて作成したランタイムアプリケーションとして配布あるいは販売される場合、ご購入いただいた通常製品の Factory's FileMaker Plug-in をそのまま使用することはできません。別途ランタイムアプリケーション専用の Factory's FileMaker Plug-in が必要になります。

ランタイムアプリケーション専用の Factory's FileMaker Plug-in は、ランタイムアプリケーションとして作成されたタイトル毎のランタイムアプリケーションライセンスをお申し込みいただき、契約を結ぶ必要があります。

ランタイムアプリケーションライセンス契約完了時、お申し込みいただいたタイトルで使用可能となるランタイムアプリケーション専用 Factory's FileMaker Plug-in をご送付いたします。したがって、ご契約いただいたタイトル以外のランタイムアプリケーションでは使用することができません。

ランタイムアプリケーションライセンスの詳細および申込書の入手方法につきましては、下記の Factory's FileMaker Plug-in サポートページをご覧ください。

http://www.factory-1987.co.jp/Factory/FileMaker_Plug-in_folder/FFPISupport.html

Factory's FileMaker Plug-in の使い方

書式について

ファイルメーカー Pro にあらかじめ用意された関数と同様に利用することができます。

External ("FImg-DrawImage", 引数)

のように[計算式の指定]ダイアログボックスにコマンド名と引数の形式で入力します。

プラグインの機能は、ファイルメーカー Pro の External 関数を使って呼び出されます。この関数は二つの文字列引数を取り、最初の文字列はコマンドの名前、二つ目の文字列はそのコマンドが必要とする引数となります。それぞれの文字列は、その場で直接指定する文字列定数でもかまいませんし、テキスト型のフィールドでもかまいません。また計算の結果得られる文字列を与えることもできます。

Factory's FileMaker Plug-in のコマンドは "FImg-" で始まるものと "FUtl-" で始まるものがあります。

引数について

関数には引数を必要とするものとししないものがあります。また引数を必要とする場合にも、指定するタイプや内容がいろいろあります。

引数を必要としないコマンドの例です。

External ("Flmg-Version" , "")

このコマンドはインストールされているプラグインのバージョンを調べるもので、実行に当たり他の情報を必要としないため、引数は記述する必要がありません。ただし、この場合でも空の文字列を形式的に渡さなければいけません。

引数を必要とするコマンドの例です。

**External ("Flmg-DrawImage" , "SRC='D:¥¥Data¥¥Sample.JPG'
DESTRECT=10,10,310,210 ERASE OFFSCREEN")**

引数には情報の名前（以降「タグ」）を付け、タグの後に内容の値を記述するという形式になります。情報の種類が複数に渡る場合は、スペースで区切って次の情報を続けて記述します。このタグとタグの情報の連続した文字列を「タグ付きフォーマット」といいます。詳しくは「タグ付きフォーマットについて」の説明を御覧ください。

タグの値については、あらかじめ用意された固定の文字列や数値を記述する場合もありますし、パス（ファイルの位置情報）のように状況によって内容が変わる場合もあります。

[例題]

パスまたは "clipboard:" のいずれかを指定します。

SRC = path | 'clipboard:'

← "SRC='D:¥¥Data¥¥Sample.JPG'"

← "SRC='clipboard:'"

表示エリアをレクタングルで指定します。

DESTRECT = rect

← "DESTRECT=10,10,310,210"

反転の値として "horizontal"、"vertical"、"both" のいずれかの値を指定します。

FLIP = horizontal | vertical | both

← "FLIP = horizontal"

色の階調に "1"、"2"、"4"、"8"、"16"、"32"、"40" のいずれかの値を指定します。

DEPTH = 1 | 2 | 4 | 8 | 16 | 32 | 40

← "DEPTH=32"

幅をピクセルまたはパーセントの値で指定します。

WIDTH = pixel | percent

← "WIDTH=80"

← "WIDTH=50%"

"right"、"left"、のいずれかの値または回転する度数を指定します。

ROTATE = **right** | **left** | **integer**

← "ROTATE=right"

← "ROTATE=30"

1つまたは複数の拡張子を指定します。

EXTENSION = **拡張子1, 拡張子2, ..., 拡張子N**

← "EXTENSION='PCT,JPG'"

タグ名だけの機能を有効にする引数もあります。

"OFFSCREEN" という記述だけで OFFSCREEN の機能が有効になります。

OFFSCREEN [= **boolean**]

← "OFFSCREEN"

以上のように引数の指定にはいろいろな指定方法があり、さらにタグ名が同じであってもコマンドによって指定方法が異なりますので、詳しくは各コマンドの説明を御覧ください。

[参照]

タグ付きフォーマットについて P.28

値の形式について P.23

Macintosh と Windows が混在する環境

MacintoshとWindowsでは引数が異なるものがあります。二つのプラットフォームが混在するときの引数の記述について説明します。

引数の異なるコマンドに「FUtl-OpenFileDailog」があります。このコマンドではファイルの種類を引数で指定するようになっていますが、Macintosh と Windows では引数が異なります。

Macintosh ではファイルの種類を "FILETYPE" で指定します。
`External("FUtl-OpenFileDialog" , "FILETYPE='PICT,JPEG'")`

Windows ではファイルの種類を "EXTENSION" で指定します。
`External("FUtl-OpenFileDialog" , "EXTENSION='PCT,JPG'")`

二つのOSが混在する環境で引数を記述するときは、両方の引数を記述することで、一つの命令を二つのプラットフォームで実行させることが可能になります。

`External("FUtl-OpenFileDialog" , "FILETYPE='PICT,JPEG' EXTENSION='PCT,JPG'")`

上記のコマンドをMacintoshの環境で実行した場合は、"FILETYPE" の記述を判断して処理を実行し、"EXTENSION" の記述については無視します。Windowsの環境で実行した場合は、"EXTENSION" の記述を判断して処理を実行し、"FILETYPE" の記述については無視します。また、無視された引数の記述があっても実行時にエラーになりません。

返値について

Factory's FileMaker Plug-inのコマンドは、実行結果の返値でコマンドが正常に動作したかどうかを判断することができます。正常に動作した場合と動作しなかった場合の返値について説明します。

コマンドが正常に動作した場合の実行結果

何か情報を取得するためのコマンドを実行した場合は、正常に動作した結果として、目的の情報が返値として返されます。それ以外の場合、例えば画像を表示するようなときは、表示するという動作自体がコマンドの結果なので返値はありません。

[例題]

画像の情報を得るために「FImg-ImageInfo」を実行した場合：

クリップボードにある画像（幅 389、高さ 266、フルカラー、72DPI、サイズ 150K）の情報を調べる例です。

引数

← "SRC='clipboard:'"

返値

→ "WIDTH=389 HEIGHT=266 DEPTH=32 RESOLUTION=72 SIZE=151102"

複数の情報が返される場合はタグ付きフォーマットで返されます。

画像を表示するための「FImg-DrawImage」を実行した場合：

External ("FImg-DrawImage" , "SRC='C:¥¥DATA¥¥Sample.JPG'")

DESTRECT=10,10,310,210 ERASE OFFSCREEN")

→ ""

ウィンドウの指定エリアに画像が表示されたことで正常に動作したことが確認できます。

ファイルのパスを取得するために「FUtl-OpenFileDialog」を実行した場合：

External ("FUtl-OpenFileDialog" , "EXTENSION='PCT,JPG'")

→ "C:¥¥DATA¥¥Sample.jpg"

返される値が 1 種類の場合はタグは付きません。

コマンドが正常に動作しなかった場合の実行結果

コマンドが正常に動作しない場合には、エラーの種類を示すエラー番号が返されます。エラーの原因として、コマンドの使い方が間違っている場合と環境に問題がある場合が考えられます。どういう場合に実行結果がエラーになるのかについて説明します。

[例題]

画像を表示するのに引数の記述にミスがあるのに「FImg-DrawImage」を実行した場合：

External ("FImg-DrawImage" , "SRCC='C:¥¥Data¥¥Sample.JPG'")

DESTRECT=10,10,310,210 ERASE OFFSCREEN")

→ "ERROR=-50"

ファイルのパスを指定するタグ名は "SRC" と記述しなければいけないのに "SRCC" と記述した場合は必要とする "SRC" タグが見つからないのでパラメータエラーになります。省略可能なのでタグの綴りを間違えた場合は単に無視されます。

Factory's FileMaker Plug-in が「System」フォルダにインストールされていない、またはファイルメーカーProのアプリケーション環境設定でプラグインの使用状態がチェックされていないのに「FUtl-Version」を実行した場合：

`External ("FUtl-Version" , "")`

→ なし

この場合は「FUtl-Version」コマンドに限らず、Factory's FileMaker Plug-in のすべてのコマンドが同様の結果になります。

正常に動作した結果として返されたエラー

返値がエラーになる場合でも正常に動作するコマンドがあります。

[例題]

ファイルのパスを取得するために「FUtl-OpenFileDialog」を実行し、表示されたファイル選択ダイアログでキャンセルボタンをクリックした場合：

`External ("FUtl-OpenFileDialog" , "EXTENSION='PCT,JPG'")`

→ "ERROR=-128"

この場合エラーとしていますが、本来パスを取得するために実行したコマンドを途中でキャンセルしたもので、環境の問題でもなくまたコマンドの使い方を間違えたということでもありません。

同じ名前のファイルが複数ある場合に、各ファイルの場所をすべて検索するために「FUtl-FindFile」を繰り返し実行した場合：

`External ("FUtl-FindFile" , "")`

→ "ERROR=-43"

この場合のエラーは該当するファイルがなくなるまで最後まで検索したことを示しています。

ほとんどのコマンドが返値のエラーコードを見て正常に動作したのかどうか実行結果を判断することができます。ただしエラーコードの内容によっては正常な動作として判断できるものやエラーコードがなくても環境に問題があるケースもあるのだということを覚えておいてください。

返値からエラー番号を調べる方法

エラー番号はタグ付きフォーマットで返されます。タグは"ERROR"です。

この返値からエラー番号を抜き出すためには「FUtl-Tag」を使います。

[例題]

「gResult」フィールドに返値が入っている場合：

`External ("FUtl-Tag" , "TAG=ERROR " & gResult)`

→ "-128"

[参照]

FUtl-Tag P.48

タグ付きフォーマットについて P.28

値の形式について P.23

値の形式について

Factory's FileMaker Plug-inにはたくさんのコマンドが含まれていますが、それらの引数や返値として使われる値には、文字列や数値、ファイルのパスのような決まった形式があります。それらを正しく理解しない限り思った通りの結果は得られません。ここでは値の形式について説明します。

値の形式は大きく分けると二つに別れます。一つは基本的な値の形式でこれには、文字列、整数値、真偽値があります。そしてそれらの基本的な値を元にした値の形式がいくつか定義されています。このような値の形式は、基本的には文字列か整数値、または真偽値なのですが、ある特定の使われ方をするために区別しています。それは複数の値のリストであったり、値の範囲に制限が加えられていたりします。例えばファイルの場所をあらわす path がありますが、これは実際にはただの文字列です。しかしパスを表す決まったルールがあり、区別しておいた方が便宜上都合がいいので一つの形式として扱っています。

基本的な値の形式

文字列 (text)

ひとつながりの文字の集まりが文字列です。文字列は一般的にシングルクォーテーションかダブルクォーテーションで前後を囲む必要があります。ただし、文字列が単純な英数字しか含まず、途中で空白文字などがない場合には、クォーテーションで囲む必要はありません。扱える文字の長さは、最大二千文字までです。

[例題]

Hello
"Hello World"
"私の名前は鈴木です。"

文字列の中に特殊な文字やクォーテーションを含めたいときは、後述する「エスケープ・シーケンス」を使う必要があります。

.....

