





4D Pack


 はじめに

 4D Pack : ANSIストリーム


 4D Pack : Windowsヘルプファイル

 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

 4D Pack : 外部時計

 新着

 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* マニュアルを参照してください。

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

4D製品の改良に関連し、以前のバージョンに存在していたいくつかのコマンドが4D Pack v11では取り除かれています。以下は削除されたコマンドと、推奨される代替りのソリューションです：

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 -> 現在代替りのソリューションはありません。

4D Pack : ANSIストリーム

 *AP FCLOSE*

 *AP fopen*

 *AP FPRINT*

 *AP fread*

AP FCLOSE

AP FCLOSE (fopenValue)

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

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、将来のバージョンでは削除されます。4Dの"通信"テーマのコマンドの利用を強くお勧めします。

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

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 사용되는値, または ファイルやポートが開けなかった場合0

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、将来のバージョンでは削除されます。4Dの"通信"テーマのコマンドの利用を強くお勧めします。

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、*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、*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:=AP fopen("LPT1";0)
AP FPRINT(port;"hello world")
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:=AP fopen("COM1";0)
AP FPRINT(port;"hello world")
AP FCLOSE(port)
```

例題 3

ファイルに "hello world" を書き込みます:

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

例題 4

ファイルを読み込みます:

```
port:=AP fopen("C:\myfile.txt";1)
```

```
MyVar:=AP fread(port)
```

```
AP FCLOSE(port)
```

AP FPRINT (fopenValue ; data)

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

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、将来のバージョンでは削除されます。4Dの"通信"テーマのコマンドの利用を強くお勧めします。

AP FPRINT コマンドは 指定されたdataテキストを、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**Mac to Win**コマンドを使用できます。

⚙️ AP fread

AP fread (fopenValue) -> 戻り値





引数	型	説明
fopenValue	倍長整数	→ AP fopen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

AP freadは、*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**Win to Mac**コマンドを使用できます。

4D Pack : Windowsヘルプファイル

-  AP CLOSE HELP
-  AP HELP INDEX
-  AP HELP ON HELP
-  AP HELP ON KEY

AP CLOSE HELP

AP CLOSE HELP

このコマンドは引数を必要としません

説明

互換性メモ: このコマンドはWindows Vistaと互換がありません。このコマンドの利用は推奨されません。
コマンドはWindows Helpアプリケーションを閉じます (Windowsのみ)。

AP HELP INDEX (help)

引数	型		説明
help	文字	→	ヘルプファイルの名前

説明

互換性メモ: このコマンドはWindows Vistaと互換がありません。このコマンドの利用は推奨されません。
コマンドは、引数で渡した名前のヘルプファイルで利用可能なキーワードのリストを表示します (Windowsのみ)。

AP HELP ON HELP

このコマンドは引数を必要としません

説明

互換性メモ: このコマンドはWindows Vistaと互換がありません。このコマンドの利用は推奨されません。
コマンドは、プログラム独自のヘルプファイルのオンラインヘルプを開きます (Windowsのみ)。

AP HELP ON KEY (help ; keyword)

引数	型		説明
help	文字	→	ヘルプファイルの名前
keyword	文字	→	ヘルプを検索するために使用するキーワード

説明

互換性メモ: このコマンドはWindows Vistaと互換がありません。このコマンドの利用は推奨されません。

コマンドは引数で渡した名前のヘルプファイルを開きます。このコマンドは標準のWindows Helpユーティリティを使用して、キーワードに対応するページを開きます (Windowsのみ)。

4D Pack : ピクチャファイル

 AP Is Picture Deprecated New 13.2

 *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)。

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フォーマットのもの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:

