

4D_Pack

version 2004.1



4D_Pack

Version 2004.1 for Windows® and Mac™ OS

© 4D SA/4D, Inc. 1995-2005.
All rights reserved.

The Software described in this manual is governed by the grant of license in the 4D Product Line License Agreement provided with the Software in this package. The Software, this manual, and all documentation included with the Software are copyrighted and may not be reproduced in whole or in part except for in accordance with the 4D Product Line License Agreement.

4th Dimension, 4D, the 4D logo, 4D Server are registered trademarks of 4D, Inc.

Microsoft and Windows are registered trademarks of Microsoft Corporation.

Apple, Macintosh, Mac OS and QuickTime are trademarks or registered trademarks of Apple Computer, Inc.

Mac2Win Software Copyright © 1990-2005, is a product of Altura Software, Inc. This product includes software developed by the Apache Software Foundation (<http://www.apache.org/>).

4th Dimension includes cryptographic software written by Eric Young (eay@cryptsoft.com)

4th Dimension includes software written by Tim Hudson (tjh@cryptsoft.com).

ACROBAT © Copyright 1987-2005, Secret Commercial Adobe Systems Inc. All rights reserved. ACROBAT is a registered trademark of Adobe Systems Inc.

4D_Pack is based in part on the work of the Independent JPEG Group and the Quest Center at Cold Spring Harbor Labs.

All other referenced trade names are trademarks or registered trademarks of their respective holders.

Contents

1. Preface.....7

4D_Pack, Introduction.....9

2. 4D_Pack : User Interface..... 11

AP Rect dragger.....13
AP PICT DRAGGER.....14
%AP PICT displayer.....15
AP PICT UPDATER.....16
AP FULL SCREEN.....17
AP NORMAL SCREEN.....18

3. 4D_Pack : Windows Help Files..19

AP HELP ON KEY.....21
AP HELP INDEX.....22
AP HELP ON HELP.....23
AP CLOSE HELP.....24

4. 4D_Pack : Utilities..... 25

AP Sublaunch.....27
AP ShellExecute.....29
AP Toolbar installed.....31
AP Get table info.....32
AP Get field infos.....33
AP Add table and fields.....34
AP Get templates.....37
AP Create relation.....38
AP Create method.....39
AP SET WEB FILTERS.....40
AP Get tips state.....42
AP SET TIPS STATE.....43
AP Set palette.....44
AP SET PICT MODE.....45

AP Timestamp to GMT.....	46
AP Get picture type.....	47
AP AVAILABLE MEMORY.....	48
AP Select document.....	49
AP GET PARAM.....	51
AP SET PARAM.....	52

5. 4D_Pack : External Clock..... 55

_AP External clock.....	57
AP SET CLOCK.....	58

6. 4D_Pack : Printing..... 59

AP PrintRec to text.....	61
AP Text to PrintRec.....	62
AP Print settings to BLOB.....	63
AP BLOB to print settings.....	64
AP PrValidate.....	65
AP PrintDefault.....	66

7. 4D_Pack : ANSI Streams..... 67

AP fopen.....	69
AP FPRINT.....	71
AP FCLOSE.....	72
AP fread.....	73

8. 4D_Pack : Picture Files..... 75

AP Read Picture File.....	77
AP Save BMP 8 bits.....	79
AP Read Picture BLOB.....	80
AP Save GIF.....	81

Command Index.....83

1

Preface

The 4D_Pack plug-in provides 4D developers with useful additional routines.

The 4D_Pack command names have the “AP” prefix in order to avoid any confusion with 4D commands. They also use the uppercase/lowercase convention depending on whether they are commands or functions.

Installation

You install the 4D_Pack plug-in just as any 4th Dimension plug-in. For more information, refer to the 4th Dimension Product Line *Installation Guide*.

Package Contents

The 4D_Pack package contains several routines grouped into the following themes:

Themes	Description
User Interface	User interface management
Windows Help Files	Using the Help Manager (Windows only)
Utilities	Miscellaneous routines
External Clock	“Analog” clock displayed in a plug-in area
Picture Files	Picture management
Printing	Management of the printing parameters
ANSI streams	Management of sending data to files and ports

2

4D_Pack : User Interface

AP Rect dragger (left; top; right; bottom; x; y) → Integer

Parameter	Type		Description
left	Longint	→	Left coordinate of the visible rectangle
top	Longint	→	Top coordinate of the visible rectangle
right	Longint	→	Right coordinate of the visible rectangle
bottom	Longint	→	Bottom coordinate of the visible rectangle
x	Longint	←	Horizontal location of the mouse
y	Longint	←	Vertical location of the mouse
Function result	Integer	←	Process ID of the process in which the item was dragged

Description

AP Rect dragger allows you to implement a “Drag and Drop” interface within 4D. It displays a graphical representation of an item being dragged from one place to another and returns the coordinates of the mouse release and the process ID of the process in which the item was dragged.

See also

AP PICT DRAGGER, CALL PROCESS.

AP PICT DRAGGER (picture; x; y)

Parameter	Type		Description
picture	Picture	→	Picture to be dragged
x	Longint	→	Horizontal coordinate after the move
y	Longint	→	Vertical coordinate after the move

Description

The AP PICT DRAGGER command allows you to implement a drag and drop operation on an image in the active window.

The call is implemented from an invisible button. The x and y variables receive the new coordinates generated by the mouse release.

See Also

AP Rect dragger.

%AP PICT displayer is a plug-in area that allows you to display a picture set using the AP PICT UPDATER routine or automatically loaded from a PICT resource. To automatically display a picture coming from a resource, give your plug-in area a name with an alphabetic first character, followed by the resource ID number of the picture you want to see. For example, to display the PICT resource #128, you may name your area "v128".

PICT resources should belong to the database structure resource file.

See Also

AP PICT UPDATER.

AP PICT UPDATER (area; image)

Parameter	Type		Description
area	Longint	→	%AP PICT displayer plug-in area
image	Picture	→	Image to be updated

Description

The AP PICT UPDATER command enables you to set or change the picture displayed in a plug-in area managed by the %AP PICT displayer routine. Unlike regular 4D areas, which are redrawn according to form events, AP PICT UPDATER immediately redraws the area at the screen. You can thus use it within a loop to perform animation effects.

See Also

%AP PICT displayer.