整数値 (integer)

整数値は数をあらわす数字の集まりです。先頭に「- (マイナス)」記号をつけると負の数を表します。実数は扱えませんが小数点などは使えません。また位取りのためのカンマ (,) も使うことはできません。扱える値の範囲は -2147483648 から 2147483647 までの間です。

[例題]

0
1024
-36000
14008922

.....

真偽値 (boolean)

真偽値は正しいか正しくないかを表すものです。ある機能を設定するかしないかの指示や、機能の状態を調べる時に使われます。値にはTrue (真) か False (偽) しかありません。大文字小文字は区別されませんが、返値として使われる場合には「True / False」の形で返されます。

[例題]

True
False
TRUE
false

.....

それ以外の値の形式

pixel (ピクセル)

画像の幅や高さなど、ピクセル単位の整数値を指定する場合に使われます。値は0以上の整数値でなくてはなりません。

[例題]

640
228
1

.....

percent (比率)

元の大きさに対する比率をパーセント単位で指示する時に使う整数値です。他の形式と併用することがありますので、それらと区別するために「% (パーセント記号)」を数値の後に付けます。値は1以上の整数値でなくてはなりません。

[例題]

100%
250%
33%

.....

point (ポイント)

座標を示す時に使う二つの数値のリストです。それぞれX座標、Y座標を表します。数値と数値はカンマ (,) で区切ります。数字とカンマの間に空白を含めることはできません。

[例題]

0,0
200,150
-10,-30

.....

rect (レクタングル)

レクタングル (矩形) を指定する時に使う四つの数値のリストです。それぞれレクタングルの左、上、右、下の座標を表します。数値と数値はカンマ (,) で区切ります。数字とカンマの間に空白を含めることはできません。

[例題]

0,0,200,150
-100,-100,100,100
160,96,380,274

RGBColor (カラー)

色を指定する時に使う値の形式です。RGB カラーの構成要素である赤、緑、青、それぞれの色を三つの数値のリストで指定します。各色の値は0から255の間です。0が最も暗く、255が最も明るい色になります。

指定する色がグレースケールの色の場合、RGB 各色の値は同じになります。これを三つ指定するのは冗長ですので、この場合には数値を一つだけ指定することでグレースケールの色と判断します。値の範囲はやはり0から255の間です。この場合、0が黒で、255が白です。

また、HTMLなどで使われている「#RRGGBB」という形も使うことができます。この形の場合には「# (シャープ)」に続けて16進数で赤、緑、青の値を指定します。値の範囲は00からFFまでです。AからFの16進数の部分は小文字でも大文字でもかまいません。

[例題]

255,0,0	赤 (赤が255で最大で、他の色が0で最小なので赤だけになる)
128	グレー (128,128,128と同じ)
#CCCCCC	ライトグレー (204,204,204と同じ)

OSType

ファイルの拡張子などいろいろな種類を表すためにWindows OSによって使われる3文字または4文字までの文字列です。

[例題]

BMP
JPG,GIF,PDS (複数の指定が許されている場合のみ)

path (単純なパス)

ファイルの場所を指定する場合、ファイルのパスを文字列で記述します。ファイルのパスの場合、ハードディスクの名前、途中のフォルダの名前、最終的なファイルの名前を「¥ (円マーク)」で区切って表します。フォルダのパスの場合には最後のフォルダのあとにさらに「¥ (円マーク)」を続けます。これでそのパスがフォルダであることを指示します。このルールはWindowsのパスでしか使うことはできません。Factory's FileMaker Plug-inでは円マークを認識するためには「¥¥」と重ねて記述します。詳しくは「エスケープ・シーケンスについて」をご覧ください。

[例題]

"C:¥¥DATA¥¥Sample.pct"
"C:¥¥WINDOWS¥¥"

path (ユニバーサル・パス)

ファイルがコンピュータのハードディスクの中でなくサーバー上にある場合には、単純なパスの指定だけではファイルを特定できません。サーバー上のファイルの場合には、さらにゾーン名、ホスト名が必要になります。また、サーバー上のファイルはMacintosh環境だけでなくWindowsからでも参照される可能性があります。こうした必要性から、Factory's FileMaker Plug-in では、独自に「ユニバーサル・パス」というものを定義し、Macintosh からでもWindows からでも同じパスの表記で参照できるようにしました。

ユニバーサル・パスは、単純なパスの先頭にホスト名とゾーン名を追加したものです。以下のような規則を使います。

ゾーンが定義してある場合：

Host Name@Zone Name::Hard Disk Name:Folder Name 1:...:File Name

ゾーンが存在しない場合：

Host Name::Hard Disk Name:Folder Name 1:...:File Name

単純なパスとの明確な違いは、文字列の中に「::」というコロンの並びが存在することです。これが含まれたパスはユニバーサル・パスとして扱われます。「::」の後ろはボリューム名、フォルダ名、ファイル名が続きます。区切りはMacintosh でもWindows でも「:(コロン)」で共通です。このため、ユニバーサル・パスはどちらの環境でも同じように扱うことができます。

[例題]

"Factory Server::画像 HD:風景:世田谷公園:犬.jpg"

"TestServer@SampleZone::SomeHD:UnknownFolder:AnyFile.txt"

path (URL)

QuickTime 4をインストールしてある場合、インターネット上の画像を直接持ってきて表示することが可能です。HTTPとFTPのプロトコルが使えます。HTTPの場合には「http://」で始めます。FTPの場合には「ftp://」で始めます。参照するURLはQuickTimeが扱うことのできる画像ファイルでなくてはなりません。また、書き出し用のパスとしては使うことはできません。

[例題]

"http://www.factory-1987.co.jp/suzuki/img/yosuke.jpg"

エスケープ・シーケンスについて

区切りに使っているクォーテーションそのものを文字列の中に入れた場合、そのままクォーテーションを含めるとそこで文字列の終わりとして解釈されてしまいます。例えば以下の例では、文字列の中にシングルクォーテーションを含めたいと意図していましたが、文字列を区切る文字自体がシングルクォーテーションのため、正しく認識されません。

'Hello, I'm Yosuke.' → 「Hello, I」までしか文字列としてみなされない。

こうした場合には、文字列中のクォーテーションの前に「¥ (円マーク)」を置くことでその次の文字を特別に扱わないようにし、文字列の終わりの意味ではないように変更できます。

'Hello, I¥'m Yosuke.' → 「Hello, I'm Yosuke.」正しく認識されます。

このように「¥ (円マーク)」と続く文字を文字列中に置くことをエスケープ・シーケンスと呼びます。上記の例のようなシングルクォーテーション以外にも、改行コードやタブなど見えない文字を含めたい場合にも使われます。Factory's FileMaker Plug-inが認識するエスケープ・シーケンスには以下のものがあります。

文字	意味
¥"	ダブルクォーテーション
¥'	シングルクォーテーション
¥r	改行
¥n	復帰
¥t	タブ
¥s	スペース
¥¥	円マーク

最後のエスケープ・シーケンスは円マークを表します。円マークがあらわれるとそこからエスケープシーケンスと判断されてしまうため、円マーク自体を文字列に含めたい場合には円マークを二つ重ねる必要があります。

タグ付きフォーマットについて

引数で必要な情報を記述するときや返値でいろいろな情報が返されるときには、ほとんどの場合タグ付きフォーマットが利用されています。タグが付いていることによって、いろいろな情報が混在する文字列から欲しい情報だけを抜き出すことができるようになっています。

[例題]

FImg-DrawImageコマンドで引数として指定したタグ付きフォーマットの文字列の例です。

← "SRC='D:¥¥DATA:IMG001.JPG' DESTRECT=10,80,330,320 ERASE OFFSCREEN "

FUtl-FileInfoコマンドを実行した結果として取得したタグ付きフォーマットの文字列の例です。

→ "NAME='IMG001.jpg' KIND=file SIZE=20084 DATASIZE=20084 EXTENSION='.jpg'
LOCKED=False INVISIBLE=False CREATEDDATE=2001.4.21 CREATEDTIME=4:20:18
MODIFIEDDATE=2000.3.4 MODIFIEDTIME=18:45:40 LOCATION='D:¥¥DATA¥¥'
PATH='D:¥¥DATA¥¥IMG001.JPG'"

またFactory's FileMaker Plug-inにタグ付きフォーマットの文字列を制御するコマンドも用意されています。指定したタグの値を取得するコマンドとして「FUtl-Tag」が、タグ付きフォーマットの文字列からタグの名前を取得するコマンドとして「FUtl-TagList」があります。

詳しくはリファレンスをご覧ください。

[参照]

FUtl-Tag P.48

FUtl-TagList P.49

引数について P.18

返値について P.21

Image リファレンス

Flmg-Version

Factory's FileMaker Plug-in のバージョンを返します。

[書式] External ("Flmg-Version", 引数)

[引数] なし

[返値] Plug-in のバージョンを返します。

ファイルメーカー Pro の「System」フォルダに Plug-in が入っていない場合、またはファイルメーカー Pro のアプリケーション環境設定で Factory's FileMaker Plug-in の使用がオフにされているときには、この関数の返値がヌル文字列になります。

VERSION = text

Plug-in のバージョン番号を返します。

.....