```
PICTURE LIBRARY LIST($arrPictRef;$arrPictName)
$vPictNums:=Size of array($arrPictRef)
If($vPictNums>0)
  For($vPict;1;$vPictNums) // 各ピクチャーごとに
    $vPictRef:=$arrPictRef{$vPict}
    $vPictName:=$arrPictName{$vPict}
    GET PICTURE FROM LIBRARY($arrPictRef{$vPict};$pvPict)
    $vPointer :=->$vPict
    $isObs:=AP Is Picture Deprecated($vPointer)
    If($isObs=1) // フォーマットが無効なら
      CONVERT PICTURE($vPointer->".PNG") // pngに変換
      SET PICTURE TO LIBRARY($vPointer->,$vPictRef;$vPictName) // ライブラリに格納
    End if
    $vPict:=$vPict+1
  End for
Else
  ALERT("The picture library is empty.")
End if
```


⚙️ AP Save BMP 8 bits

AP Save BMP 8 bits (image ; file) -> 戻り値

引数	型		説明
image	ピクチャー	→	4Dピクチャー変数名
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、将来のバージョンでは削除されます。4Dの"ピクチャー"テーマのコマンドの利用を強くお勧めします。

コマンドは、4Dピクチャー変数のピクチャーを8-bit BMP ピクチャー (256色圧縮なし) で保存します。ピクチャー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 Rect dragger

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。

⚙️ AP Rect dragger

AP Rect dragger (left ; top ; right ; bottom ; x ; y) -> 戻り値
















引数	型		説明
left	倍長整数	→	四角の左座標
top	倍長整数	→	四角の上座標
right	倍長整数	→	四角の右座標
bottom	倍長整数	→	四角の下座標
x	倍長整数	←	マウスの水平位置
y	倍長整数	←	マウスの垂直位置
戻り値	整数	↻	アイテムがドラッグされたプロセスの プロセスID

説明

AP Rect dragger を使用すると、4Dでドラッグ&ドロップインタフェースの実装ができます。このコマンドは、ある場所から他の場所にドラッグされたアイテムのグラフィックな表示を提供し、マウスがリリースされた座標と、アイテムがドラッグされたプロセスのプロセスIDを返します。これにより例えば**CALL PROCESS**でドロップ先プロセスを呼び出して、プロセス間のドラッグ&ドロップ処理を行うことができます。

このコマンドは非表示ボタンから呼び出さなければなりません。

4D Pack : ユーティリティ

-  AP Create method
-  AP Does method exist
-  AP Get field infos
-  AP Get table info
-  AP Get templates
-  AP Get tips state
-  AP Modify method
-  AP SET TIPS STATE
-  AP Timestamp to GMT
-  *AP AVAILABLE MEMORY*
-  *AP Get file MD5 digest*
-  *AP GET PARAM*
-  *AP Get picture type*
-  *AP SET PARAM*
-  *AP ShellExecute*

🔧 AP Create method

AP Create method (methodName ; propertiesArray ; methodCode ; folderName) -> 戻り値

引数	型	説明
methodName	文字	➡ メソッド名 (31文字)
propertiesArray	倍長整数配列	➡ プロパティ配列 (7要素)
methodCode	BLOB	➡ メソッドテキストを含むBLOB
folderName	文字	➡ メソッドが作成されるエクスプローラのフォルダ
戻り値	倍長整数	➡ エラーコード (0=エラーなし)

説明

AP Create method コマンドは、4Dデータベースストラクチャにプロジェクトメソッドを追加するために使用します (インタプリタデータベースのみ)。

methodNameにはメソッド名を渡します。この名前は31文字までを指定できます。4Dオブジェクトの命名規則に従う限り、文字、数字、スペース、下線などを含めることができます。名前が31文字を超えると、余った分は切り捨てられます。

propertiesArray引数はメソッドプロパティを指定するために使用します。配列は7つの要素を持つ倍長整数配列として事前に定義しなくてはなりません:

- propertiesArray{1}: メソッドが表示の時1、非表示なら0。
- propertiesArray{2}: メソッドが4DACTION、4DMETHOD、4DSCRIPTから利用可能であれば1、そうでなければ0。
- propertiesArray{3}: メソッドをWebサービスとして提供する場合1、そうでなければ0。
- propertiesArray{4}: propertiesArray{3}=1のとき、メソッドをWSDLで公開する場合1、そうでなければ0。
- propertiesArray{5}: メソッドをコンポーネントとホストデータベースで共有する場合1、そうでなければ0。
- propertiesArray{6}: メソッドがSQLから利用可能であれば1、そうでなければ0。
- propertiesArray{7}: メソッドに"サーバ上で実行"属性を設定する場合は1、そうでなければ0。

methodCodeには、メソッドのテキストを含むBLOBを渡します。TEXT TO BLOB コマンドを使用する際はMac C string (または0) をこのコマンドの3番目の引数に渡します。

methodCodeに、メソッド書き出し...メニューコマンドでファイルに書き出されたメソッドのテキストを渡すことができます (この場合、テキストにはコマンドや定数などが参照 (トークン) として含まれています)。

folderNameには、メソッド作成先のエクスプローラのフォルダ名を渡します。このフォルダはエクスプローラのホームページで管理され、オブジェクトをカスタマイズされた方法で管理できます。この引数はオプションです。省略された場合、メソッドは"デフォルトプロジェクトメソッド"フォルダに作成されます。

処理が成功すると0が返されます。そうでなければエラーコードが返されます:

- 1 = 引数の型が正しくありません。
- 2 = データベースはコンパイルモードで実行中です。
- 3 = メソッド名が空です。
- 4013 = メソッド名が正しくありません。
- 4014 = メソッドを作成できません。

Note: このコマンドはコンパイルモードで実行中のアプリケーションでは動作しません。

⚙️ AP Does method exist

AP Does method exist (methodName) -> 戻り値

引数	型		説明
methodName	文字	➡	テストするメソッド名
戻り値	整数	↻	0 = メソッドは存在しない, 1 = メソッドは既に存在する

説明

AP Does method exist コマンドを使用して、カレントデータベースに *methodName* という名前のプロジェクトメソッドが存在するかどうかを知ることができます。このコマンドは、データベースにインストールされたコンポーネントのプロジェクトメソッドを考慮に入れません。

AP Create method コマンドを使用する際、このコマンドは、同じ名前のメソッドが既に存在する際のエラーメッセージを避ける目的で使用されます。

⚙️ 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 templates

AP Get templates (templateNameArray)

引数	型	説明
templateNameArray	文字配列	← テンプレート名の配列

説明

AP Get templates コマンドは、カレントの4Dアプリケーションで利用可能なフォームテンプレートのリスト配列 *templateNameArray* を作成します。

フォームテンプレートはフォームウィザードダイアログで選択できます。組み込みのテンプレートの例は、ノスタルジア、OSX、OSX (ラベルなし)、フィラデルフィアなどです。フォームウィザードの詳細ページでカスタマイズされたフォームをテンプレートとして追加できます。

Note: このコマンドは、4D Volume Licenseがマージされたコンパイル済みアプリケーションで実行された時は、動作しません。

コマンドが正しく実行されると、0が返されます。そうでなければエラーコードが返されます。

例題

この例題は、利用可能なテンプレートを表示するウィンドウを作成します:

