

# 4D Pack

-  はじめに
-  4D Pack : ANSIストリーム
-  4D Pack : ピクチャファイル
-  4D Pack : ユーザインタフェース
-  4D Pack : ユーティリティ
-  4D Pack : 印刷
-  コマンドリスト (文字順)

## はじめに

 4D Packの紹介

## 4D Packの紹介

---

4D Packプラグインは、4Dデベロッパに、有用な追加のルーチンを提供します。

4D Pack コマンド名には、4Dコマンドと区別するため、接頭辞“AP”が付けられています。またコマンドか関数かを大文字と小文字で区別しています。

### インストール

---

4D Packプラグインも他の4Dプラグインと同様にインストールします。実行可能アプリケーションまたはストラクチャファイルと同階層に**Plugins**フォルダを置き、その中にプラグインをコピーします。詳細は *4D Design Reference* マニュアルの [プラグインやコンポーネントのインストール](#) を参照してください。

4D v14(Windows)と4D v14 R5(OS X)以降、4D Packプラグインは64-ビット版も提供されています。これによって、64-ビットWindows版の4D Server またはOS X用の64-bit版4D Server の環境設定で64-ビットアーキテクチャの恩恵を受けることができます。64-ビット版は**bundle** に同梱されており、特別なインストールなどは何も必要ありません。

**互換性に関する注:** `_o_AP Save BMP 8 bits` コマンドはOS X用の64-bit版4D Server ではサポートされていません。

### 4D Packから削除されたコマンド

---

テクノロジーの変遷と、4D の新機能の継続的な統合により、一部の4D Pack コマンドが以前のバージョンから削除されています。このセクションでは削除されたコマンドの一覧と、推奨されるそれらの代替案が記載されています。

#### 4D v14から削除されたコマンド

削除されたコマンド	推奨される代替案
AP AVAILABLE MEMORY	<b>GET MEMORY STATISTICS</b>
AP CLOSE HELP, AP HELP	
INDEX, AP HELP ON HELP, AP HELP ON KEY	廃止予定のコマンド - Windows Vista以降、Windows ヘルプアプリケーション (WinHlp32.exe) はWindows に組み込まれていません。
AP Create method, AP Modify method	<pre>METHOD SET CODE ("myMeth";vCode;*) METHOD SET ATTRIBUTE ("myMeth";vInvisible;2;v4DAction;3;vWebService;4;vWSDL;5;vExported;7;vSQL;8;vRemote;1024;vFolderName;*)</pre>
AP Does method exist	<pre>METHOD GET NAMES (\$arrNames;"myMeth") \$exists:=(Size of array (\$arrNames)&gt;0) // -&gt; メソッドが存在すれば True</pre>
AP Get picture type	"ピクチャ" テーマのコマンド
AP Get templates	廃止予定のコマンド
_AP External clock, AP SET CLOCK	TimePicker ウィジェット(4D v14)
AP Rect Dragger	<b>SET DRAG ICON</b>
AP Timestamp to GMT	<pre>\$laDate:=String (Current date;ISO Date GMT;Current time) // "2013-05-06T12:19:23Z" のように返します</pre>
AP GET PARAM, AP SET PARAM	廃止予定のコマンド - これらのコマンドの第二引数は使用されていません。