AP FULL SCREEN

Parameter	Type	Description
		This command does not require any parameters

Description

The AP FULL SCREEN command enables you to hide the menu bar and the title bar of the 4D application window and to set this window to full-screen size.

This command only works in Windows.

See Also

AP NORMAL SCREEN.

AP NORMAL SCREEN

Parameter	Type	Description
		This command does not require any parameters

Description

The AP NORMAL SCREEN command enables you to resize a menu bar that was previously hidden by the AP FULL SCREEN command.

This command only works in Windows.

See Also

AP FULL SCREEN.

3

4D_Pack : Windows Help Files

AP HELP ON KEY (help; keyword)

Parameter	Type		Description
help	String	→	Name of the Help file
keyword	String	→	Keyword used to search for help

Description

The AP HELP ON KEY command opens a Help file whose name is passed as a parameter. This command uses the standard Windows Help utility, which displays a page corresponding to the keyword found.

AP HELP INDEX (help)

Parameter	Type		Description
help	String	→	Name of the Help file

Description

The AP HELP INDEX command displays the list of keywords available in the Help file whose name was passed as a parameter.

AP HELP ON HELP

Parameter	Type	Description
		This command does not require any parameters

Description

The AP HELP ON HELP command displays the on-line help for the program's own Help file.

AP CLOSE HELP

Parameter	Type	Description
		This command does not require any parameters

Description

The command AP CLOSE HELP closes the Windows Help application (on Windows only).

4

4D_Pack : Utilities

AP Sublaunch (filename; mode) → Longint

Parameter	Type		Description
filename	String	→	Name of the application or document to launch
mode	Number	→	Status of windows at time of launch (Windows) Synchronous mode (Macintosh)
Function result	Longint	←	0, if the command execution succeeded; otherwise a system error code

Description

AP Sublaunch enables you to launch an application or to open a document from within 4D.

On Windows

- Applying this command to an executable file is equivalent to running that file using the **Open** command from the **File** menu of the window. Therefore, you can pass switches as well as names of documents to be opened by the executable file.
- The mode parameter sets the appearance mode of the window in which the application will be launched:
 - if mode is set to 0, the application window is normally displayed.
 - if mode is set to 1, the application window is in full screen.
 - if mode is set to 2, the application window is reduced and appears in the task bar (Windows 95/98 and NT 4).

On Macintosh

- You can pass a filename or an application name in filename. Applying this command to an application or a document is equivalent to double-clicking the icon of the application or the document in the Finder.
- The mode parameter set to 0 specifies an asynchronous launch. mode set to 1 specifies a synchronous launch.

When the application is launched with or without documents, you can drive it using DDE Tools, OLE Tools (Windows) or Apple Events (Macintosh).

If the operation is successfully completed, the function returns 0; otherwise it returns a negative error code.

Warning: The behavior of this command depends on many parameters; its result is not guaranteed for all applications.

Examples

1. Windows

- ⇒ `$errCode:=AP Sublaunch("C:\WINDOWS\CALC.EXE";1)`
` Will start the Calculator
- ⇒ `$errCode:=AP Sublaunch("C:\WINDOWS\notepad.exe D:\DOCS\README.TXT";1)`
` Will start the Windows Note Pad, which will open the document README.TXT

2. Macintosh

- ⇒ `$errCode:=AP Sublaunch("MyDisk:MyFolder:MyApp";0)` ` Will start MyApp
- ⇒ `$errCode:=AP Sublaunch("MyDisk:MyFolder:MyDoc";0)`
` Will start the application to which MyDoc belongs, then open this later

See Also

AP ShellExecute.

AP ShellExecute (fileName{; mode{; param}}) → Longint

Parameter	Type		Description
fileName	String	→	Name of file or document to be launched
mode	Longint	→	Status of window at time of launch (Windows) Synchronous mode (MacOS)
param	Text	→	Application parameters (Windows)
Function result	Longint	←	0, if the command execution succeeded; otherwise, a system error

Description

The AP ShellExecute command enables you to launch an application or open a document from a 4D database.

Note : This command is similar to the existing AP Sublaunch command; however, its operation has been optimized under Windows, in particular via the use of the 3rd parameter.

Under Windows

In the fileName parameter, pass the name or complete (absolute) access path of the file to be launched. When the application or document to be opened is located next to the database structure, it is possible to pass only the file name or a relative access path. Simply indicate a document name and Windows handles the execution of the associated application.

The mode parameter enables you to specify the display mode of the window in which the application will be launched:

- if you pass 0, the window will be normal;
- if you pass 1, the window will be full-screen;
- if you pass 2, the window will be reduced and will appear in the Windows task bar.

The param parameter enables you to pass any additional type of parameter that may be necessary for the launched application. For instance, if you use this command to launch a Web browser, you can pass a specific URL in this parameter.

Note : The param parameter is only used under Windows. It is ignored under MacOS.

Under MacOS

In the fileName parameter, pass the name or complete (absolute) access path of the file to be launched. When the application or document to be opened is located next to the database structure, it is possible to pass only the file name or a relative access path.

Examples

(1) This example opens a Word® document located at the root of the main disk:

```
`Under Windows
$err:=AP ShellExecute ("C:\Test.doc")
`Under MacOS
$err:=AP ShellExecute ("HardDisk:Test.doc")
```

(2) This example launches an internet browser, in full-screen mode by default, on the 4D site:

```
$err:=AP ShellExecute ("www.4d.fr";"1")
```

(3) This example (Windows) opens the 4D site in full-screen on Netscape® whatever the default internet browser:

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

(4) This example opens the Windows Explorer in a normal window and displays the contents of the Music folder:

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

See also

AP Sublaunch.

AP Toolbar installed → Longint

Parameter	Type	Description
This command does not require any parameters		
Function result	Longint	← Status of the toolbar

Description

AP Toolbar installed returns 1 if the toolbar is installed; otherwise, it returns 0. You can use it to determine where to create a window on screen.

AP Get table info (table; invisible; destruct) → Integer

Parameter	Type		Description
table	Integer	→	Table number
invisible	Integer	←	1, if the table is invisible 0, if the table is visible
destruct	Integer	←	1, for physical deletion of the records 0, for logical deletion of the records
Function result	Integer	←	Error code

Description

AP Get table info returns information on 4D tables.