```
$ref:=Open window (<>WLeft;<>WTop;450;400;8;"Template List")
ARRAY STRING (255;ArrTemplates;0)
$error:=AP Get templates(ArrTemplates)
DIALOG ([Table 1];"ArrTemplates")
CLOSE WINDOW ($ref)
```

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 Modify method

AP Modify method (methodName ; methodCode) -> 戻り値

引数	型		説明
methodName	文字	→	変更するメソッドの名前
methodCode	BLOB	→	メソッドテキストを含むBLOB
戻り値	倍長整数	↻	エラーコード (0=エラーなし)

説明

AP Modify method コマンドを使用して、インタプリタモードの4Dデータベースにおいて、ストラクチャのプロジェクトメソッドのソースコードを変更できます。

methodName 引数に既存のプロジェクトメソッドの名前を渡します (4Dメソッドの名前は31文字までです)。

*methodCode*には、メソッドの新しい内容を含んだBLOBを渡します。**TEXT TO BLOB** コマンドを使用してBLOBを作成する場合、Mac C Stringフォーマット (または0) を3番目の引数に渡します。

メソッドが正しく更新されると、コマンドは0を返します。そうでなければ以下のいずれかのエラーコードを返します:

- 1 = 引数の型が正しくありません。
- 2 = データベースはコンパイルモードで実行中です。
- 3 = メソッド名が空です。
- 4013 = メソッド名が正しくありません。
- 4015 = そのメソッドソースコードは更新できません。

コンパイル済みアプリケーションから実行された場合、このコマンドは動作しません。

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。

⚙️ AP Timestamp to GMT

AP Timestamp to GMT (localDate ; localTime ; gmtDate ; gmtTime) -> 戻り値

引数	型		説明
localDate	日付	→	変換に使用するローカルの日付
localTime	時間	→	変換に使用するローカルの時刻
gmtDate	日付	←	GMTに変換された日付
gmtTime	時間	←	GMTに変換された時刻
戻り値	文字	↻	RFC標準に基づくGMT時刻

説明

互換性に関する注意: 4Dバージョン12.1より、*AP Timestamp to GMT* コマンドは4Dの**String**コマンドにDate RFC 1123定数を指定することで、置き換えられます。**String**コマンドを使用すれば日付と時刻を主要な標準フォーマットに変換できます。

*AP Timestamp to GMT*はローカルの日付と時刻を、マシンの日付と時刻コントロールパネルの設定に基づき、GMTの日付と時刻に変換します。

このコマンドは以下の形式で文字列を返します: "Weekday, DD MM YYYY HH:MM:SS GMT".このフォーマットはRFC 822標準で定義されています。

例題

日付に1997/11/13 (木)、時刻に2:35PM (Parisタイムゾーン) を渡すと、関数は文字列 "Thu, 13 Nov 1997 13:35:00 GMT"を返します。*\$gmtd*は1997/11/13に、*\$gmtt*は13:35:00が返されます。