### 以前のバージョンの4D Packを使用しているユーザへの注意

**AP PICT DRAGGER** -> 4Dに統合されたドラッグ&ドロップ機能を使用する (クリックタイプのイベントの場合、*MouseDown*、*MouseX*、そして*MouseY* システム変数)

**AP PICT UPDATER** と **%AP Pict displayer** -> ピクチャ フィールドを使用する

**AP Read picture BLOB** -> **BLOB TO PICTURE**を使用する

**AP Read picture File** -> **READ PICTURE FILE**を使用する

**AP Save GIF** -> **CONVERT PICTURE + WRITE PICTURE FILE**を使用する

**AP SET PICT MODE** -> **SET FORMAT**を使用する

**AP Select document** -> **Select document**を使用する

**AP Set palette** -> **Select RGB Color**を使用する

**AP Sublaunch** -> **LAUNCH EXTERNAL PROCESS**を使用する

**AP PrintDefault** -> **AP BLOB to print settings + AP Print settings to BLOB**を使用する

**AP Text to PrintRec** -> **AP BLOB to print settings + AP Print settings to BLOB**を使用する

**AP PrintRec to text** -> **AP BLOB to print settings + AP Print settings to BLOB**を使用する

**AP PrValidate** -> **AP BLOB to print settings + AP Print settings to BLOB**を使用する

**AP Toolbar installed** -> **Tool bar height**を使用する (ツールバーが隠されている場合0が返される)  
**AP SET WEB FILTERS** -> **SET DATABASE PARAMETER** と **Get database parameter**を使用する  
**AP ShellExecute** (Mac OS) -> **LAUNCH EXTERNAL PROCESS**を使用する  
**AP Add table and fields** -> 4D SQLエンジンをを使用する  
**AP Create relation** -> 現在代わりのソリューションはありません。

## 4D Pack : ANSIストリーム

 *\_o\_AP FCLOSE*

 *\_o\_AP fopen*

 *\_o\_AP FPRINT*

 *\_o\_AP fread*

## **\_o\_AP FCLOSE**

`_o_AP FCLOSE ( fopenValue )`

引数	型	説明
<code>fopenValue</code>	倍長整数	→ AP <code>fopen</code> から返されたポートまたはファイル参照

### 説明

---

`_o_AP FCLOSE` コマンドは `_o_AP fopen` コマンドで開かれたポートやファイルを閉じます。

`_o_AP fopen ( file | port ; mode ) -> 戻り値`

引数	型	説明
file   port	文字	➔ 開くファイル名またはポート
mode	整数	➔ 開くモード 0 = 書き込みのみ 1 = 読み込みのみ
戻り値	倍長整数	➔ AP FPRINTで使用される値, または ファイルやポートが開けなかった場合0

## 説明

`_o_AP fopen` コマンドは `fopen` ANSI C コマンドを呼び出し、シリアルまたはパラレル (Windows) ポートを開いたり、直接ファイルを作成したりするために使用します。

この関数から返される値は、`_o_AP FPRINT` コマンドを使用してポート経由で書き込みを行ったり、ファイルに書き込みを行ったりするために使用します。開かれたポートやファイルは、`_o_AP FCLOSE` コマンドで閉じます。

**Note:** Mac OSでは、この関数はファイルの作成のみに使用できます。

## 例題 1

パラレルポート上で "hello world" を書き込みます:

```
port:=_o_AP fopen("LPT1";0)
_o_AP FPRINT(port;"hello world")
_o_AP FCLOSE(port)
```

## 例題 2

シリアルポート上で "hello world" を書き込みます:

```
port:=_o_AP fopen("COM1";0)
_o_AP FPRINT(port;"hello world")
_o_AP FCLOSE(port)
```

## 例題 3

ファイルに "hello world" を書き込みます:

```
port:=_o_AP fopen("C:\MyDoc.txt";0)
_o_AP FPRINT(port;"hello world")
_o_AP FCLOSE(port)
```

## 例題 4

ファイルを読み込みます:

```
port:=_o_AP fopen("C:\MyDoc.txt";1)
MyVar:=_o_AP fread(port)
_o_AP FCLOSE(port)
```

## ⚙️ **\_o\_AP FPRINT**

`_o_AP FPRINT ( fopenValue ; data )`

引数	型	説明
<code>fopenValue</code>	倍長整数	→ AP <code>fopen</code> から返されたポートまたはファイル参照
<code>data</code>	テキスト	→ ポートやファイルに送信するデータ

### 説明

---

**\_o\_AP FPRINT** コマンドは 指定された`data`テキストを、 **\_o\_AP fopen** コマンドで開かれた、参照が`fopenValue`であるポートまたはファイルに送信します。

`data` は変換されずにそのまま送信されます。4DはMacintoshのASCII テーブルを内部的に使用しているため、Windows上では開発者が適切な変換を行わなければなりません。これを行うには **\_o\_Mac to Win** コマンドを使用できます。