[メモ] Factory's FileMaker Plug-in が正しくインストールされている場合には、必ず何らかの文字列が返るので、Factory's FileMaker Plug-in がインストールされているかをチェックするために利用することもできます。

[例題] Factory's FileMaker Plug-in のバージョンを調べたときの例です。

→ "VERSION=1.0"

FImg-DrawImage

ファイルやクリップボードに収められた画像をウィンドウの指定の位置に表示します。

表示する画像がファイルの場合はパスで指定します。クリップボードにある場合はクリップボード ("clipboard:") を指定します。さらに画像の表示エリアとしてウィンドウ上の4つのポイント (左, 上, 右, 下) を指定します。画像の表示方法として拡大縮小や回転などの効果を指定できます。詳しくは各引数の説明を御覧ください。

[書式] External ("FImg-DrawImage", 引数)

[引数]

SRC = path | 'clipboard:'

画像をパスまたはクリップボードのいずれかで指定します。この引数は省略できません。

DESTRECT = rect

画面上の表示エリアを指定します。起点はウィンドウのステータスエリアを除いた左上になります。この引数は省略できません。

ウィンドウの左上 10,10 から右下 310,210 に表示エリアを指定します。

← "DESTRECT=10,10,310,210"

STATUSAREA [= boolean]

ステータスエリアの有無状態を指定します。ステータスエリアがあるときは True にします。ステータスエリアがないときは False とします。省略した場合は最適化された状態をある程度自動的に判断します。

OFFSCREEN [= boolean]

画像を画面に表示する前に一度オフスクリーンに展開する場合は True、展開しない場合は False にします。

その他の引数については「FImg-DrawImage, FImg-GrabImage, FImg-SaveImage 共通のオプション引数」をご覧ください。

[返値]

実行結果

[例題] 画像ファイルから 200,100,250,150 の部分をトリミングして表示する例です。

```
← "SRC='D:¥¥DATA¥¥Sample.JPG' DESTRECT=10,80,330,320  
TRIMRECT=200,100,250,150 ERASE OFFSCREEN"
```

表示サイズを 50%、背景色を黒に設定して表示する例です。

```
← "SRC='D:¥¥DATA¥¥Sample.JPG' DESTRECT=10,80,330,320 WIDTH=50%  
HEIGHT=50% ERASE=0,0,0 OFFSCREEN"
```

[参照] FImg-DrawImage, FImg-GrabImage, FImg-SaveImage 共通のオプション引数 P.35
FImg-DrawImage, FImg-GrabImage, FImg-SaveImage 複数コマンドの一括指定 P.37

Flmg-GrabImage

画像をクリップボードに取り込みます。

[書式] External ("Flmg-GrabImage", 引数)

[引数] **SRC = path | 'clipboard:'**

画像をパスまたはクリップボードのいずれかで指定します。この引数は省略できません。

.....
DESTRECT = rect

描画エリアの大きさを指定します。指定した場合には、このエリアを基準として描画が行われます。省略した場合は最初の画像の大きさになります。複数の画像を重ね合わせたものの全体を取り込む場合は全体が納まるエリアを DESTRECT に指定します。

.....
その他の引数については、「Flmg-DrawImage, Flmg-GrabImage, Flmg-SaveImage 共通の引数」の説明を御覧ください。

[返値] 実行結果

[例題] 画像ファイルを縮尺を変えずオリジナルサイズのままクリップボードに取り込む例です。

← "SRC='D:¥¥DATA¥¥Sample.JPG'"

オリジナルの 50% の大きさ画像ファイルをクリップボードに取り込む例です。

← "SRC='D:¥¥DATA¥¥Sample.JPG' WIDTH=50% HEIGHT=50%"

[参照] Flmg-DrawImage, Flmg-GrabImage, Flmg-SaveImage 共通のオプション引数 P.35
Flmg-DrawImage, Flmg-GrabImage, Flmg-SaveImage 複数コマンドの一括指定 P.37

Flmg-SaveImage

画像をファイルに保存します。

[書式] External ("Flmg-SaveImage", 引数)

[引数] **SRC = path | 'clipboard:'**

画像をパスまたはクリップボードのいずれかで指定します。この引数は省略できません。

.....
DESTRECT = rect

描画エリアの大きさを指定します。指定した場合には、このエリアを基準として描画が行われます。省略した場合は最初の画像の大きさになります。複数の画像を重ね合わせたもの全体を取り込む場合は全体が納まるエリアを DESTRECT に指定します。

.....
EXTENSION = 拡張子

拡張子でファイルの種類を指定します。省略した場合はダイアログが表示されますので、ダイアログ上でファイルの種類を選ぶようになります。

.....
DEST = path

ファイルを保存するときのパスを指定します。省略した場合はダイアログが表示されますので、ダイアログ上で名前を入力するようになります。

.....
その他の引数については、「Flmg-DrawImage, Flmg-GrabImage, Flmg-SaveImage 共通の引数」の説明を御覧ください。

[返値] 実行結果 | パス

[例題] クリップボードにある画像を JPEG ファイルとして保存する例です。

[参照] ← "SRC='clipboard:' EXTENSION='JPG' DEST='D:¥¥DATA¥¥Sample.PCT'"

オリジナルの 50% のサイズで画像ファイルを保存する例です。

← "SRC='D:¥¥DATA¥¥Sample.PCT' EXTENSION='JPG'
DEST='D:¥¥DATA¥¥Sample.PCT' WIDTH50% HEIGHT=50%"

Flmg-DrawImage, Flmg-GrabImage, Flmg-SaveImage 共通のオプション引数 P.35

Flmg-DrawImage, Flmg-GrabImage, Flmg-SaveImage 複数コマンドの一括指定 P.37

Flmg-DrawImage, Flmg-GrabImage, Flmg-SaveImage 共通のオプション引数

[引数]

WIDTH = *pixel* | *percent*

表示する画像サイズ (幅) をピクセルまたはパーセントで指定します。いずれも整数値で指定してください。省略した場合は 100% となります。

幅 100 ピクセル

← "WIDTH=100"

幅 80 パーセント

← "WIDTH=80%"

HEIGHT = *pixel* | *percent*

表示する画像サイズ (高さ) をピクセルまたはパーセントで指定します。いずれも整数値で指定してください。省略した場合は 100% となります。

高さ 80 ピクセル

← "HEIGHT=80"

高さ 80 パーセント

← "HEIGHT=80%"

TRIMRECT = *rect*

画像の表示される部分を *rect* のエリアだけに制限します。指定がない場合は画像全体が表示されます。*rect* は元画像の座標で指定します。

FLIP = *horizontal* | *vertical* | *both*

画像を反転させた状態で描画します。*horizontal* は上下反転します。*vertical* は左右反転します。*both* は上下と左右で反転します。

ROTATE = *right* | *left* | *integer*

画像を回転させた状態で描画します。右 90 度回転させるときは *right* を指定します。左 90 度回転させるときは *left* を指定します。また角度を任意で指定する時は *integer* で指定します。

DISPLAYMODE = *ScaleToFit* | *ShrinkToFit* | *Fill* | *Anchored*

画像の表示モードを指定します。省略した場合は *Anchored* です。

ScaleToFit は表示エリアに対して縦横の縮尺比を変えずにエリア全体に納まるように表示します。

ShrinkToFit は縦横比を崩さず画像全体を表示します。エリアより小さい場合はそのまま表示されます。

Fill は表示エリア全体に表示します。縦横比を無視して表示します。

Anchored は原寸大で表示します。

ANCHOR = TopLeft | TopRight | BottomLeft | BottomRight | Center

DISPLAYMODE=Anchor のとき有効になります。画像の表示位置を指定します。省略した場合は Center です。

TopLeft は画像の左上を表示エリアの左上に寄せます。

TopRight は画像の右上を表示エリアの右上に寄せます。

BottomLeft は画像の左下を表示エリアの左下に寄せます。

BottomRight は画像の右下を表示エリアの右下に寄せます。

Center は画像の中央を表示エリアの真ん中に合わせます。

OFFSET = point

画像の表示位置を右下方向に指定した量だけ移動します。省略した場合は (0,0) です。

SCROLL = point

画像の表示位置を左上方向に指定した量だけ移動します。オフセットと逆方向になります。省略した場合は (0,0) です。

ERASE [= RGBColor]

表示の前に描画エリアを指定の色で塗りつぶします。色を省略した場合は白で塗りつぶします。

GRAPHICSMODE = srcCopy | notSrcCopy | blend | transparent | addPin | addOver | subPin | subOver | addMax | addMin

画像の描画モードを指定します。詳しくは「付録 2：描画モード一覧」をご覧ください。

OPCOLOR = RGBColor

GRAPHICSMODE が blend、transparent、addPin、subPin の時に有効となります。GRAPHICSMODE で使われるオペレーションカラーを指定します。省略するとグレーになります。GRAPHICSMODE の指定がないとき、または上記以外のモードの時は無効になります。詳しくは付録をご覧ください。

PAGE = integer

複数ページから構成される画像のページ番号を指定します。この引数は QuickTime4 が必要になります。

[参照]

FImg-DrawImage P.31

FImg-GrabImage P.33

FImg-SaveImage P.34

付録 2：描画モード一覧 P.95

FImg-DrawImage, FImg-GrabImage, FImg-SaveImage 複数コマンドの一括指定

FImg-DrawImage, FImg-GrabImage, FImg-SaveImage コマンドでは複数の指示を 1 回のコマンドで実行することができます。

棒線 ("|") で区切ることで複数回の指示を並べて記述することができます。一度設定された内容は同じコマンド内であれば、継承されますので、同じ設定であれば毎回設定する必要はありません。

二重縦棒 ("||") で区切った場合は、以降の設定は前回の設定を継承せずに引数の設定を行う場合に利用します。

[例題]

2つの画像ファイルを同一のエリアに表示する例題です。

```
→ "SRC='D:¥¥DATA¥¥Sample.JPG' DESTRECT=21,21,410,188  
DISPLAYMODE=ANCHORED ANCHOR=CENTER | SRC='Clipboard:'  
ANCHOR=BOTTOMRIGHT OFFSET=-2,-2 ROTATE=0 WIDTH=100% HEIGHT=100%"
```

ERASE の指示については最初に 1 回だけ有効になります。

[メモ]

FImg-DrawImage P.31

[参照]

FImg-GrabImage P.33

FImg-SaveImage P.34

Flmg-ImageInfo

画像の情報を取得します。

[書式] External ("Flmg-ImageInfo", 引数)

[引数] **SRC = path | clipboard:**

画像をパスまたはクリップボードのいずれかで指定します。この引数は省略できません。

.....
PAGE = integer

複数ページから構成される画像のページ番号を指定します。この引数はQuickTime4が必要になります。

.....

[返値] 幅、高さ、階調、解像度、サイズ、ページ数

WIDTH = pixel

幅を返します。

.....

HEIGHT = pixel

高さを返します。

.....

DEPTH = 1 | 2 | 4 | 8 | 16 | 32 | 34 | 36 | 40

階調を返します。34、36、40 はそれぞれ2ビット、4ビット、8ビットのグレースケールです。

.....

RESOLUTION = integer | point

解像度を返します。縦横の解像度が異なる場合はカンマで区切られた値を返します。単位はdpiです。

.....

SIZE = integer

SRC='clipboard:' のとき、画像のメモリ上での容量を返します。

.....

PAGECOUNT = integer

複数ページで構成される画像の場合、ページ数を返します。このタグを参照するにはQuickTime4が必要になります。

.....