```
$s:=AP Timestamp to GMT(Current date;Current time;$gmtd;$gmtt)
```

AP AVAILABLE MEMORY

AP AVAILABLE MEMORY (totalMemory ; physicalMemory ; freeMemory ; freeStack)

引数	型		説明
totalMemory	倍長整数	←	総メモリ
physicalMemory	倍長整数	←	物理メモリ
freeMemory	倍長整数	←	空きメモリ
freeStack	倍長整数	←	空きスタック

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、将来のバージョンでは削除されます。

AP AVAILABLE MEMORY コマンドは、このコマンドが実行されているマシンにインストールされたメモリ、空きメモリをキロバイト単位で返します。

このコマンドはマシンにインストールされた総メモリと物理メモリ、4Dが利用可能な空きメモリ、カレントプロセスの空きスタックを返します。

AP Get file MD5 digest

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**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。**AP Get file MD5 digest**は互換性のためにのみ保持されています。

AP GET PARAM

AP GET PARAM (option ; value)

引数	型		説明
option	整数	→	読みだすオプション番号
value	整数	←	オプションの値

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、将来のバージョンでは削除されます。[Get database parameter](#)コマンドの利用を強くお勧めします。

*AP GET PARAM*を使用して、通常4D Customizer Plusで設定される4Dの内部的なオプションを読みだせます。このコマンドを使用して、4D Customizer Plusで設定した値を取り出し、*AP SET PARAM*で一時的に変更した値を元に戻すことができます。

value 引数の詳細は、[AP SET PARAM](#)コマンドの説明を参照してください。

例題

以下のコードは引数 2の値を返します:

```
AP GET PARAM(2;$param)
```

⚙️ AP Get picture type

AP Get picture type (storedPicture) -> 戻り値

引数	型		説明
storedPicture	BLOB	→	BLOBとして格納されたピクチャ
戻り値	倍長整数	↩	ピクチャのタイプ

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。

はBLOB *storedPicture*に格納されている画像のタイプを返します:

- 1 未知のタイプ
- 0 'PICT' ピクチャ
- 1 QuickTime 圧縮 'PICT' ピクチャ
- 2 JPEG ピクチャ
- 3 WMF ピクチャ
- 4 EMF ピクチャ
- 5 BMP ピクチャ
- 6 GIF ピクチャ

例題

この例題はピクチャをBLOBに格納し、そのタイプを\$typeに格納します。