Given the table number, the invisible parameter is set to 1 if the table is declared as invisible in Design mode, otherwise it is set 0.

The destruct variable is set to 1 for physical deletion of the records, or 0 for logical deletion. The physical deletion of the records is slower than the logical one but it prevents the record from reappearing in case of a "recover by tags" with 4D Tools.

AP Get field infos (table; field; relTable; relFld; props; choiceList) → Integer

Parameter	Type		Description
table	Integer	→	Table number
field	Integer	→	Field number
relTable	Integer	←	Related table
relFld	Integer	←	Related field
props	Longint	←	Field properties
choiceList	String	←	Choice list
Function result	Integer	←	Error code

Description

AP Get field infos returns specific information about a field. Given the table number and field number, the command returns the linked table and the linked field number, respectively.

If there is no relation with the given field, relTable and relFld are set to 0.

The 16 bits of props are meaningful. Each bit can be read via bits operators of 4D language.

Bit number:	Is set to 1 if:	Ignored if field type is:
15	Indexed	Text, Picture, Subtable or Blob
14	Indexed unique	Text, Picture or Subtable or Blob or if bit 15 is set to 0
13	Mandatory	Subtable
12	Choice list	Picture, Subtable or Blob
11	Modifiable	Subtable
10	Enterable	Subtable
8	Invisible	
6	N to 1 relation	Text, Picture, Subtable or Blob
5	1 to N relation	Text, Picture, Subtable or Blob
2 and 1	Automatic 1 to N relation whose deletion control is:	
	00 No deletion control	
	01 No assignment of related records	
	10 Deletion of related records	
	11 Deletion of related records not allowed	
0	Automatic N to 1 relation with automatic assignment	Text, Picture, Subtable, Blob, or if bit 5 not set

Note: Bits 3, 4, 7 and 9 are reserved for future use.

If bit 12 is set, the variable choiceList contains the name of the list.

AP Add table and fields (tableName; fieldNamesArray; fieldTypesArray; fieldLengthsArray; listFormTemplate; detailFormTemplate)) → Longint

Parameter	Type		Description
tableName	String	→	Name of table to be created
fieldNamesArray	String Array	→	Array of field names
fieldTypesArray	Longint Array	→	Array of field types
fieldLengthsArray	Longint Array	→	Array of field lengths for Alpha fields (0 for non-Alpha fields)
listFormTemplate	String	→	Template to use for the default list form creation
detailFormTemplate	String	→	Template to use for the default detail form creation
Function result	Longint	←	Number of the table created

Description

The AP Add table and fields command lets you add a table and its fields in a 4th Dimension database structure.

Note: This command was created mainly for the needs of 4D ODBC Pro plug-in users.

Pass the name of the table in tableName. This name can be up to 31 characters long. It can contain any combination of letters, numbers, spaces and underlines.

The command creates as many fields in the table as there are elements in the fieldNamesArray, fieldTypesArray and fieldLengthsArray arrays. The information passed in each element is used to create the corresponding fields. The first element of the arrays will be used for the first field of the table, and so on.

Note: To create an empty table, pass empty arrays.

- Pass the names of all the fields to be created in the tableName table in fieldNamesArray. Each field name can be up to 31 characters long. It must begin with a letter but you can then use any type of letter or number, as well as spaces and/or underlines.

- Pass the types of each field in `fieldTypesArray`. You can use the following 4D constants, located in the “Field and Variable Types” theme:

Constant	Type	Value
Is Alpha Field	Longint	0
Is Text	Longint	2
Is Real	Longint	1
Is Integer	Longint	8
Is Longint	Longint	9
Is Date	Longint	4
Is Time	Longint	11
Is Boolean	Longint	6
Is Picture	Longint	3
Is BLOB	Longint	30

- For each Alpha field created, pass a length in `fieldLengthsArray`. For other types of fields, this value is not necessary and you can pass 0.

The fields are created with the default properties. The table is automatically placed in the “Top Level” group.

The table is created with its two default forms. These forms are designed using the current form template. You can define the templates to use for the creation of these default forms using the optional `listFormTemplate` and `detailFormTemplate` parameters. In `listFormTemplate`, pass the name of the template you want to be used for the default list (output) form. In `detailFormTemplate`, pass the name of the template you want to be used for the default detail (input) form.

The form templates can be selected in the Form Wizard dialog box. You can get the list of form templates available in your 4D application using the 4D Pack command `AP Get templates`.

If it has been successfully executed, the command returns the number of the table created. This number, which is automatically assigned by 4th Dimension, corresponds to the value returned by the expression: `Count tables + 1`.

If the table could not be created (for example, if you pass the name of an existing table), the command returns 0.

Examples

(1) This method lets you create an [Employees] table:

```

ARRAY STRING(31;tFieldNames;5)
ARRAY LONGINT(tFieldTypes;5)
ARRAY LONGINT(tFieldLengths;5)
C_LONGINT($ret)

```

```

Initialization of the arrays
tFieldNames{1}:="Name"
tFieldTypes{1}:=Is Text
tFieldLengths{1}:=0

```

(2) This method will create a table with two fields; default forms will use the 2nd and 3rd templates:

```
C_STRING(25;TableName)
C_STRING(255;$FormTempList;$FormTempDetail)
ARRAY STRING(32;FieldsNameArray;2)
ARRAY LONGINT(FieldsTypeArray;FieldsLengthArray;2)
ARRAY STRING(255;$ArrTemplates;0)
TableName:="NewTable"
FieldsNameArray{1}:="FirstField"
FieldsTypeArray{1}:="Is Alpha Field"
FieldsNameArray{2}:="SecondField"
FieldsTypeArray{2}:="Is LongInt"

$error:=AP Get templates ($ArrTemplates)
If ($error=0)
    $FormTempList:=$ArrTemplates{3}
    $FormTempDetail:=$ArrTemplates{2}
    $NumTable:=AP Add table and fields(TableName;FieldsNameArray;FieldsTypeArray;
                                        FieldsLengthArray;$FormTempList;$FormTempDetail)
Else
    $NumTable:=AP Add table and fields(TableName;FieldsNameArray;FieldsTypeArray;
                                        FieldsLengthArray)
End if
```

AP Get templates (templateNameArray)

Parameter	Type	Description
templateNameArray	String Array ←	Array of template names

Description

The AP Get templates command populates the array templateNameArray with the list of the form templates available in the current 4th Dimension application.