## ⚙️ **\_o\_AP fread**

`_o_AP fread ( fopenValue ) -> 戻り値`

引数	型	説明
<code>fopenValue</code>	倍長整数	→ AP <code>fopen</code> から返されたポートまたはファイル参照
戻り値	テキスト	↩️ ポートまたはファイルからのデータ

### 説明

---

`_o_AP fread` は、`_o_AP fopen` で開かれた、参照が `fopenValue` であるポートやファイルからデータを読み込み、テキストとして返します。

`data` は変換されずにそのまま返されます。4DはMacintoshのASCII テーブルを内部的に使用しているため、Windows上では開発者が適切な変換を行わなければなりません。これを行うには `_o_Win to Mac` コマンドを使用できます。

# 4D Pack : ピクチャファイル

 AP Is Picture Deprecated

 *\_o\_AP Save BMP 8 bits*

## 🔧 AP Is Picture Deprecated

AP Is Picture Deprecated ( ptrPict ) -> 戻り値

引数	型	説明
ptrPict	ポインター	→ テストするピクチャーフィールドや変数へのポインター
戻り値	倍長整数	↩ 0 = 有効なフォーマット、1 = 無効なフォーマット (PICT)

### 説明

**AP Is Picture Deprecated** コマンドは *ptrPict* 引数により参照されるピクチャーが (古いフォーマットである) PICT フォーマットを含む場合 1 を、そうでなければ 0 を返します。ピクチャーが PICT フォーマットである場合、あるいは PICT フォーマットを含む場合 (ピクチャー内には複数のフォーマットが含まれる場合があります) に、古いフォーマットと判定されます。

*ptrPict* には、4Dピクチャーフィールドや変数、あるいはピクチャー配列の要素へのポインターを渡します。

このコマンドは、4Dの次のメジャーバージョンの準備をするために用意されました。Macの"PICT"フォーマットはAppleにより、すでに廃止予定としてアナウンスされています ([WikipediaのPICTフォーマットの記事参照](#))。互換性の目的で4DはPICTフォーマットをサポートしてきましたが、4D v14以降はサポートされません。アプリケーション移行を開始するために、このコマンドを使用してPICTピクチャーを探し、**CONVERT PICTURE**コマンドを使用して現在利用可能なピクチャーに変換することができます。変換後ピクチャーには指定したフォーマットのみが含まれます。

**注:** ピクチャーが印刷に使用されていた場合、PICTフォーマットを削除する前に事前の検証が必要です。PICTはベクターフォーマットであったため、印刷目的の場合にはEMFなどの他のベクターフォーマットに変換することが推奨されます。

### 例題

ライブラリ中のピクチャーをテストし、必要であれば変換を行う:

```
C_LONGINT ($v1Pict; $vPictCount; $vPictRef; $vTotal)
C_TEXT ($vPictName)
C_POINTER ($vPointer)
C_PICTURE ($vpPict)
ARRAY TEXT ($arrPictNames; 0)
ARRAY LONGINT ($arrPictRefs; 0)

$vTotal := 0
PICTURE LIBRARY LIST ($arrPictRefs; $arrPictNames)
$vPictCount := Size of array ($arrPictRefs)
If ($vPictCount > 0)
  For ($v1Pict; 1; $vPictCount) // 各ピクチャーごとに
    $vPictRef := $arrPictRefs {$v1Pict}
    $vPictName := $arrPictNames {$v1Pict}
    GET PICTURE FROM LIBRARY ($arrPictRefs {$v1Pict}; $vpPict)
    $vPointer := -> $vpPict // ポインターを渡す
    $isObs := AP Is Picture Deprecated ($vPointer)
    If ($isObs = 1) // フォーマットが廃止されていれば
      CONVERT PICTURE ($vPointer -> ".PNG") // png へと変換
    // そしてライブラリに保存
    SET PICTURE TO LIBRARY ($vPointer -> $vPictRef; $vPictName)
    $vTotal := $vTotal + 1
  End if
End for
ALERT (String ($vTotal) + " picture(s) out of " + String ($vPictCount) + " were converted to png.")
Else
  ALERT ("The picture library is empty.")
End if
```