【例題】

クリップボードにある画像（幅 389、高さ 266、フルカラー、72DPI、サイズ 150K）の情報を調べる例です。

引数

← "SRC='clipboard:'"

返値

→ "WIDTH=389 HEIGHT=266 DEPTH=32 RESOLUTION=72 SIZE=151102"

Flmg-PreviewImage

画像のサムネールを作成しクリップボードに取り込みます。

[書式] External ("Flmg-PreviewImage", 引数)

[引数] **SRC = path | 'clipboard:'**
 画像をパスまたはクリップボードのいずれかで指定します。この引数は省略できません。

WIDTH = pixel
 サムネールの幅をピクセルで指定します。省略した場合は80ピクセルになります。

HEIGHT = pixel
 サムネールの高さをピクセルで指定します。省略した場合は80ピクセルになります。

BORDER = pixel
 サムネールを指定の太さの枠線で囲みます。省略した場合は0になります。

PAGE = integer
 複数ページから構成される画像のページ番号を指定します。この引数はQuickTime4が必要になります。

[返値] 画像

クリップボードにサムネール画像を返します。

[メモ] サムネールの幅と高さの指定が指定画像より大きい場合、指定画像のサイズでサムネール画像を返します。

[例題] 100x100ピクセルのサムネールを作成する例です。

← "SRC='D:¥¥DATA¥¥Sample.PCT' WIDTH=100 HEIGHT=100"

Flmg-CompressImage

クリップボードの画像を指定の圧縮方法で圧縮し、クリップボードに戻します。

[書式] External ("Flmg-CompressImage", 引数)

[引数] **CODEC = OSType**
圧縮法を指定します。

.....
QUALITY = integer | Min | Low | Normal | High | Max | Lossless

画質を指定します。*integer* の場合は 0 ～ 1024 の範囲で指定します。

0 が最低画質で 1024 が最高画質です。画質が上がれば圧縮率は下がります。

.....
DEPTH = 0 | 1 | 2 | 4 | 8 | 16 | 24 | 32 | 34 | 36 | 40

色の階調を指定します。34 は 4 色、36 は 16 色、40 は 256 色グレースケールです。

0 は最適な階調を選び使用します。
.....

[返値] 圧縮した画像

[メモ] 主な圧縮法には 'jpeg' (フォト JPEG)、'tiff' (TIFF)、'WRLE' (BMP) などがあります。

[例題] クリップボードの画像を圧縮法 JPEG、階調をフルカラー、圧縮率を 512 で圧縮する例です。

← "CODEC=jpeg QUALITY=512 DEPTH=24"

[参照] Flmg-CompressImageDialog P.42

FImg-CompressImageDialog

画像の圧縮するときの圧縮方法、画質、階調をユーザに選択させます。

クリップボード上の画像がダイアログにサンプルとして表示されます。画像がない場合はエラーになります。

[書式] External ("FImg-CompressImageDialog", 引数)

[引数]

CODEC = *OSType*

ダイアログに表示される圧縮法の初期値を指定します。省略した場合は画像に最適な圧縮法を自動的に設定します。

.....
QUALITY = *integer* | Min | Low | Nomal | High | Max | Lossless

ダイアログに表示される画質の初期値を指定します。*integer* の場合は 0 ～ 1024 の範囲で指定します。

0 がクオリティ最低で 1024 がクオリティ最高です。画質が上がれば圧縮率は下がります。

.....
DEPTH = *integer*

ダイアログに表示される階調の初期値を指定します。指定した圧縮法に対応した値を指定してください。

[返値]

圧縮方法、画質、階調

CODEC = *OSType*

圧縮法を返します。

.....
QUALITY = *integer*

画質を返します。

.....
DEPTH = 0 | 1 | 2 | 4 | 8 | 16 | 24 | 32 | 34 | 36 | 40

色の階調を指定します。34 は 4 色、36 は 16 色、40 は 256 色グレースケールです。

0 は最適な階調です。

[メモ]

この返値はそのまま FImg-CompressImage の引数にも使えます。

[参照]

FImg-CompressImage P.41

Flmg-CheckArea

指定エリアをチェックして再表示のタイミングを判定します。

ウィンドウに画像を表示した後で、その表示エリアに別のウィンドウなどがかぶさってしまった場合、表示されていた画像が欠けてしまいます。この場合、もう一度画像を表示する必要があります。この「指定エリアのチェック」機能を使えば、この処理を自動化できます。

【書式】 External ("Flmg-CheckArea", 引数)

【引数】

DESTRECT = rect

画面上の表示エリアを指定します。起点はウィンドウのステータスエリアを除いた左上になります。この引数は省略できません。

ウィンドウの左上 10,10 から右下 310,210 に表示エリアを指定します。

← "DESTRECT=10,10,310,210"

.....
STATUSAREA [= boolean]

ステータスエリアの有無状態を指定します。ステータスエリアがあるときはTrueにします。ステータスエリアがないときはFalseとします。省略した場合は最適化された状態をある程度自動的に判断します。

.....

【メモ】

ファイルメーカーのスクリプトを定期的に行う設定にしてその中で画像の表示エリアをチェックすることにより、前回の状態からなにか変化しているかどうかわかります。変化している場合には再度表示コマンドを実行することになります。これにより、ダイアログの表示などで画像の表示が乱れてしまった場合でも、自動的に書き直しが行われます。

返値の数値は一過性のものであり、現在の画面の状態を表すものでしかありません。

【返値】 実行結果

【例題】

表示エリア 10,80,330,320 をチェックする例です。

← "10,80,330,320"

→ "-97228352"

【参照】 Flmg-DrawImageae P.31

Flmg-Effect

指定したエフェクトを使って画面の表示に効果を加えます。

ズームのアニメーション効果について説明します。

[書式] External ("Flmg-Effect", 引数)

[引数]

KIND = Zoom

ズームのエフェクトを指定します。この引数は省略できません。

SRCRECT = rect

ズームを開始する枠を指定します。

DESTRECT = rect

ズームが終了する枠を指定します。

DELAY = integer

待ち時間を tick 単位で指定します。省略した場合は 1 になります。1 tick は 1/60 秒です。

STEP = integer

ズームのステップを指定します。省略した場合は 8 になります。

ACCELERATE [= boolean]

加速しながらズームします。省略した場合は単調にズームします。

[返値] なし

[例題] ズームエフェクトの例です。

```
← "KIND=Zoom SRCRECT=20,20,410,188 DESTRECT=120,70,310,138 DELAY=1
STEP=20 ACCELERATE"
```

[メモ]

このエフェクトは画面上の二つのエリアの間をレクタングルが移動することにより、あたかもそのエリアがズームしたように見せるだけで、実際に内容がズームするわけではありません。

Flmg-GetMousePos

ウィンドウ内でマウスの位置を取得します。

[書式] External ("Flmg-GetMousePos", 引数)

[引数]

DESTRECT = rect

マウスがクリックされる対象となる領域を指定します。省略した場合はステータスエリアを除いたウィンドウ全体が対象となります。

.....
STATUSAREA [= boolean]

ステータスエリアの有無状態を示します。ステータスエリアがあるときはTrueにします。ステータスエリアがないときはFalseとします。省略した場合は最適化された状態をある程度自動的に判断します。

.....
ANCHOR = TopLeft | TopRight | BottomLeft | BottomRight | Center

マウスがクリックされた位置をどこを起点にした座標値にするかを指定します。省略した場合はTopLeft になります。

[返値]

マウス座標

[例題]

マウスがクリックされた位置を取得する例です。

← "DESTRECT=20,20,410,188 ANCHOR=Center"

Utility リファレンス

FUtl-Version

Factory's FileMaker Plug-in のバージョンを返します。

[書式] External ("FUtl-Version", 引数)

[引数] なし

[返値] Plug-in のバージョンを返します。

ファイルメーカー Pro の「System」フォルダに Plug-in が入っていない場合、またはファイルメーカー Pro のアプリケーション環境設定で Factory's FileMaker Plug-in の使用がオフにされているときには、この関数の返値がヌル文字列になります。

VERSION = text

Plug-in のバージョン番号を返します。

.....

[メモ] Factory's FileMaker Plug-in が正しくインストールされている場合には、必ず何らかの文字列が返るので、Factory's FileMaker Plug-in がインストールされているかをチェックするために利用することもできます。

[例題] Factory's FileMaker Plug-in のバージョンを調べた例です。

← "VERSION=1.0"

FUtl-Tag

タグ付きフォーマットから指定したタグの値を取得します。

タグ付きフォーマットとは「タグ名=値」の形式で連続した文字列となっています。その中から指定のタグの値を取り出します。

[書式] External ("FUtl-Tag", 引数)

[引数] **TAG = text**

タグの名前を指定します。この引数は省略できません。

.....

text

タグ付きフォーマットの文字列を指定します。この引数は省略できません。

.....

[返値] タグの値

[例題] FImg-ImageInfoコマンドで取得した画像情報 (gResult) から階調("DEPTH")を調べる例です。

gResult の内容については以下のようになっています。

"WIDTH=389 HEIGHT=266 DEPTH=32 RESOLUTION=72"

引数

← "TAG=DEPTH " & gResult

返値

→ "32"

[参照] FUtl-TagList P.49

タグ付きフォーマットについて P.28

FUtl-TagList

タグ付きのフォーマットからタグの名前の一覧を取得します。

文字列に含まれるタグの名前をすべて抜き出しスペース区切りの一覧にして返します。

[書式] External ("FUtl-TagList", 引数)

[引数] **text**

タグ付きフォーマットの文字列を指定します。この引数は省略できません。

.....

[返値] タグの名前

[例題] FUtl-FileInfo コマンドで取得したファイル情報 (gResult) からタグの名前を取得する例です。

gResult の内容については以下のようにになっています。

```
"NAME='Sample.jpg' KIND=file SIZE=20084 DATASIZE=20084 EXTENSION='jpg'
LOCKED=False INVISIBLE=False CREATEDDATE=2001.4.21 CREATEDTIME=14:54:38
MODIFIEDDATE=2000.3.4 MODIFIEDTIME=18:45:40 LOCATION='D:¥¥DATA¥¥'
PATH='D:¥¥DATA¥¥Sample.JPG'"
```

引数

← gResult

返値

→ "NAME KIND SIZE DATASIZE EXTENSION LOCKED INVISIBLE CREATEDDATE
CREATEDTIME MODIFIEDDATE MODIFIEDTIME LOCATION PATH"

[メモ] 返値のスペース区切りの文字列からタグを抜き出すためには、ファイルメーカーProの組み込み関数である LeftWords や RightWords を使います。

[参照] FUtl-Tag P.48

タグ付きフォーマットについて P.28

FUtl-OpenFileDialog

ファイル選択ダイアログを表示しファイルの場所を指定します。

[書式] External ("FUtl-OpenFileDialog", 引数)

[引数] **EXTENSION = 拡張子1, 拡張子2, ..., 拡張子N**

指定した拡張子と一致するファイルを表示します。

.....

MESSAGE = text

ダイアログにメッセージを表示します。省略した場合は「ファイルを開く」になります。

.....

SHOWALL [= boolean]

すべてのファイルを表示します。

.....

[返値] パス

[メモ] SHOWALLが指定されているときはEXTENSIONの指定は無効になり、すべてのファイルがダイアログに表示されます。

省略されている時は、'PICT image' のファイルをダイアログに表示します。

[例題] ファイル選択ダイアログを表示する例です。メッセージには「Please choose a file.」を指定します。またすべてのファイルを表示するように指定します。

← "MESSAGE='Please choose a file.' SHOWALL"

FUtl-OpenFolderDialog

フォルダ選択ダイアログを表示しフォルダの場所を指定します。

[書式] External ("FUtl-OpenFolderDialog", 引数)

[引数] **MESSAGE = text**
ダイアログにメッセージを表示します。

[返値] パス

[例題] フォルダ選択ダイアログを表示する例です。メッセージには「Please choose a folder.」を指定します。

← "MESSAGE='Please choose a folder.'"