The form templates can be selected in the Form Wizard dialog box. Available built-in templates are for example Nostalgia, OSX, OSX (no labels), Philadelphia, etc. Customized forms built in the Advanced pages of the Form Wizard can be added as templates.

This command is useful when you use the AP Add table and fields command, because you can determine the template to use for the default forms.

If the command has been executed properly, it returns 0. Otherwise, it returns an error code.

Example

This example creates a window containing all the available templates:

```

    $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)

```

See Also

AP Add table and fields.

AP Create relation (sourceTableNum; sourceFieldNum; destTableNum; destFieldNum) → Longint

Parameter	Type		Description
sourceTableNum	Longint	→	Number of the table from which the relation starts
sourceFieldNum	Longint	→	Number of the field from which the relation starts
destTableNum	Longint	→	Number of the table where the relation arrives
destFieldNum	Longint	→	Number of the field where the relation arrives
Function result	Longint	←	Error code (0=no error)

Description

The AP Create relation command allows you to create a relation between two tables. In sourceTableNum and sourceFieldNum, pass the numbers of the source table and field, and in destTableNum and destFieldNum, pass the numbers of the destination table and field.

The relation is created with standard properties. You can use 4D relation management commands to get or set these properties.

If the operation is completed successfully, the function returns 0. Otherwise, it returns an error code.

Example

This example creates a relation from table 1 to table 2:

```

C_LONGINT($SourceTable;$SourceField;$DestTable;$DestField)
$SourceTable:=1
$SourceField:=2
$DestTable:=2
$DestField:=2
⇒ $error:=AP Create relation ($SourceTable;$SourceField;$DestTable;$DestField)

```

AP Create method (methodName; propertiesArray; methodCode{; folderName}) → Longint

Parameter	Type	Description
methodName	String (31)	→ Name of the method (31 chars)
propertiesArray	Longint Array	→ Array of properties (4 elements)
methodCode	BLOB	→ BLOB containing the method text
folderName	String	→ Explorer folder in which the method will be created
Function result	Longint	← Error code (0=no error)

Description

The AP Create method command lets you add a project method in a 4th Dimension database structure (interpreted databases only).

Pass the name of the method in `methodName`. This name can be up to 31 characters long. It can contain any combination of letters, numbers, spaces and underlines with respect to the standard 4D objects naming rules. If the name contains more than 31 characters, it will be truncated to 31.

The `propertiesArray` parameter allows defining the method properties. It must have been declared previously as a longint array and must contain 4 items:

- pass 1 in `propertiesArray{1}` if the method must be visible and 0 if not.
- pass 1 in `propertiesArray{2}` if the method must be available for 4DACTION, 4DMETHOD and 4DSCRIPT, and 0 if not.
- pass 1 in `propertiesArray{3}` if the method must be offered as a Web Service and 0 if not.
- in case of `propertiesArray{3}=1`, pass 1 in `propertiesArray{4}` if the method must be published in WSDL and 0 for the other cases.

Pass a BLOB containing the text of the method in `methodCode`. If you use the TEXT TO BLOB command to fill the BLOB, pass C string (or 0) in the third parameter of this command.

In `folderName`, pass the name of the Explorer folder in which the method will be created. These folders are managed on the Home page of the Explorer window and allow you to organize the objects in a customized manner. This parameter is optional; if it is omitted, the method will be created in the "Default Project Methods" folder.

If the operation is completed successfully, the function returns 0. Otherwise, it returns an error code:

- 1 = A bad parameter type has been passed.
- 2 = The database runs in compiled mode.
- 3 = The method name is an empty string.

AP SET WEB FILTERS (filterIn; filterOut; charSet)

Parameter	Type		Description
filterIn	String	→	Name of 4D method to call
filterOut	String	→	Name of 4D method to call
charSet	Longint	→	Character set to use with outgoing filter

Description

The AP SET WEB FILTERS command specifies the 4D methods called when data is incoming from or outgoing to an HTTP stream.

In a client/server configuration, this command must be called from a method executed on the server. These filters are applied for all processes.

To specify no incoming filter, pass an empty string as the filterIn parameter. To specify no outgoing filter, pass an empty string as the filterOut parameter.

The charSet parameter specifies the character set to use with the outgoing filter:

- 0: Data to filter are in the Web browser character set
- 1: Data to filter are in the Macintosh character set

Examples

1. Incoming filter: this method is called each time the Web browser sends a request. It is designed to collect information rather than to filter data.

```

    \ Incoming filter sample
    \ to know the address of the request sender
C_BLOB($0;$1)
    \ $1 contains data coming from the browser
    \ $0 contains data to return to 4D
    \ We assume that the request is less than 32Kb
    $stream:=BLOB to text($1;Text without length)
    $pos:=Position("Host:";$stream)
If ($pos>0)
    $host:=Substring($stream;$pos+5;Length($stream))
    $pos2:=Position(Char(13)+Char(10);$host)
    \ The host address is included between Host: and the carriage return.
    $hostaddress:=Substring($host;1;$pos2-1)
    \ ...
End if
    $0:=$1 \ return data as they are
  
```


2. Outgoing filter: this is called each time 4D returns a reply to the browser (except the pictures). This function is designed to modify the HTTP stream header, or the HTML data of the page.

```
  ` Sample of outgoing filter
  ` which inserts a text in a page
C_BLOB($0;$1)
  ` $1 contains data generated by 4D
  ` $0 contains data to send to Web browser
  ` We assume that the page is less than 32Kb
$0:=$1 `copy the data
$stream:=BLOB to text($0;Text without length)
$pos:=Pos("<BODY>";$stream)
If ($pos>0)
  $text:="<BR><B>The text</B><BR><BR>"
  TEXT TO BLOB($text;$blob;Text without length)
  $pos:=$pos+5 `Insert text after <BODY>
  INSERT IN BLOB($0;$pos;Length($text))
  $postxt:=0
  COPY BLOB($blob;$0;$postxt;$pos;Length($text))
End if
```

AP Get tips state → Longint

Parameter	Type	Description
-----------	------	-------------

This command does not require any parameters

Function result	Longint	←	Tip display status
-----------------	---------	---	--------------------

Description

AP Get tips state returns 1 if tips are displayed; otherwise it returns 0.