## ⚙️ \_o\_AP Save BMP 8 bits

\_o\_AP Save BMP 8 bits ( image ; file ) -> 戻り値

引数	型		説明
image	ピクチャー	→	4Dピクチャ変数名
file	テキスト	→	BMPを保存するファイル名
戻り値	倍長整数	↩	処理に成功したら0; そうでなければエラーコード

### 説明

---

#### 互換性メモ:

- このコマンドは廃止予定です。このコマンドは互換性のためだけに残されていて、将来のバージョンのプラグインでは削除されます。4Dの"ピクチャ"テーマのコマンドの利用を強くお勧めします。
- このコマンドはOS X用の4D Packの64-bit版ではサポートされていません。

*AP Save BMP 8 bits* コマンドは、4Dピクチャ変数のピクチャを8-bit BMP ピクチャ (256色圧縮なし) で保存します。ピクチャ保存場所に同じ名前のファイルが存在する場合、警告なしで置き換えられます。このコマンドはMacintoshとWindows両プラットフォームで動作します。

処理に成功すると、関数は0を返します。そうでなければ負数のエラーコードを返します。

# 4D Pack : ユーザインタフェース

 AP FULL SCREEN

 AP NORMAL SCREEN

## AP FULL SCREEN

### AP FULL SCREEN

このコマンドは引数を必要としません

### 説明

---

*AP FULL SCREEN* コマンドは、4Dアプリケーションウィンドウのメニューバーやタイトルバーを隠し、このウィンドウをフルスクリーンサイズにすることを可能にします。

このコマンドはWindowsでのみ動作します。

## AP NORMAL SCREEN

AP NORMAL SCREEN

このコマンドは引数を必要としません

### 説明

---

*AP NORMAL SCREEN* コマンドは、*AP FULL SCREEN* コマンドで隠されたメニューバーやタイトルバーを再びサイズ変更できるようにします。

このコマンドはWindowsでのみ動作します。

## 4D Pack : ユーティリティ

-  AP Get field infos
-  AP Get table info
-  AP Get tips state
-  AP SET TIPS STATE
-  *\_o\_AP Get file MD5 digest*
-  *\_o\_AP ShellExecute*

## ⚙️ AP Get field infos

AP Get field infos ( numTable ; numField ; relTable ; relFld ; props ; choiceList ) -> 戻り値

引数	型		説明
numTable	整数	→	テーブル番号
numField	整数	→	フィールド番号
relTable	整数	←	リレートしたテーブル
relFld	整数	←	リレートしたフィールド
props	倍長整数	←	フィールドプロパティ
choiceList	文字	←	選択リスト
戻り値	整数	↻	エラーコード

### 説明

**AP Get field info**はフィールドに関する特定の情報を返します。

テーブル番号とフィールド番号を指定することで、このコマンドはリンクしたテーブルとリンクしたフィールドを各々返します。

指定されたフィールドにリレーションがない場合、*relTable*と*relFld*には0が返されます。

*props*には16ビットの値が返され、4Dのビット演算子を使用して値を読むことができます。

ビット番号	1に設定されていれば	無視するフィールドタイプ
15	インデックス	テキスト, ピクチャ, サブテーブル, Blob
14	インデックス 重複不可	テキスト, ピクチャ, サブテーブル, Blob, または15ビットが0
13	必須入力	サブテーブル
12	選択リスト	ピクチャ, サブテーブル, Blob
11	更新可	サブテーブル
10	入力可	サブテーブル
8	非表示	
6	N対1リレーション	テキスト, ピクチャ, サブテーブル, Blob
5	1対Nリレーション	テキスト, ピクチャ, サブテーブル, Blob
2と1	自動1対Nリレーションただし削除管理が: 00 削除管理をしない 01 リレートしたレコードを残して削除 10 リレートしたレコードを削除 11 リレートしたレコードがなければ削除	
0	自動N対1リレーション	テキスト, ピクチャ, サブテーブル, Blob, または5ビットが自動代入に設定されていない