```
DOCUMENT TO BLOB($thefile;$theblob)
$type:=AP Get picture type($theblob)
```

AP SET PARAM (option ; value)

引数	型	説明
option	整数 →	設定するオプションの番号
value	整数 →	オプションの新しい値

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、将来のバージョンでは削除されます。**SET DATABASE PARAMETER**コマンドの利用を強くお勧めします。

AP SET PARAM コマンドは、通常4D Customizer Plusで設定する内部的な4Dのオプションを設定可能にします。これらのオプションはCUST ID=0のリソースに格納されます。

整数値は以下のとおりです:

- 0 - valueが0のとき、4Dの印刷進捗ウィンドウが表示されている間、デスクアクセサリを開くことはできません。valueが1のとき、そのウィンドウが表示されている間でもデスクアクセサリを開くことができます。デフォルト値は0です。
- 1 - valueが0のとき、検索や並び替えのシーケンシャル処理進捗ウィンドウが表示されている間、デスクアクセサリを開くことはできません。valueが1のとき、そのウィンドウが表示されている間でもデスクアクセサリを開くことができます。Windowsバージョンでは、valueに-3を渡すと、4Dはスクリーンのグラフィックを更新するための、オフスクリーンのビットマップを使用しません。すべての描画処理は直接スクリーンに対して行われます。valueが-4のとき、4Dは起動時に、スクリーンのサイズや深度に対応する永久的なオフスクリーンバッファを割り当てます。そしてビットマップベースのグラフィックの更新にはこのバッファを使用します。デフォルト値は0です。
これら以外の値の場合、4Dはスクリーンの更新のたびにに対応するバッファを割り当てようとしていきます (十分なメモリがある場合)。この処理に成功すると、4Dはオフスクリーンのビットマップベース更新を行い、バッファを破棄します。オフスクリーンのビットマップベース更新を有効にすると、たくさんのオブジェクトを再描画しなければならないような場合のスクリーンのちらつきを抑えることができます。
永久的なオフスクリーンバッファはスペースを必要とします (例えば640 x 480、256色のスクリーンで300K)。なので、メモリに余裕がない場合、valueを-3や-4以外の値に設定したいと考えるでしょう。メモリに問題がない場合、このオプションを-4に設定すると、描画処理を高速化できます。オフスクリーンのビットマップベース更新を無効にすると (-3を設定)、Windowsで画面が更新されない4Dプラグインの非互換の問題が解決されることがあります。
- 2 - 4Dがフォーミュラを実行する際の回転カーソルの回転速度を設定します。valueが0のとき、カーソルは表示されません。正または負の数を指定すると、回転速度を32のモジュールに設定します。平均の値は16です。デフォルト値は0です。
- 3 - データ入力中のアクティブな組み込みフォームで、新規にサブレコードやレコードを作成するためのショートカットをASCIIコードで指定します。デフォルト値は47 ([/] キー)です。
- 4 - option 3に関連づけられるモディファイアキーの値を含みます。デフォルト値は256 (Command キー)です。指定可能な値は256-コマンド (Ctrl) キー、512-Shiftキー、1024-Caps Lockキー、2048- Option (Alt) キーです。
- 5 - データ入力を受け付けるショートカットのASCIIコードを指定します。デフォルト値は3 (Enter キー)です。
- 6 - option 5に関連付けられるモディファイアキーを指定します。デフォルト値は0 (モディファイアキーなし)です。
- 7 - データ入力をキャンセルするショートカットのASCIIコードを指定します。デフォルト値は46 ([.]キー)です。
- 8 - option 7に関連付けられるモディファイアキーを指定します。デフォルト値は256 (Command キー)です。
- 9 - OKショートカットのASCIIコードを指定します。デフォルト値は3 (Enter キー)です。
- 10 - option 9に関連付けられるモディファイアキーを指定します。デフォルト値は0 (モディファイアキーなし)です。
- 11 - ダイアログボックスをキャンセルするショートカットのASCIIコードを指定します。デフォルト値は46 ([.]キー)です。
- 12 - option 11に関連付けられるモディファイアキーを指定します。デフォルト値は256 (Command キー)です。
- 13 - 4Dメインウィンドウのデフォルトの位置を指定します。valueが0のとき、ウィンドウはスクリーンサイズになり、タイトルバーはメニューバーの下に表示されます。valueが1のとき、ウィンドウはスクリーンサイズになり、タイトルバーは隠されます。valueが2のとき、ウィンドウのサイズはoption 14, 15, 16, 17で指定した座標となります。valueが3のとき、ウィンドウのサイズはoption 14, 15, 16, 17で指定した座標となりますが、スクリーンの中央に配置されます。valueが4のとき、ウィンドウは前セッションと同じサイズと位置に表示されます。デフォルト値は4で

す。

次の4つのオプションは、デフォルト値が-1で、*option* 13が2または3の場合にのみ使用されます：

- 14 - 4Dウィンドウの上座標
- 15 - 4Dウィンドウの左座標
- 16 - 4Dウィンドウの下座標
- 17 - 4Dウィンドウの右座標

- 18 - ランタイム環境に移動した際にメインの4Dウィンドウをリサイズするかどうかを指定します。*value*が0のとき、ウィンドウはリサイズされません。1のとき、ウィンドウは*option* 13の設定に従いリサイズされます。デフォルト値は0です。
- 19 - カスタムメニューから4Dメソッドが呼び出された際に、メインの4Dウィンドウをリサイズするかどうかを指定します。*value*が0のとき、ウィンドウはリサイズされません。1のとき、ウィンドウは*option* 13の設定に従いリサイズされます。デフォルト値は0です。
- 20 - 予約済み。この*option*を変更しないでください。
- 21 - 予約済み。この*option*を変更しないでください。
- 22 - 予約済み。この*option*を変更しないでください。
- 23 - 予約済み。この*option*を変更しないでください。
- 24 - 予約済み。この*option*を変更しないでください。
- 25 - データ入力をキャンセルした際に、フォームメソッドのDuringフェーズを呼び出すかを指定します。*value*が0のとき、Duringフェーズは呼び出されません。そうでなければ呼び出されます。デフォルト値は0です。
- 26 - セレクションを表示あるいは印刷時に、自動でテーブルステータスの読み込みのみ/読み書きを切り替えるかを指定します。*value*が0のとき、自動スイッチが行われます。1の場合、自動スイッチは行われません。デフォルト値は0です。
- 27以降のオプションは予約済みです。これらの*option*を変更しないでください。

例題

コマンド実行時にカードルを回転車にしたい場合、以下のコードを実行します：