In versions of 4D prior to 6.0.5, this command always returns 0.

See Also

AP SET TIPS STATE.

AP SET TIPS STATE (status)

Parameter	Type		Description
status	Integer	→	Tips status to be set

Description

The AP SET TIPS STATE command allows activation or suppression of tips.

In versions of 4D prior to 6.0.5, this command does nothing.

See Also

AP Get Tips state.

AP Set palette (resNum) → Longint

Parameter	Type		Description
resNum	Integer	→	ID of resource "pltt"
Function result	Longint	←	Error code

Description

AP Set palette indicates the foreground palette used. The pltt ID is passed in the resNum parameter.

Once the command has been called, an error code is returned to indicate if the command has been executed properly.

This function works only on Macintosh.

AP SET PICT MODE (image; mode)

Parameter	Type		Description
image	Picture	→	Picture to modify
mode	Integer	→	Display mode

Description

The AP SET PICT MODE command enables you to set the picture display mode. Using programming, you can set the appropriate mode in User mode when double-clicking on the background picture. It is useful if you want to import an image from a file to display it in the background.

The following values are same as QuickDraw transfer modes:

0	srcCopy	Overwriting mode
1	srcOr	Transparent mode
2	srcXor	Invert mode
3	srcBic	Erase mode
4	notsrcCopy	Negative overwriting mode
5	notsrcOr	Negative transparent mode
6	notsrcXor	Negative invert mode
7	notsrcBic	Negative erase mode

AP Timestamp to GMT (localDate; localTime; gmtDate; gmtTime) → String

Parameter	Type		Description
localDate	Date	→	Local date to use for the conversion
localTime	Time	→	Local time to use for the conversion
gmtDate	Date	←	GMT date returned by the conversion
gmtTime	Time	←	GMT time returned by the conversion
Function result	String	←	GMT time in compliance with RFC standard

Description

AP Timestamp to GMT converts local date and time to GMT date and time based on the machine location (set with the Date/Time control panel). It returns a string in the following format: "Weekday, DD MM YYYY HH:MM:SS GMT"

This format is in compliance with RFC 822 standard.

Example

Given the date of Thursday 11/13/97 and the time of 2:35PM (Paris time zone), the function returns the string "Thu, 13 Nov 1997 13:35:00 GMT". \$gmtd is set to 11/13/97 and \$gmtt to 13:35:00.

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

AP Get picture type (storedPicture) → Longint

Parameter	Type		Description
storedPicture	Blob	→	Picture stored as a BLOB
Function result	Longint	←	Type of the picture

Description

AP Get picture type returns the type of the image stored in the BLOB storedPicture:

- 1 unknown type
- 0 'PICT' picture
- 1 QuickTime compressed 'PICT' picture
- 2 JPEG picture
- 3 WMF picture
- 4 EMF picture
- 5 BMP picture
- 6 GIF picture

Example

This example stores a picture in a BLOB and then stores its type in \$type.

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

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

Parameter	Type		Description
totalMemory	Longint	←	Total memory
physicalMemory	Longint	←	Physical memory
freeMemory	Longint	←	Free memory
freeStack	Longint	←	Free stack

Description

The AP AVAILABLE MEMORY command returns information in bytes on the installed and free memory on the machine where it is executed.

The command returns the machine installed memory (total and physical), the free memory available for 4D and the free stack (rounded to the closest Kb) for the current process.

AP Select document (path; types) → Longint

Parameter	Type		Description
path	Text	←	Complete path to the file
types	Text Array/String Array	→	List of file types
Function result	Longint	←	Error code

Description

AP Select document displays a standard Open file dialog box that allows the user to select a file. Once the choice is validated, the path variable contains the complete access path to the selected file and the function returns 1.

The maximum length of the stored path is 1024 characters.

Unlike Open document, AP Select document does not open the document; it only stores the file complete path.

You can specify up to 16 document types in the types array. In this case, only the files that belong to the specified types are displayed in the Open file dialog box. If the types list is an empty array, all types of document will be displayed in the Open file dialog box.

- On Windows, the type of a document is designated by its extension.
- On Mac OS, the type of a document is designated by four characters.

Examples

1. The user can choose any type of document:

```

ARRAY TEXT($types;0)
⇒ $ok:=AP Select document($path;$types)
  If($ok=1)
    ProcessDoc($path)
  End if

```

2. Limiting the file type choice on Mac OS:

```

  `Restricting the choice to the AIFF file type (system sound)
ARRAY TEXT($types;2)
  $types{1}:="sfil"
  $types{2}:="AIFF"
  $path:=""
⇒ $ok:=AP Select document($path;$types)
  If($ok=1)
    ProcessSoundFile($path)

```

3. Limiting the file type choice on Windows:

```
    `Restricting the choice to the .WAV or .AIF file types (sound types)
ARRAY TEXT($types;2)
$types{1}:="WAV"
$types{2}:="AIF"
$path:=""
⇒ $ok:=AP Select document($path;$types)
If($ok=1)
    ProcessSoundFile($path)
End if
```

AP GET PARAM (option; value)

Parameter	Type		Description
option	Integer	→	Number of the option to read
value	Integer	←	Value for the option

Description

AP GET PARAM allows you to read the internal 4D options usually set with 4D Customizer Plus. Using this command, you can restore a value set by 4D Customizer Plus when you want to temporarily change it using the command AP SET PARAM.

For more information about the value parameter, please refer to the AP SET PARAM command description.

Example

The following code allows you to retrieve the value of the parameter 2:

```
⇒ AP GET PARAM(2;$param)
```

See Also

AP SET PARAM.

AP SET PARAM (option; value)

Parameter	Type		Description
option	Integer	→	Number of the option to set
value	Integer	→	New value for the option

Description

The AP SET PARAM command allows you to set the internal 4D options usually set with Customizer Plus. These options are stored in the CUST ID=0 resource which consists in a table of integer values numbered from 0 to n.

Note: Some default values may be different in 4D versions prior to 6.7.