FUtl-SaveFileDialog

ファイル保存ダイアログを表示しファイルの保存場所を指定します。

[書式] External ("FUtl-SaveFileDialog", 引数)

[引数] **MESSAGE = text**
 ダイアログにメッセージを表示します。

 DEFAULT = text
 ダイアログに表示されるファイル名の初期値を指定します。省略した場合は空欄になります。

[返値] パス

[例題] ファイル保存ダイアログを表示する例です。メッセージには「Please enter file name.」を指定します。

← "MESSAGE='Please enter file name.'"

FUtl-FindFile（最初の検索）

ファイルを検索します。

条件を指定しファイルを検索します。一致するファイルがなければ「File not found」のエラーを返します。また繰り返し検索を実行し条件に一致するファイルを次々検索することもできます。

繰り返し検索については「FUtl-FindFile（繰り返し検索）」コマンドの説明を御覧ください。

[書式] External ("FUtl-FindFile", 引数)

[引数]

NAME = text

ファイル名を指定します。

EXTENSION = 拡張子

拡張子をひとつ指定します。

VOLUME = text

ボリューム名を指定します。この場合はボリューム名の最後にコロン ("¥¥") は必要ありません。

LOCALONLY [= boolean]

ローカルのボリュームだけを検索対象にします。省略した場合はサーバボリュームを含めたすべてのボリュームが検索対象となります。ボリューム名が指定されている場合はLOCALONLYの指定は無効になります。

[返値] パス

[メモ] NAME と EXTENSION の 3 つの引数は絞り込み (AND) 条件として検索されます。

256 バイトを超えるパスの取得はできません。

[例題] 「D:」 ボリュームから拡張子が「JPG」のファイルを検索する例です。

← "VOLUME='D:' EXTENSION='JPG'"

[参照] FUtl-FindFile（繰り返し検索） P.54

FUtl-FindFile（繰り返し検索）

前回の入力条件に従って次のファイルを検索します。

繰り返し検索を実行し条件に一致するファイルを次々検索することもできます。一致するファイルがなければ-43（File not found）のエラーを返します。
検索を連続して実行する場合には、引数をブランクにすることで前回と同じ条件として再検索します。

[書式] External ("FUtl-FindFile", 引数)

[引数] なし

[返値] パス

[例題] フィールド設定 [「FileList」, 「"」]
フィールド設定 [「gResult」, 「External("FUtl - FindFile","VOLUME='D:'
EXTENSION=JPG")」]
フィールド設定 [「gError」, 「External("FUtl - Tag","TAG=ERROR " & gResult)」]
Loop
Exit Loop If [「gError <> "」]
フィールド設定 [「FileList」, 「FileList & gResult & " ¶ "」]
フィールド設定 [「gResult」, 「External("FUtl - FindFile","")」]
フィールド設定 [「gError」, 「External("FUtl - Tag","TAG=ERROR " & gResult)」]
End Loop

[参照] FUtl-FindFile（最初の検索） P.53

FUtl-VolumeCount

ボリュームの個数を調べます。

[書式] External ("FUtl-VolumeCount", 引数)

[引数] なし

[返値] ボリュームの個数

[参照] FUtl-GetVolume P.56

FUtl-GetVolume

n 番目のボリューム名を取得します。

[書式] External ("FUtl-GetVolume", 引数)

[引数] **INDEX = integer**

ボリュームのインデックス (1 ～ N) を指定します。この引数は省略できません。

.....

[返値] パス

パスの文字列の最後にはコロン (":") が付きます。

[例題] 5 番目のボリューム名を取得する例です。

← "INDEX=5"

[参照] FUtl-VolumeCount P.55

FUtl-FolderItemCount

指定したフォルダ内のアイテム（ファイルやフォルダ）の個数を取得します。

[書式] External ("FUtl-FolderItemCount", 引数)

[引数] **PATH = path**
フォルダのパスを指定します。文字列の最後には円マーク ("¥¥") が必要です。
この引数は省略できません。

.....

[返値] アイテムの個数

非表示アイテムもカウントに含まれます。

[例題] 「D:¥¥IMAGE¥¥」フォルダのアイテム数を取得する例です。

← "PATH='D:¥¥IMAGE¥¥'"

[参照] FUtl-GetFolderItem P.58

FUtl-GetFolderItem

指定したフォルダ内の n 番目のアイテムの名前を取得します。

[書式] External ("FUtl-GetFolderItem", 引数)

[引数]

PATH = path

フォルダのパスを指定します。この引数は省略できません。

.....

INDEX = integer

アイテムのインデックス (1 ~ N) を指定します。この引数は省略できません。

.....

[返値]

アイテムの名前

アイテムがフォルダの場合、名前の最後には円マーク ("¥¥") が付きます。

[例題]

「D:¥¥IMAGE¥¥」フォルダの 2 番目のアイテムの名前を取得する例です。

← "PATH='D:¥¥IMAGE¥¥' INDEX=2"

[参照]

FUtl-FolderItemCount P.57

FUtl-FileInfo（基本情報）

アイテム（ファイル、ショートカット、フォルダ、ボリューム）の情報を調べます。

〔書式〕 External ("FUtl-FileInfo", 引数)

〔引数〕 **PATH = path**

調べるアイテムのパスを指定します。この引数は省略できません。

.....

〔返値〕

基本情報には以下の内容が含まれます。

ファイル名、種類、含まれるアイテム数、非表示設定、サイズ、空き容量、ファイルロック設定、イジェクダブル、拡張子、データサイズ、作成日、作成時間、修正日、修正時間

詳細情報については、「FUtl-FileInfo（詳細情報）」コマンドの説明を御覧ください。

アイテムの種類がファイル、ショートカット、フォルダ、ボリュームかによって返される情報が異なります。

NAME = text

ファイル名を返します。

ファイル：○ ショートカット：○ フォルダ：○ ボリューム：○

.....

KIND = text

種類を返します。

ファイル：○ ショートカット：○ フォルダ：○ ボリューム：○

.....

ITEMCOUNT = integer

その中に含まれるアイテム数を返します。

ファイル：－ ショートカット：－ フォルダ：○ ボリューム：○

.....

INVISIBLE = boolean

非表示設定の状態を返します。

ファイル：○ ショートカット：○ フォルダ：○ ボリューム：－

.....

SIZE = integer

ボリュームのサイズを返します。

ファイル：○ ショートカット：－ フォルダ：－ ボリューム：○

.....

FREE = integer

ボリュームの空き容量を返します。

ファイル：－ ショートカット：－ フォルダ：－ ボリューム：○

LOCKED = boolean

ファイルロック設定の状態を返します。

ファイル：○ ショートカット：○ フォルダ：○ ボリューム：－

EJECTABLE = boolean

イジェクトできるかどうかを返します。

ファイル：－ ショートカット：－ フォルダ：－ ボリューム：○

EXTENSION = 拡張子

ファイルタイプを返します。

ファイル：○ エイリアス：－ フォルダ：－ ボリューム：－

DATASIZE = integer

データフォークのサイズを返します。

ファイル：○ ショートカット：－ フォルダ：－ ボリューム：－

CREATEDDATE = date

作成日を返します。

ファイル：○ ショートカット：○ フォルダ：○ ボリューム：－

CREATEDTIME = time

作成時間を返します。

ファイル：○ ショートカット：○ フォルダ：○ ボリューム：－

MODIFIEDDATE = date

修正日を返します。

ファイル：○ ショートカット：○ フォルダ：○ ボリューム：－

MODIFIEDTIME = time

修正時間を返します。

ファイル：○ ショートカット：○ フォルダ：○ ボリューム：－

基本情報のアイテムごとに返す情報を表にまとめました。

	File	Shortcut	Folder	Volume
NAME	○	○	○	○
KIND	○	○	○	○
ITEMCOUNT	—	—	○	○
INVISIBLE	○	○	○	—
SIZE	○	—	—	○
FREE	—	—	—	○
LOCKED	○	○	○	—
EJECTABLE	—	—	—	○
EXTENSION	○	—	—	—
DATASIZE	○	—	—	—
CREATEDDATE	○	○	○	—
CREATEDTIME	○	○	○	—
MODIFIEDDATE	○	○	○	—
MODIFIEDTIME	○	○	○	—

【例題】 「D:¥¥IMAGE¥¥Sample.JPG」 ファイルの基本情報を取得する例です。

← "PATH='D:¥¥IMAGE¥¥Sample.JPG'"

→ "NAME='Sample.jpg' KIND=file SIZE=20084 DATASIZE=20084 EXTENSION='.jpg'
 LOCKED=False INVISIBLE=False CREATEDDATE=2001.4.21 CREATEDTIME=14:54:38
 MODIFIEDDATE=2000.3.4 MODIFIEDTIME=18:45:40"

【参照】 FUtI-FileInfo（詳細情報） P.62
 FUtI-SetFileInfo P.64

FUtl-FileInfo（詳細情報）

アイテム（ファイル、ショートカット、フォルダ、ボリューム）の詳細情報を調べます。

[書式] External ("FUtl-FileInfo", 引数)

[引数]

PATH = path

調べるアイテムのパスを指定します。この引数は省略できません。

EXTRA [= boolean]

詳細情報を調べる場合に指定します。この引数を省略すると基本情報だけが返されます。

[返値]

詳細情報には、すべての基本情報に加えて以下の内容が含まれます。

ファイルシステムの種類、サーバー名、ユーザ名、ロケーション、パス、オリジナルのパス

基本情報については、「FUtl-FileInfo（基本情報）」コマンドの説明を御覧ください。

アイテムの種類がファイル、ショートカット、フォルダ、ボリュームかによって返される情報が異なります。

FILESYSTEM = text

ファイルシステムの種類を返します。対象となるボリュームによって "FAT"、"FAT32"、"CDFS"、"NTFS" といった値を返します。

ファイル：－ ショートカット：－ フォルダ：－ ボリューム：○

SERVER = text

ボリュームがサーバの場合には、サーバーのコンピュータ名を返します。

ファイル：－ ショートカット：－ フォルダ：－ ボリューム：○

USER = text

ボリュームがサーバの場合には、ユーザのコンピュータ名を返します。ただし、ゲストでアクセスしている場合は、ユーザ名は返りません。

ファイル：－ ショートカット：－ フォルダ：－ ボリューム：○

LOCATION = text

ロケーションを返します。

ファイル：○ ショートカット：○ フォルダ：○ ボリューム：－

PATH = path

パスを返します。

ファイル：○ ショートカット：○ フォルダ：○ ボリューム：○

.....

ORIGINAL = text

アイテムがエイリアスの場合には、そのオリジナルのパスを返します。

ファイル：－ ショートカット：○ フォルダ：－ ボリューム：－

.....