**Note:** 3, 4, 7 そして 9 番目のビットは将来のため予約されています。

12番目のビットが設定されていれば、変数*choiceList*にリストの名前が格納されます。

## ⚙️ AP Get table info

AP Get table info ( numTable ; invisible ; destruct ) -> 戻り値

引数	型		説明
numTable	整数	→	テーブル番号
invisible	整数	←	1: テーブルは非表示 0: テーブルは非表示でない
destruct	整数	←	1: レコードを物理的に削除 0: レコードを論理的に削除
戻り値	整数	↩	エラーコード

### 説明

---

AP Get table infoは4Dテーブルの情報を返します。

このコマンドはテーブル番号を受け取り、デザインモードでテーブルが非表示に設定されている場合、*invisible* 引数に1を、そうでなければ0を返します。

*destruct*変数には、レコードを物理的に削除するに設定されている場合1が返され、論理削除の場合は0が返されます。レコードの物理削除は論理削除より遅いですが、"タグによる修復"を行った際のレコードの再出現を防ぎます。

## AP Get tips state

AP Get tips state -> 戻り値

引数	型		説明
戻り値	倍長整数		Tip表示ステータス

### 説明

---

*AP Get tips state*は、Tipsの表示ステータスが表示の時1を返します。そうでなければ0を返します。

## AP SET TIPS STATE

AP SET TIPS STATE ( status )

引数	型	説明
status	整数 →	Tipsのステータス 0 = Tipsを表示しない 1 = Tipsを表示する

### 説明

---

AP SET TIPS STATEコマンドを使用してTipsの表示・非表示を設定できます。

## 🔧 **\_o\_AP Get file MD5 digest**

`_o_AP Get file MD5 digest ( filePath ; digest ; fork ) -> 戻り値`

引数	型		説明
filePath	テキスト	→	ファイルの完全パス名
digest	テキスト	←	ファイルのMD5 Digest
fork	倍長整数	→	0=データフォーク, 1=リソースフォーク
戻り値	倍長整数	↪	エラーコード

### 説明

---

**互換性に関する注意:** 4D v13より**\_o\_AP Get file MD5 digest**は廃止予定であり、使用することはできません。このコマンドは4Dの**Generate digest**コマンドで置き換えてください。**\_o\_AP Get file MD5 digest**は互換性のためにのみ保持されています。

`_o_AP ShellExecute ( fileName ; mode ; param ) -> 戻り値`

引数	型	説明
fileName	文字	→ 起動するファイルやドキュメントの名称
mode	倍長整数	→ 起動時のウィンドウのステータス
param	テキスト	→ アプリケーションパラメタ
戻り値	倍長整数	🔄 0: コマンドの実行に成功した; 0以外の場合、システムエラー

### 説明

**互換性メモ:** このコマンドは廃止予定です。これは互換性のために残されていて、将来のバージョンのプラグインでは削除されます。4Dの **LAUNCH EXTERNAL PROCESS** コマンドの利用を強く推奨します。4D v11.0より、**\_o\_AP ShellExecute** はWindowsでのみ動作します。

**Note:** このコマンドはWindowsでのみ動作します。Mac OS上では効果がありません。

**\_o\_AP ShellExecute** コマンドを使用して、4Dデータベースからアプリケーションを起動したり、ドキュメントを開いたりできます。

*fileName* 引数には、起動するファイルの名前または完全なアクセスパスを渡します。開くアプリケーションやドキュメントが、ストラクチャファイルと同階層にある場合のみ、ファイ名や相対アクセスパスを渡せます。

ドキュメント名を渡すと、Windowsが関連付けられたアプリケーションの実行を管理します。

*mode* 引数を使用して、アプリケーションが起動されるウィンドウの表示モードを指定できます:

- 0を渡すと、ウィンドウは普通です;
- 1を渡すと、ウィンドウはフルスクリーンです;
- 2を渡すと、ウィンドウは最小化され、Windowsのタスクバーに表示されます。

*param* 引数を使用して、起動されるアプリケーションに必要な追加の引数を渡すことができます。例えば、このコマンドを使用してWebブラウザを起動する場合、この引数に特定のURLを渡せます。