The integer values are as follows:

- 0 - If the value is 0, it is not possible to open a desk accessory while a 4D printing progress window is displayed. If the value is 1, it is possible to open a desk accessory when such a window is displayed. The default value is 0.
- 1 - If the value is 0, it is not possible to open a desk accessory while a sequential operation (search, sort...) progress window is displayed. If the value is 1, opening a desk accessory is possible while such a window is displayed. In the Windows version, the value -3 signifies that 4D does not use an offscreen bitmap to perform graphical updates to the screen. All drawing operations are directly performed on the screen. If the value is -4, at start-up 4D allocates a permanent offscreen buffer corresponding to the size and depth of your screen; it then uses this buffer to perform offscreen bitmap-based graphical updates. The default value is 0.
Values other than those described specify that for each screen update 4D tries to allocate a corresponding buffer (provided there is enough memory to do so). If this is successful, 4D performs an offscreen bitmap-based update and then disposes of the buffer. Activating the offscreen bitmap-based updates allows you to eliminate screen blinking when many objects have to be redrawn.
A permanent offscreen buffer may take some space (i.e., 300K for a 640 x 480, 256 color screen), so for low memory configurations you may want to set this option to a value other than -3 or -4. Setting this option to -4 when memory is not an issue allows you to accelerate drawing operations. Deactivating offscreen bitmap-based updates (setting the option to -3) to take the screen updating scheme into account may solve incompatibility issues with 4D plug-ins that have not been updated under Windows.
- 2 - Sets the speed of rotation of the spinning cursor when 4D executes formulas. If the value is 0, the spinning cursor does not appear. A positive or negative value sets the speed of rotation to modulo 32. The average value to pass is 16. The default value is 0.

- 3 - Contains the ASCII code for the shortcut generating a new subrecord or record in the active included area of a data entry. The default value is 47 ([/] Key).
- 4 - Contains the value of the modifier key associated with option 3. The default value is 256 (**Command** Key). Possible values include: 256 for the **Command (Ctrl)** Key, 512 for **Shift** Key, 1024 for the **Caps Lock** Key, 2048 for the **Option (Alt)** Key.
- 5 - Contains the ASCII code for the shortcut generating the validation of a data entry. The default value is 3 (**Enter** Key).
- 6 - Contains the value indicating the modifier key associated with the character indicated by option 5. The default value is 0 (no modifier key).
- 7 - Contains the ASCII code for the shortcut generating the cancellation of a data entry. The default value is 46 ([.] Key).
- 8 - Contains the value indicating the modifier key associated with option 7. The default value is 256 (**Command** Key).
- 9 - Contains the ASCII code for the validation shortcut. The default value is 3 (**Enter** Key).
- 10 - Contains the value indicating the modifier key associated with option 9. The default value is 0 (no modifier key).
- 11 - Contains the ASCII code for the shortcut generating the cancellation of a dialog box. The default value is 46 ([.] Key).
- 12 - Contains the value indicating the modifier key associated with option 11. The default value is 256 (**Command** Key).
- 13 - Indicates the default positioning of the main 4th Dimension window. If the value is 0, the window is set to the size of the screen with its title bar visible under the menu bar. If the value is 1, the window is set to the size of the screen with its title bar hidden under the menu bar. If it is 2, the window is sized with the coordinates specified by options 14, 15, 16 and 17. If it is 3, the window is sized with the coordinates specified by options 14,15,16, and 17, but is centered in the middle of the screen. Finally, if the value is 4, the window appears with the size and location it had in the previous working session. The default value is 4.

The next four options have a default value equal to -1 and are only used if option 13 contains 2 or 3:

- 14 - Indicates the top coordinate of the 4D window.
- 15 - Indicates the left coordinate of the 4D window.
- 16 - Indicates the bottom coordinate of the 4D window.
- 17 - Indicates the right coordinate of the 4D window.

- 18 - Indicates if the main 4D window is resized when you switch to the Runtime environment. If the value is 0, the window is not resized. If it is 1, the window is resized according to option 13. The default value is 0.
- 19 - Indicates if the main 4D window is resized after the execution of a 4D method called from a custom menu. If the value is 0, the window is not resized. If it is 1, the window is resized according to option 13. The default value is 0.
- 20 - Reserved. Do not modify this option.
- 21 - Reserved. Do not modify this option.
- 22 - Reserved. Do not modify this option.
- 23 - Reserved. Do not modify this option.
- 24 - Reserved. Do not modify this option.
- 25 - Indicates if the During phase of a form method is called when you cancel a data entry. If the value is 0, the During phase is not triggered, otherwise it is triggered. The default value is 0.
- 26 - Indicates if the automatic switch of the read only/read write status of the tables is performed by operations such as display or printing of a selection. If the value is 0, the automatic switch is performed. If it is 1, the automatic switch is not performed. The default value is 0.

Options beyond 26 are reserved. Do not modify them.

Example

If you want the cursor to be a spinning wheel when it executes your commands, call:

⇒ *AP SET PARAM(2;16)*

See Also

AP GET PARAM.

5

4D_Pack : External Clock

The external area **_AP External clock** draws an analog clock using the PICT resource 17890 of the 4D_Pack plug-in. This resource draws the dial of the clock.



You can draw your own clock and then place a 17890 PICT resource in the structure file. This PICT resource can be created on a Macintosh using ResEdit™.

The picture can be of any size, as the size of the clock hands will be adjusted accordingly. In the same manner, the dial shape does not need to be contained in a square, but can be contained in a rectangle.

By default, the clock displays the System time; however, the **AP SET CLOCK** command allows you to set a delay between the time displayed by the clock and the current time. Thus, you could display different clocks corresponding to different time zones.

See Also

AP SET CLOCK.

AP SET CLOCK (clock; hour; minute)

Parameter	Type		Description
clock	Longint	→	_AP External clock external area
hour	Integer	→	Hour delay
minute	Integer	→	Minute delay

Description

The AP SET CLOCK command enables you to set the delay between the time of the clock displayed by the _AP External clock external area and the current time.

See Also

_AP External clock.

6

4D_Pack : Printing

AP PrintRec to text (text) → Longint

Parameter	Type		Description
text	Text	←	Text that contains the parameters
Function result	Longint	←	Error code -1, if you do not have access to the parameters -2, if the text cannot be re-allocated -3, if the text cannot be correctly interpreted

Description

Compatibility note: Starting with version 6.8 of 4D Pack, this command no longer works under MacOS.

Under Windows, this command returns a text variable containing the Windows printing settings.

You can then analyze byte by byte these settings, modify them, then use them with the AP Text to PrintRec. Thanks to this command, you can use the same printing settings between two working sessions or use different settings according to the pages.