詳細情報のアイテムごとに返す情報を表にまとめました。

	File	Shortcut	Folder	Volume
FILESYSTEM	－	－	－	○
SERVER	－	－	－	○
USER	－	－	－	○
LOCATION	○	○	○	－
PATH	○	○	○	○
ORIGINAL	－	○	－	－

[例題] 「Document:Image01:A001」 ファイルの基本情報だけでなく詳細情報も取得する例です。

← "PATH='D:¥¥IMAGE¥¥Sample.JPG' EXTRA"

→ "NAME='Sample.jpg' KIND=file SIZE=20084 DATASIZE=20084 EXTENSION='jpg'
 LOCKED=False INVISIBLE=False CREATEDDATE=2001.4.21 CREATEDTIME=14:54:38
 MODIFIEDDATE=2000.3.4 MODIFIEDTIME=18:45:40 LOCATION='D:¥¥IMAGE¥¥'
 PATH='D:¥¥IMAGE¥¥Sample.JPG'"

[参照] FUtl-FileInfo（基本情報） P.59

FUtl-SetFileInfo P.64

FUtl-SetFileInfo

ファイル情報を変更します。

[書式] External ("FUtl-SetFileInfo", 引数)

[引数]

PATH = path
対象となるファイル、フォルダ、ボリュームのパスを指定します。この引数は省略できません。
.....

NAME = text
名前を変更します。
.....

INVISIBLE [= boolean]
非表示の状態を変更します。
.....

LOCKED [= boolean]
ファイルのロックの状態を変更します。
.....

EXTENSION = 拡張子
拡張子を変更します。
.....

MODIFIEDDATE = date
修正日を変更します。
.....

MODIFIEDTIME = time
修正時間を変更します。
.....

CREATEDDATE = date
作成日を変更します。
.....

CREATEDTIME = time
作成時間を変更します。
.....

TOUCH [= boolean]
修正日を今日の日付けに設定し、修正時間を現在の時刻に変更します。この指定がある場合、MODIFIEDDATE、MODIFIEDTIME、CREATEDDATE、CREATEDTIME のタグは無視されます。
.....

[返値] 実行結果

【例題】 「D:¥¥IMAGE¥¥Sample.JPG」 ファイルにロックをかける例です。

← "PATH='D:¥¥IMAGE¥¥Sample.JPG' LOCKED"

【メモ】 複数のタグを一度に指定できます。タグが指定されていないものは変更されません。

【参照】 FUtl-FileInfo（基本情報） P.59
 FUtl-FileInfo（詳細情報） P.62

FUtl-CopyFile

指定のアイテム（ファイルやフォルダ）をコピーします。

【書式】 External ("FUtl-CopyFile", 引数)

【引数】 **SRC = path**
 コピーの元になるファイルやフォルダを指定します。この引数は省略できません。

 DEST = path
 コピー先の場所を指定します。フォルダを指定するとその場所にコピーされます。ファイル名まで指定すると、SRC と別名でコピーすることができます。この引数は省略できません。

【返値】 実行結果

コピー先に同じ名称がすでに存在すれば重複エラーになります。

【注意】 対象がフォルダの場合には、名前を変更してコピーすることはできません。

【例題】 「D:¥¥IMAGE¥¥Sample.JPG」 ファイルを「E:¥¥PJ001¥¥IMAGE¥¥」 フォルダにコピーする例です。

← "PATH='D:¥¥IMAGE¥¥Sample.JPG' DEST='E:¥¥PJ001¥¥IMAGE¥¥'"

「D:¥¥IMAGE¥¥Sample.JPG」 ファイルを「E:¥¥PJ001¥¥IMAGE¥¥」 フォルダに「01A001.JPG」というファイル名でコピーする例です。

← "PATH='D:¥¥IMAGE¥¥Sample.JPG' DEST='E:¥¥PJ001¥¥IMAGE¥¥01A001.JPG'"

FUtl-MoveFile

ファイルやフォルダなど、指定のアイテムを同じボリューム内に移動します。

[書式] External ("FUtl-MoveFile", 引数)

[引数]

SRC = path

移動するファイルやフォルダを指定します。この引数は省略できません。

DEST = path

移動先の場所を指定します。フォルダを指定すると、その場所に移動されます。ファイル名まで指定すると、SRC と別名で移動することができます。この引数は省略できません。

[返値]

実行結果

移動先に同じ名称がすでに存在すれば重複エラーになります。

[注意]

SRC と DEST は、同じボリュームである必要があります。

[例題]

「D:¥¥IMAGE¥¥A001.JPG」ファイルを「D:¥¥SELECTED¥¥」フォルダに移動する例です。

← "PATH='D:¥¥IMAGE¥¥A001.JPG' DEST='D:¥¥SELECTED¥¥'"

「D:¥¥IMAGE¥¥A001.JPG」ファイルを「D:¥¥SELECTED¥¥」フォルダに「01A001.JPG」というファイル名で移動する例です。

← "PATH='D:¥¥IMAGE¥¥A001.JPG' DEST='D:¥¥SELECTED¥¥01A001.JPG'"

FUtl-NewFolder

フォルダを作成します。

[書式] External ("FUtl-NewFolder", 引数)

[引数] **PATH** = *path*

作成する場所を指定します。この引数は省略できません。

.....

NAME = *text*

新しく作るフォルダ名を指定します。この引数は省略できません。

.....

[返値] 実行結果

[例題] 「D:¥¥PART01¥¥」フォルダに「IMAGE」という名前のフォルダを作成する例です。

← "PATH='D:¥¥PART01¥¥' NAME='IMAGE'"

FUtl-DeleteFile

ファイルまたはフォルダを削除します。

[書式] External ("FUtl-DeleteFile", 引数)

[引数] **PATH = path**

削除するファイルまたはフォルダを指定します。この引数は省略できません。

.....

[注意] フォルダに含まれるファイルも削除されます。

[返値] 実行結果

[例題] 「D:¥¥IMAGE¥¥A001.JPG」 ファイルを削除する例です。

← "PATH='D:¥¥IMAGE¥¥A001.JPG'"

FUtl-EjectVolume

ボリュームをイジェクトします。

[書式] External ("FUtl-EjectVolume", 引数)

[引数] **VOLUME** = *path*

ボリュームを指定します。この引数は省略できません。

.....

[返値] 実行結果

オープンしているものがある場合はエラーになります。

[例題] 「F:」 ボリュームの CD-ROM をイジェクトする例です。

← "VOLUME='F:'"

FUtl-LaunchApp (アプリケーション)

アプリケーションを起動します。同時に書類を指定することもできます。

[書式] External ("FUtl-LaunchApp", 引数)

[引数] **APP = path**

アプリケーションのパスを指定します。この引数は省略できません。

.....

DOC = path

ドキュメントのパスを指定します。

.....

[返値] 実行結果

[例題] アプリケーション「PictureViewer.exe」で、書類「Sample.JPG」を開く例です。

```
← "APP='C:¥¥Program Files¥¥QuickTime¥¥PictureViewer.exe'
DOC='D:¥¥DATA¥¥Sample.JPG'"
```

[メモ] アプリケーションによっては、ドキュメントのパスにスペースが含まれていると開くことができないものがあります。フォルダやファイルの名称の付け方には注意が必要です。

[参照] FUtl-LaunchApp (書類) P.72
FUtl-QuitApp P.73

FUtl-LaunchApp（書類）

書類を開きます。

【書式】 External ("FUtl-LaunchApp", 引数)

【引数】 **DOC = path**
ドキュメントのパスを指定します。登録されているアプリケーションを起動します。この引数は省略できません。
.....

【返値】 実行結果

【例題】 例えば、JPG の拡張子をアプリケーション「PictureViewer.EXE」で登録している場合は、書類「A001.JPG」は「PictureViewer.EXE」で開かれます。

← "DOC='D:¥¥DATA¥¥A001.JPG'"

【参照】 FUtl-LaunchApp（アプリケーション） P.71
FUtl-QuitApp P.73

FUtl-QuitApp

アプリケーションを終了します。

[書式] External ("FUtl-QuitApp", 引数)

[引数] **APP = path**
アプリケーションを指定します。この引数は省略できません。

.....

[返値] 実行結果

[例題] アプリケーション「NOTEPAD.EXE」を終了する例です。

← "APP='C:¥¥WINDOWS¥¥NOTEPAD.EXE'"

[参照] FUtl-LaunchApp（アプリケーション） P.71
FUtl-LaunchApp（書類） P.72

FUtl-PickColor

色選択ダイアログを表示します。

[書式] External ("FUtl-PickColor", 引数)

[引数] **MESSAGE = text**
 ダイアログに表示するメッセージを指定します。省略した場合には「色を選択:」となります。

COLOR = RGBColor
 初期値の色を指定します。

[返値] 選択した色

[例題] 色選択ダイアログを表示する例です。メッセージには「Please choose color.」を指定します。
 また初期値の色を「グレー (127,127,127)」に指定します。

← "MESSAGE='Please choose color.' COLOR=127,127,127"

FUtl-CalcColor

色に関する計算を行います。

[書式] External ("FUtl-CalcColor", 引数)

[引数]

COLOR = RGBColor

計算の対象となる色を指定します。この引数は省略できません。

.....

RED

指定した色の値から赤の値を返します。

→ *integer*

.....

GREEN

指定した色の値から緑の値を返します。

→ *integer*

.....

BLUE

指定した色の値から青の値を返します。

→ *integer*

.....

RED = integer

指定した色の赤の値を置き換えた色を返します。

→ *RGBColor*

.....

GREEN = integer

指定した COLOR の緑の値を置き換えた色を返します。

→ *RGBColor*

.....

BLUE = integer

指定した色の青の値を置き換えた色を返します。

→ *RGBColor*

.....

INVERT

指定した色を反転した色を返します。

→ *RGBColor*

.....

LIGHTEN

指定した色を明るくした色を返します。

→ *RGBColor*

.....

DARKEN

指定した色を暗くした色を返します。

→ *RGBColor*

SAMPLE [= point]

指定した色のカラーサンプルの画像をクリップボードに返します。*point*の指定を省略した場合は、16 x 16 ピクセルのサンプルが作成されます。*point*を指定した場合には、その大きさのサンプルが作成されます。

→ *ColorSample*

[返値]

色、赤の値、緑の値、青の値、カラーサンプル

カラーサンプルはクリップボードに返ります。

返値は引き数の指定によって変わります。

引数ごとに返す情報の内容を表にまとめました。

引数	値	返値
RED	—	integer
GREEN	—	integer
BLUE	—	integer
RED	integer	RGBColor
GREEN	integer	RGBColor
BLUE	integer	RGBColor
INVERT	—	RGBColor
LIGHTEN	—	RGBColor
DARKEN	—	RGBColor
SAMPLE	point	colorSample

【例題】 色「20,50,100」の赤の値を取得する例です。

← "COLOR=20,50,100 RED"

→ "20"

色「20,50,100」の青の値を 200 に変更する例です。

← "COLOR=20,50,100 BLUE=200"

→ "20, 50, 200"

色「20,50,100」のサンプルをクリップボードに取り込む例です。サンプルのサイズは「20,20」に指定します。

← "COLOR=20,50,100 SAMPLE=20,20"

FUtl-CalcPoint

ポイントに関する計算を行います。

[書式] External ("FUtl-CalcPoint", 引数)

[引数]

POINT = point

計算の対象となるポイントを指定します。この引数は省略できません。

.....