```
AP SET PARAM(2;16)
```

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、将来のバージョンで削除されます。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、*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。

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:

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

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:=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 *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ければ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 : 外部時計

 _AP External clock

 AP SET CLOCK

_AP External clock

_AP External clock

このコマンドは引数を必要としません

説明

外部エリアの *_AP External clock* は、4D Pack プラグインの PICT 17890 リソースを使用して、アナログ時計を表示します。このリソースは時計のダイヤルを描画します。

□

ストラクチャファイルに PICT 17890 リソースを置いて、カスタマイズした時計を描画できます。この PICT リソースは Macintosh で ResEdit を使用して作成できます。

ピクチャはどのようなサイズでも使用できます。時計のサイズは適当に調整されます。同様に、ダイヤルの形は正方形に収まっている必要はありません。

デフォルトで、時計はシステムの時間を表示します。しかし *AP SET CLOCK* コマンドを使用して、現在の時間と表示される時間の間の遅延を指定できます。つまり異なるタイムゾーンに対応する時計を表示できます。

AP SET CLOCK

AP SET CLOCK (clock ; hour ; minute)

引数	型		説明
clock	倍長整数	→	_AP External clock 外部エリア
hour	整数	→	時間の遅延
minute	整数	→	分の遅延

説明

AP SET CLOCK コマンドは現在の時間と *_AP External clock* 外部エリアに表示される時間の遅延を設定するために使用します。

4D Pack - 新着

13.2

13.0

 AP Is Picture Deprecated New 13.2

4D Pack - コマンドリスト (文字順)

A B C D F G H M N P R S T _

 AP AVAILABLE MEMORY