The error code returned indicates if the command has been executed properly.

See Also

AP Text to PrintRec.

AP Text to PrintRec (text) → Longint

Parameter	Type		Description
text	Text	→	Text that contain the parameters
Function result	Longint	←	Error code -1, if you have no access to printing settings -2, if the text does not contain 120 characters

Description

Compatibility note: Starting with version 6.8 of 4D Pack, this command no longer works under MacOS.

AP Text to PrintRec accepts a text expression containing the Windows printing parameters to be passed as a parameter. These parameters then become the new printing settings.

Use the AP PrValidate function to make sure that these parameters are correct. You should also take into account the byte swapping under Windows in fields that contain integers or long integers.

The error code returned lets you know if the command has been correctly executed.

See Also

AP PrValidate.

AP Print settings to BLOB (printSettings) → Longint

Parameter	Type		Description
printSettings	BLOB	→	BLOB to store the print settings
Function result	Longint	←	Error code: 1 = Operation successful 0 = No current printer

Description

The AP Print settings to BLOB command stores the current print settings of 4th Dimension in the printSettings BLOB.

The printSettings parameter stores all the settings used for printing:

- Layout parameters: paper, orientation, scale,
- Print parameters as such: number of copies, paper source, etc.

The BLOB generated must not be modified by programming; it can only be used by the AP BLOB to print settings command.

The command returns 1 if the BLOB has been correctly generated and 0 if no current printer is selected.

See also

AP BLOB to print settings.

AP BLOB to print settings (printSettings{; paramType}) → Longint

Parameter	Type	Description
printSettings	BLOB	→ BLOB containing the print settings
paramType	Longint	→ 0 = Layout and print, 1 = Print
Function result	Longint	← 1 = Operation successful, 0 = No current printer, -1 = Incorrect parameters

Description

The AP BLOB to print settings command replaces the current print settings of 4th Dimension by the parameters stored in the printSettings BLOB. This BLOB must have been generated by the AP Print settings to BLOB command.

The printSettings parameter contains two types of parameters:

- Layout parameters: paper, orientation, scale,
- Print parameters as such: number of copies, paper source, etc.

You can choose to load only the print parameters.

If you pass 0 in paramType or omit this parameter, all the parameters are used. If you pass 1, only the print parameters are used; the layout parameters remain unchanged.

The new print parameters are applied to the entire database and for the whole session, as long as no command such as PAGE SETUP, SET PRINT OPTION or PRINT SELECTION without the > parameter modifies them.

The parameters set are used more particularly by the PRINT SELECTION, PRINT LABEL, PRINT RECORD, Print form and QR REPORT commands, as well as by the menu commands of 4th Dimension, including those of the Design environment.

The PRINT SELECTION, PRINT LABEL and PRINT RECORD commands must be called with the > parameter (if applicable) in order for the settings defined by AP BLOB to print settings to be kept.

Note: Print settings are not formatted in the same way under Windows and MacOS. As a result, the compatibility of the printSettings BLOB between the two platforms is not guaranteed.

The command returns 1 if the BLOB has been correctly loaded, 0 if no current printer is selected and -1 if the BLOB is incorrect.

See also

AP Print settings to BLOB.

AP PrValidate → Integer

Parameter	Type	Description
------------------	-------------	--------------------

This command does not require any parameters

Function result	Integer	← Error code 1, if the printing settings are correct 0, if the printing settings are not correct -1, if you cannot have access to the settings
-----------------	---------	---

Description

AP PrValidate enables you to ensure that the printing settings modified with the AP Text to PrintRec are correct.

If the settings are incorrect, we advise you to use the AP PrintDefault command which returns to the default settings.

See Also

AP Text to PrintRec.

AP PrintDefault → Integer

Parameter	Type	Description
This command does not require any parameters		
Function result	Integer	← Error -1, if you cannot have access to these parameters

Description

AP PrintDefault enables you to return to the default printout settings.

If you cannot have access to the parameters, this function returns -1.

7

4D_Pack : ANSI Streams

AP fopen (file | port; mode) → Long integer

Parameter	Type		Description
file port	String	→	Name of the file or port to be opened
mode	Integer	→	Opening mode 0 = write only 1 = read only
Function result	Long integer	←	Value to be used with AP FPRINT, or 0 if the file or port cannot be open

Description

The AP fopen command calls the “fopen” ANSI C command, which enables the opening of serial or parallel ports (on Windows) or the direct creation of files.

The value returned by this function is then used with the AP FPRINT command to write through the port, or to write the file. The opened port or file can then be closed using the command AP FCLOSE.

Note: On MacOS, this function works only with the direct creation of files.

Examples

1. Writing "hello world" on the parallel port:

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

2. Writing "hello world" on the serial port:

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

3. Writing "hello world" in a file:

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

4. Reading a file:

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

See Also

AP FPRINT.

AP FPRINT (file | port; data)

Parameter	Type		Description
file port	String	→	Port or file opened by the command AP fopen
data	Text	→	Data to be sent to the port or file

Description

The AP FPRINT command sends the specified text to the port or file opened by the AP fopen command.

data is sent as it is without any conversion. Since 4D uses the Macintosh ASCII table internally, you must perform the required conversion on Windows. To do so, use the Mac to Win 4D command.

See Also

AP FCLOSE, AP fopen, Mac to Win.

AP FCLOSE (port | file)

Parameter	Type		Description
port file	String	→	Name of the file or port

Description

The AP FCLOSE command closes the port or file previously opened using the AP fopen command.

See also

AP fopen, AP FPRINT.

AP fread (port | file) → Text

Parameter	Type		Description
port file	Longint	→	Port or file opened with AP fopen
Function result	Text	←	Data from the port or file

Description

AP fread reads the data from the port or file opened with AP fopen and returns it as text.

Data is returned without any conversion. Since 4D uses the Macintosh ASCII table internally, you must perform the required conversion on Windows. To do so, use the Win to Mac 4D command.

See Also

AP fopen, Win to Mac.

8

4D_Pack : Picture Files

AP Read Picture File (file; imageVar) → Integer

Parameter	Type		Description
file	String	→	Pathname to PICT document
imageVar	Picture	←	4D picture variable that will contain the read picture
Function result	Integer	←	0, if the operation was successful; otherwise a system error code

Description

AP Read Picture File loads into the imageVar picture variable the picture contained in the PICT document whose pathname is passed in the file parameter.