H

指定したポイントの横の値を返します。

→ **integer**

.....

V

指定したポイントの縦の値を返します。

→ **integer**

.....

H = integer

指定したポイントの横の値を置き換えたポイントを返します。

→ **point**

.....

V = integer

指定したポイントの縦の値を置き換えたポイントを返します。

→ **point**

.....

INSIDE = rect

指定したポイントがレクタングルの内側に位置するかどうかを調べます。

→ **boolean**

.....

ADD = point

指定したポイントに加算します。

→ **point**

.....

SUB = point

指定したポイントに減算します。

→ **point**

.....

[返値]

横の値、縦の値、ポイント、真偽値

引数によって返される内容が変わります。

引数ごとに返す情報の内容を表にまとめました。

引数	値	返値
H	—	integer
V	—	integer
H	integer	point
V	integer	point
INSIDE	rect	boolean
ADD	point	point
SUB	point	point

【例題】

現在のポイント「20,50」の横の値を「100」に変更する例です。

← "POINT=20,50 H=100"

→ "100, 50"

ポイント「50,70」がレクタングル「20,20,400,200」に含まれるかどうか調べる例です。

← "POINT=50,70 INSIDE=20,20,400,300"

→ "True"

FUtl-CalcRect

レクタングルに関する計算を行います。

[書式] External ("FUtl-CalcRect", 引数)

[引数]

RECT = rect

計算の対象となるレクタングルを指定します。この引数は省略できません。

.....
LEFT

指定したレクタングルの左の値を返します。

→ *integer*
.....

TOP

指定したレクタングルの上の値を返します。

→ *integer*
.....

RIGHT

指定したレクタングルの右の値を返します。

→ *integer*
.....

BOTTOM

指定したレクタングルの下の値を返します。

→ *integer*
.....

TOPLEFT

指定したレクタングルの左上の値を返します。

→ *point*
.....

BOTTOMRIGHT

指定したレクタングルの右下の値を返します。

→ *point*
.....

WIDTH

指定したレクタングルの幅を返します。

→ *integer*
.....

HEIGHT

指定したレクタングルの高さを返します。

→ *integer*
.....

LEFT = integer

指定したレクタングルの左の値を置き換えたレクタングルを返します。

→ *rect*

.....

TOP = integer

指定したレクタングルの上の値を置き換えたレクタングルを返します。

→ *rect*

.....

RIGHT = integer

指定したレクタングルの右の値を置き換えたレクタングルを返します。

→ *rect*

.....

BOTTOM = integer

指定したレクタングルの下の値を置き換えたレクタングルを返します。

→ *rect*

.....

TOPLEFT = point

指定したレクタングルの左上の値を置き換えたレクタングルを返します。

→ *rect*

.....

BOTTOMRIGHT = point

指定したレクタングルの右下の値を置き換えたレクタングルを返します。

→ *rect*

.....

WIDTH = integer

指定したレクタングルの幅が指定の値になるように、右の値を置き換えたレクタングルを返します。

→ *rect*

.....

HEIGHT = integer

指定したレクタングルの高さが指定の値になるように、下の値を置き換えたレクタングルを返します。

→ *rect*

.....

INSIDE = rect

このレクタングルが指定したレクタングルに含まれるかどうかを調べます。

→ *boolean*

.....

OFFSET = point

指定したレクタングルをポイント分右下に移動したレクタングルを返します。

→ *rect*

.....

INSET = integer | point

指定したレクタングルの縦と横を内側に小さくしたレクタングルを返します。負の値の場合は外側に大きくなります。

*integer*の場合は、縦と横の値が同じになります。*point*の場合は指定したレクタングルの縦と横の値が個別に指定できます。

→ *rect*

.....

UNION = rect

指定したレクタングルとこのレクタングルの二つが納まるレクタングルを返します。

→ *rect*

.....

INTERSECT = rect

指定したレクタングルとこのレクタングルの共通部分を返します。ただし、共通部分がない場合は空のレクタングルを返します。

→ *rect*

.....

[返値] 左の値、上の値、右の値、下の値、レクタングル、ポイント、幅、高さ、真偽値

引数ごとに返す情報の内容を表にまとめました。

引数	値	返値
LEFT	—	integer
TOP	—	integer
RIGHT	—	integer
BOTTOM	—	integer
TOPLEFT	—	point
BOTTOMRIGHT	—	point
WIDTH	—	integer
HEIGHT	—	integer
LEFT	integer	rect
TOP	integer	rect
RIGHT	integer	rect
BOTTOM	integer	rect
TOPLEFT	point	rect
BOTTOMRIGHT	point	rect
WIDTH	integer	rect
HEIGHT	integer	rect
INSIDE	rect	boolean
OFFSET	point	rect
INSET	integer point	rect
UNION	rect	rect
INTERSECT	rect	rect

[例題] レクタングル「20,20,400,200」の幅の値を取得する例です。

← "RECT=20,20,400,200 WIDTH"

→ "380"

レクタングル「20,20,400,200」を「50,50」オフセットする例です。

← "RECT=20,20,400,200 OFFSET=50,50"

→ "70, 70, 450, 250"

レクタングル「20,20,400,200」と「100,100,380,250」の重なる部分のレクタングルを取得する例です。

← "RECT=20,20,400,200 INTERSECT=100,100,400,250"

→ "100, 100, 400, 200"

FUtl-OpenProgress

プログレスバーの表示を開始します。

プログレスバーを表示するときには必ず最初にこの関数を呼ぶ必要があります。

プログレスバーはダイアログに表示する方法とウィンドウの左下に表示する二つの表示方法が選べます。また通常のプログレスバーに加え、全体の仕事量がわからないときに使う「バーバーボール」型のプログレスバーも表示できます。

[書式] External ("FUtl-OpenProgress", 引数)

[引数] 「通常のプログレスバー」の場合の引数

MIN = integer

プログレスバーの最小値を指定します。

.....

MAX = integer

ログレスバーの最大値を指定します。

.....

VALUE = integer

プログレスバーの現在値を指定します。

.....

「バーバーボールのプログレスバー」の場合の引数

BARBER [= boolean]

プログレスバーの種類を「バーバーボール」に指定します。バーバーボールのプログレスバーにするときは省略できません。

.....

「ダイアログ」に表示する場合の引数

DIALOG [= boolean]

プログレスバーをダイアログに表示するかどうかを指定します。

.....

TITLE = text

ダイアログに表示するタイトルを指定いします。

.....

MESSAGE = text

ダイアログに表示するメッセージを指定します。

.....

[返値] なし

[注意] FUtl-OpenProgressが成功した場合には、必ずFUtl-CloseProgressでプログレスバーの表示を終了させてください。プログレスバーの処理が正常に終了しない場合、表示が残ったり、ダイアログが消えないなどの不都合が起こります。

ファイルメーカーProの「フィールドにペースト」スクリプトは、データベースウィンドウ以外のウィンドウが前面にある場合、正しく動作しません。従って、フィールドにペーストする必要がある場合には、プログレスバーウィンドウも「ダイアログに表示」を選ばず、ウィンドウの左下に表示する方式を使うようにしてください。

[例題] 通常のプログレスバーをダイアログに表示する場合の例です。

フィールド設定 [「gValue」,「""]]

フィールド設定 [「gResult」,「External("FUtl - OpenProgress","DIALOG TITLE=' 処理中 ' MESSAGE=' 少々お待ちください ...' MIN=0 MAX=1000")」]

Loop

Exit Loop If [「TextToNum(gValue) > 1000」]

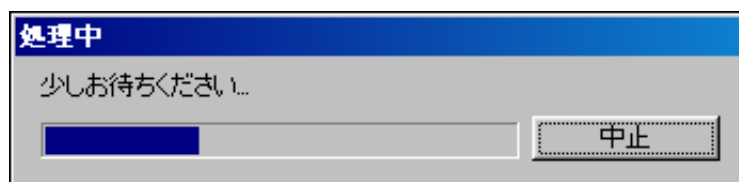
フィールド設定 [「gValue」,「NumToText(TextToNum(gValue) + 1)」]

フィールド設定 [「gResult」,「External("FUtl - UpdateProgress","VALUE=" & gValue)」]

Exit Loop If [「gResult <> ""」]

End Loop

フィールド設定 [「gResult」,「External("FUtl - CloseProgress","")」]



バーバーボールをダイアログに表示する場合の例です。

フィールド設定 [「gValue」,「"」]

フィールド設定 [「gResult」,「External("FUtl - OpenProgress","DIALOG TITLE=' 処理中 ' MESSAGE=' 少しお待ちください ...' BARBER")」]

Loop

Exit Loop If [「TextToNum(gValue) > 1000」]

フィールド設定 [「gValue」,「NumToText(TextToNum(gValue) + 1)」]

フィールド設定 [「gResult」,「External("FUtl - UpdateProgress","")」]

Exit Loop If [「gResult <> ""」]

End Loop

フィールド設定 [「gResult」,「External("FUtl - CloseProgress","")」]



【参照】

FUtl-UpdateProgress P.88

FUtl-CloseProgress P.89

FUtl-UpdateProgress

表示されているプログレスバーの内容を更新します。

FUtl-OpenProgressでプログレスバーを表示したあとは、定期的にこの関数を呼んでプログレスバーの内容を更新します。

[書式] External ("FUtl-UpdateProgress", 引数)

[引数] 「FUtl-OpenProgress」コマンドの説明を御覧ください。

[返値] なし

[メモ] FUtl-OpenProgressで指定した引数は、すべて更新する事ができます。ただし、DIALOG引数だけは途中で変更することはできません。

[例題] FUtl-OpenProgress の例題をご覧ください。

[参照] FUtl-OpenProgress P.85
FUtl-CloseProgress P.89

FUtl-CloseProgress

プログレスバーの表示を終了します。

[書式] External ("FUtl-CloseProgress", 引数)

[引数] なし

[返値] なし

[メモ] FUtl-OpenProgressが成功した場合には、必ずFUtl-CloseProgressでプログレスバーの表示を終了させてください。プログレスバーの処理が正常に終了しない場合、表示が残ったり、ダイアログが消えないなどの不都合が起こります。

[例題] FUtl-OpenProgress の例題をご覧ください。

[参照] FUtl-OpenProgress P.85
FUtl-UpdateProgress P.88

FUtl-Answer

メッセージをダイアログに表示します。

ダイアログには複数のボタンを表示することができ、押されたボタン名は返値で判断することができます。

[書式] External ("FUtl-Answer", 引数)

[引数]

MESSAGE = text

メッセージを指定します。この引数は省略できません。

.....

ICON = Stop | Note | Caution

アイコンの種類を指定します。省略した場合は、アイコンは表示されません。

.....