### 例題 1

この例題は、メインディスクのルートにある、Word® ドキュメントを開きます:

```
\On Windows
$err:=$_o_AP ShellExecute("C:\Test.doc")
```

### 例題 2

この例題は、デフォルトブラウザをフルスクリーンモードで開き、4Dサイトに接続します:

```
$err:=$_o_AP ShellExecute("www.4d.com";"1")
```

### 例題 3

この例題はデフォルトのインターネットブラウザが何であれ、Netscape® をフルスクリーンで開き、4Dサイトに接続します:

```
$err:=$_o_AP ShellExecute("C:\Program Files\Netscape\NETSCAPE.EXE";"1";"www.4d.com")
```

### 例題 4

この例題はWindows Explorerを通常のウィンドウで開き、Musicフォルダの中身を表示します:

```
$err:==_o_AP ShellExecute("C:\WINDOWS\explorer.exe";0;"C:\Music")
```

## 4D Pack : 印刷

 AP BLOB to print settings

 AP Print settings to BLOB

## 🔧 AP BLOB to print settings

AP BLOB to print settings ( printSettings ; paramType ) -> 戻り値

引数	型	説明
printSettings	BLOB	→ 印刷設定を格納したBLOB
paramType	倍長整数	→ 0 = レイアウトと印刷, 1 = 印刷
戻り値	倍長整数	↻ 1 = 処理に成功した, 0 = カレントのプリンタがない, -1 = 引数が正しくない

### 説明

AP BLOB to print settings コマンドは、4Dの現在のプリンタ設定を *printSettings* BLOB に格納された内容で置き換えます。このBLOBは *AP Print settings to BLOB* コマンドで生成されていなければなりません。

*printSettings* 引数は2つのタイプのパラメタを含んでいます:

- レイアウトパラメタ: 用紙、方向、倍率
- 印刷パラメタ: 枚数、用紙トレイなど

印刷パラメタのみをロードするよう指定できます。

*paramType* に0を渡すかこの引数を省略すると、すべてのパラメタが使用されます。1を渡すと、印刷パラメタのみが使用されます。レイアウトパラメタは変更されません。

新しい印刷パラメタは、**PAGE SETUP**、**SET PRINT OPTION**、または>引数なしの**PRINT SELECTION**などのコマンドが設定を変更するまで、データベース全体およびすべてのセッションに適用されます。

設定されたパラメタは特に**PRINT SELECTION**、**PRINT LABEL**、**PRINT RECORD**、**Print form**、そして**QR REPORT**コマンドや、デザインモードを含む印刷メニューで使用されます。

AP BLOB to print settings で定義した設定を保持するためには、**PRINT SELECTION**、**PRINT LABEL**、そして**PRINT RECORD**コマンドを> 引数 (可能であれば) 付きで呼び出さなければなりません。

**Note:** 印刷設定はWindowsとMac OSで異なるフォーマットが使用されています。その結果、2つのプラットフォーム間での *printSettings* BLOB の互換性は保証されていません。

BLOBが正しくロードされると、コマンドは1を返します。カレントプリンタが選択されていない場合は0が、BLOBが正しくなければ-1が返されます。

## ⚙️ AP Print settings to BLOB

AP Print settings to BLOB ( printSettings ) -> 戻り値

引数	型	説明
printSettings	BLOB	→ 印刷設定を格納するBLOB
戻り値	倍長整数	↻ エラーコード: 1 = 処理に成功した 0 = カレントのプリンタがない

### 説明

---

*AP Print settings to BLOB* コマンドは4Dのカレントのプリンタ設定を*printSettings* BLOBに格納します。

*printSettings* 引数は、印刷で使用されるすべての設定を格納します:

- レイアウトパラメタ: 用紙、方向、倍率
- 印刷パラメタ: 枚数、用紙トレイなど

生成されたBLOBをプログラムで変更してはいけません。これは*AP BLOB to print settings* コマンドでのみ使用できます。

コマンドは、BLOBが正しく生成されると1を返します。カレントのプリンタが選択されていないと0を返します。

## 4D Pack - コマンドリスト (文字順)

B F G N P S \_

⚙️ AP BLOB to print settings