DESCRIPTION = text

より詳細な内容のメッセージを指定します。

.....

OK = text

OK ボタンのタイトルを指定します。省略した場合は「OK」になります。

.....

CANCEL [= text]

CANCEL ボタンのタイトルを指定します。文字列を省略した場合は「キャンセル」になります。タグを省略した場合、CANCEL ボタンは表示されません。

.....

OTHER [= text]

OTHER ボタンのタイトルを指定します。文字列を省略した場合は「その他」になります。タグを省略した場合、OTHER ボタンは表示されません。

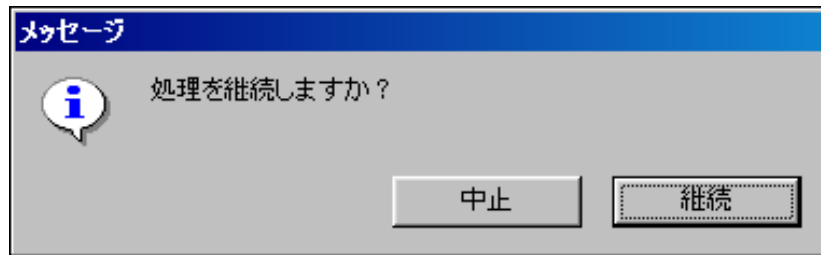
.....

[返値] ボタンの種類 (OK / CANCEL / OTHER)、エラー

[メモ] OK ボタンは省略しても表示されます。

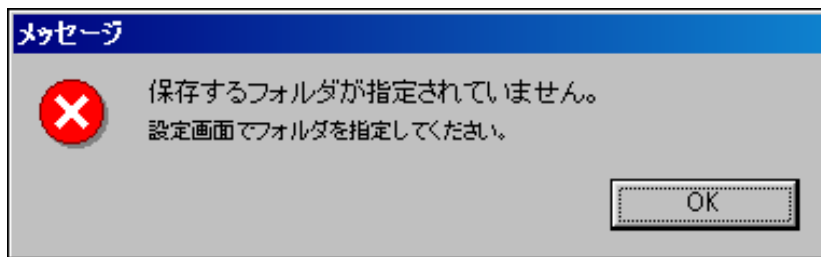
[例題] メッセージには「処理を継続しますか？」を表示し、OKボタンの名称には「継続」、CANCEL ボタンの名称を「中止」にする例です。

← "MESSAGE=' 処理を継続しますか？ ' ICON=Note OK=' 継続 ' CANCEL=' 中止 '"



メッセージには「保存するフォルダが指定されていません。」、詳細なメッセージには「設定画面でフォルダを指定してください。」を表示する例です。

← "MESSAGE=' 保存するフォルダが指定されていません。' ICON=Stop DESCRIPTION=' 設定画面でフォルダを指定してください。' "



FUtl-Ask

簡単なメッセージを伴った入力用ダイアログを表示し、入力された文字列を返します。

[書式] External ("FUtl-Ask", 引数)

[引数]

MESSAGE = text

メッセージを指定します。この引数は省略できません。

.....

DEFAULT = text

入力欄の表示する文字列の初期値を指定します。

.....

PASSWORD [= boolean]

入力された文字列を「*」に置き換えて表示します。

.....

NEEDSOMETHING [= boolean]

何も入力されていないときはOK ボタンを押せなくします。

.....

OK = text

OK ボタンのタイトルを指定します。省略した場合は「OK」になります。

.....

CANCEL = text

CANCEL ボタンのタイトルを指定します。省略した場合は「キャンセル」になります。

.....

[返値]

入力された文字列

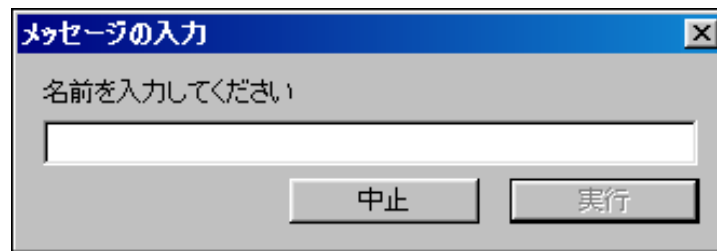
OK ボタンがクリックされときに入力された文字列が返ります。

Cancel ボタンがクリックされたときには、エラー -128 が返ります。

[例題]

メッセージには「名前を入力してください。」を表示し、OK ボタンの名称には「実行」、CANCEL ボタンの名称を「中止」にします。さらに、入力がない場合はOK ボタンを押せないようにする例です。

← "MESSAGE='名前を入力してください。' NEEDSOMETHING OK='実行' CANCEL='中止'"



付録 1 : コマンド一覧

Image

External("FImg-Version", 引数) : Facoty's FileMaker Plug-in のバージョン取得
 External("FImg-DrawImage", 引数) : 画像の描画
 External("FImg-GrabImage", 引数) : 画像データの取得
 External("FImg-SaveImage", 引数) : 画像データの保存
 External("FImg-ImageInfo", 引数) : 画像情報 (大きさ) の取得
 External("FImg-PreviewImage", 引数) : スケールした画像を取得
 External("FImg-CompressImage", 引数) : 画像の圧縮
 External("FImg-CompressImageDialog", 引数) : 圧縮設定
 External("FImg-CheckArea", 引数) : 再表示のタイミング判定
 External("FImg-Effect", 引数) : 描画エフェクト
 External("FImg-GetMousePos", 引数) : クリック位置の取得

Utility

External("FUtl-Version", 引数) : Facoty's FileMaker Plug-in のバージョン取得
 External("FUtl-Tag", 引数) : コマンドの返り値から指定したタグ情報を取得
 External("FUtl-TagList", 引数) : タグ名リストの取得
 External("FUtl-OpenFileDialog", 引数) : ファイルの選択
 External("FUtl-OpenFolderDialog", 引数) : フォルダの選択
 External("FUtl-SaveFileDialog", 引数) : ファイルの保存
 External("FUtl-FindFile", 引数) : ファイルの名前、拡張子などで該当するファイルを検索
 External("FUtl-VolumeCount", 引数) : ボリューム数の取得
 External("FUtl-GetVolume", 引数) : ボリューム名の取得
 External("FUtl-FolderItemCount", 引数) : フォルダ内のアイテム数の取得
 External("FUtl-GetFolderItem", 引数) : フォルダ内のアイテム名の取得
 External("FUtl-FileInfo", 引数) : ファイル情報の取得
 External("FUtl-SetFileInfo", 引数) : ファイルのリネーム
 External("FUtl-CopyFile", 引数) : ファイルやフォルダのコピー
 External("FUtl-MoveFile", 引数) : 同じボリューム内のファイルやフォルダの移動
 External("FUtl-DeleteFile", 引数) : ファイルやフォルダの削除
 External("FUtl-NewFolder", 引数) : フォルダの作成
 External("FUtl-EjectVolume", 引数) : ボリュームのイジェクト
 External("FUtl-PickColor", 引数) : カラーの選択
 External("FUtl-CalcColor", 引数) : カラーの演算
 External("FUtl-CalcPoint", 引数) : 座標の演算
 External("FUtl-CalcRect", 引数) : 四角形の座標の演算
 External("FUtl-OpenProgress", 引数) : プログレスバーのオープン処理
 External("FUtl-UpdateProgress", 引数) : プログレスバーのアップデート処理
 External("FUtl-CloseProgress", 引数) : プログレスバーのクローズ処理
 External("FUtl-Answer", 引数) : メッセージ表示
 External("FUtl-Ask", 引数) : メッセージの表示と文字列の入力
 External("FUtl-LaunchApp", 引数) : アプリケーションのローンチ
 External("FUtl-QuitApp", 引数) : ローンチしたアプリケーションの終了

付録 2：描画モード一覧

FImg-DrawImage、FImg-GrabImage、FImg-SaveImageの引数であるGRAPHICSMODEとOPCOLORは、画像を描画する方法を指定します。これを描画モードと呼びます。描画モードを使えば、画像を合成したり、反転させたり、様々な効果を与えることが可能になります。描画モードはGRAPHICSMODEのタグで種類を指定します。そしていくつかの描画モードはさらにOPCOLORのタグでオペレーションカラーを指定することができます。

正しく描画モードを理解するためには、その前提となるいくつかのことが必要になります。まず描画モードの考え方から説明します。描画モードに影響するのは、表示しようとする元画像と表示される場所の状態です。描画にあたって、元画像の各ピクセルの色を表示場所の色と比較し、描画モードに従った演算を行って新しい色を計算し、それを表示場所書き込みます。文章で書くと難しくなりますが、まとめると、元画像と表示場所が影響しあい結果の画像が表示されるわけです。単純に元画像のみに影響されるわけではないということがポイントです。

描画モードとその効果、そしてOPCOLORの意味を表にまとめました。

srcCopy	表示場所のピクセルに関係なく元画像を表示します。一番基本的な描画モードであり、描画モードを指定しない場合にはこれが使われます。OPCOLORは使いません。
notSrcCopy	元画像を反転してからsrcCopyで表示します。
blend	元画像のピクセルと表示場所のピクセルを比較し、その平均を表示します。効果としては元画像と表示場所を混ぜ合わせた感じになります。OPCOLORは値の平均を出す場合の比率として扱われます。黒に近ければ表示場所が強くなり、白に近付くほど元画像が強くなります。グレーを指定すれば50%のブレンドになります。平均はRGB各色に対して別々に計算されます。
transparent	元画像のなかの指定の色を透明色として扱います。他の色はそのまま表示されます。透明にしたい色はOPCOLORで指定します。
addPin	元画像のピクセルと表示場所のピクセルのRGBそれぞれの値を加えた合計の値を表示します。ただし、OPCOLORで指定した値よりも合計の値が大きくなってしまった場合にはOPCOLORを表示します。
addOver	元画像のピクセルと表示場所のピクセルのRGBそれぞれの値を加えた合計の値を表示します。ただし、合計の値が255よりも大きくなってしまった場合には、値から256を引いて表示します。OPCOLORは使いません。
subPin	表示場所のピクセルの色から元画像のピクセルの色の差分を取り、その色を表示します。ただし、OPCOLORで指定した値よりも差分の値が小さくなってしまった場合にはOPCOLORを表示します。
subOver	表示場所のピクセルの色から元画像のピクセルの色の差分を取り、その色を表示します。ただし、差分の値が0よりも小さくなってしまった場合には、値に256を足して表示します。OPCOLORは使いません。
addMax	元画像のピクセルと表示場所のピクセルを比較し、値の大きいほうを表示します。比較はRGB各色ごとに行われます。OPCOLORは使いません。
addMin	元画像のピクセルと表示場所のピクセルを比較し、値の小さいほうを表示します。比較はRGB各色ごとに行われます。OPCOLORは使いません。

描画モードは実際の効果を確認しながらでないとなかなか理解しづらいものです。付属のサンプルには描画モードを試せるものも入っておりますので、実際に確認しながらお読みいただければ理解も深まることと思います。