If the operation is completed successfully, the function returns 0; otherwise it returns a negative error code (i.e., -43 for file not found).

Note: This command recognizes the image format regardless of the file extension (under Windows).

This command can read files in the following formats:

On both Windows and Macintosh

• **BMP files**

This bitmap format is converted to PICT format within 4D. Pictures are then available for display on both platforms. Supported subformats are black and white, 16 colors, 256 colors, 24 and 32 bits. RLE compression and 16-bit color are not supported.

• **JPEG files**

This file format enables a high level of compression for photographic pictures. AP Read Picture File generates a PICT image with no compression. It does not require QuickTime to be installed. It generates a 24-bit or 256 grayscale level that can be displayed on both platforms.

Note: JPEG pictures are stored using a high compression level. Unpacking and uploading them may require a large amount of memory, even if the JPEG file is small. If the memory allocated to 4D is low, you should take this into account. In Windows, the memory temporarily allocated to unpacking is withdrawn from the memory allocated to the system. For such an operation to be successful, your machine must have enough memory. The size of the memory buffer allocated to unpacking is defined as follows: 3 X Width X Height. The resulting PICT image will have approximately the same size.

- **GIF files**

This file format is converted to PICT inside 4D. Therefore, images are correctly displayed on both Macintosh and Windows platforms. Usable formats are: black and white, 16 and 256 colors. This format is often used to publish images on the Web.

- **PICT files uncompressed or compressed with QuickTime**

PICT is the standard picture format on the Macintosh; all Macintosh drawing applications should be able to export images in PICT format. These files can be generated from a Macintosh or Windows application such as PhotoShop or CorelDraw. Using PICT format inside a database allows you to view these images on both Macintosh and Windows. On Windows, the PICT file extension is either .PIC or .PCT.

On Windows only

- **WMF files (Windows Metafile)**

These files must be "positionable" files, and include a header describing the picture size and its resolution. If no header is available, the command will not be able to read the picture file. WMF files are to the Windows platform what PICT files are to the MacOS platform; they can contain both vectorial and bitmap data (drawing and painting) that correspond to each system. The two main advantages of WMF files on Windows are the faster display speed (no conversion is required) and their universal use. All Windows applications can export in this format. However, keep in mind that using this format prevents you from displaying the pictures on the Macintosh platform.

- **EMF files (Windows Enhanced Metafile)**

This format is an improvement on WMF. Future Windows applications are likely to support it. The main advantages of this format are enhanced basic elements, such as Beziers and transformations.

Note: If you import EMF or WMF images, they will not be displayed on a Macintosh.

AP Save BMP 8 bits (image; file) → Longint

Parameter	Type		Description
image	Picture	→	Name of a 4D picture variable
file	Text	→	Name of the file in which the BMP will be saved
Function result	Longint	←	0, if the operation was successful; otherwise a system error code

Description

AP Save BMP 8 bits saves any 4D picture variable in a regular 8-bit BMP picture (256 colors and no compression). If an identically named file is located where you want to save the picture, it is replaced without warning. This command works on both Macintosh and Windows platforms.

If the operation is successfully completed, the function returns 0; otherwise, it returns a negative error code.

See also

AP Save GIF.

AP Read Picture BLOB (file; image) → Longint

Parameter	Type		Description
file	BLOB	→	BLOB containing the picture file
image	Picture	←	Image that will contain the read picture
Function result	Longint	←	0, if the operation was successful; otherwise a system error code

Description

If you store a picture file in a BLOB (using the DOCUMENT TO BLOB command), AP Read Picture BLOB allows you to retrieve this image in the image parameter. You can then display the picture in 4D.

Images are stored in their original formats and can be exported without alteration (using the command BLOB TO DOCUMENT). For example, a compressed JPEG image takes a few Kb; the same image, uncompressed, can take several Mb when stored in a 4D picture.

The file formats accepted by this command are the same as those accepted by AP Read picture file.

If the operation is completed successfully, the function returns 0; otherwise, it returns a negative error code (e.g., -108 for memory full error).

See also

BLOB TO DOCUMENT.

AP Save GIF (image; file) → Longint

Parameter	Type		Description
image	Picture	→	Name of a 4D picture variable
file	Text	→	Name of the file in which the GIF will be saved
Function result	Longint	←	0, if the operation was successful; otherwise a system error code

Description

AP Save GIF saves any 4D picture variable in a GIF picture. It works on both Macintosh and Windows platforms.

If the operation is successfully completed, the function returns 0; otherwise, it returns a negative error code.

This command requires 4D version 6.0.5 or later. In earlier versions of 4D, the command returns -1.

See also

AP Save BMP 8 bits.

Command Index

_AP External clock.....57
%AP PICT displayer.....15

A

AP Add table and fields.....34
AP AVAILABLE MEMORY.....48

B

AP BLOB to print settings.....64

C

AP CLOSE HELP.....24
AP Create method.....39
AP Create relation.....38

F

AP FCLOSE.....72
AP fopen.....69
AP FPRINT.....71
AP fread.....73
AP FULL SCREEN.....17

G

AP Get field infos.....33
AP GET PARAM.....51
AP Get picture type.....47
AP Get table info.....32

AP Get templates.....	37
AP Get tips state.....	42

H

AP HELP INDEX.....	22
AP HELP ON HELP.....	23
AP HELP ON KEY.....	21

N

AP NORMAL SCREEN.....	18
-----------------------	----

P

AP PICT DRAGGER.....	14
AP PICT UPDATER.....	16
AP Print settings to BLOB.....	63
AP PrintDefault.....	66
AP PrintRec to text.....	61
AP PrValidate.....	65

R

AP Read Picture BLOB.....	80
AP Read Picture File.....	77
AP Rect dragger.....	13

S

AP Save BMP 8 bits.....	79
AP Save GIF.....	81
AP Select document.....	49
AP SET CLOCK.....	58
AP Set palette.....	44

AP SET PARAM.....	52
AP SET PICT MODE.....	45
AP SET TIPS STATE.....	43
AP SET WEB FILTERS.....	40
AP ShellExecute.....	29
AP Sublaunch.....	27

T

AP Text to PrintRec.....	62
AP Timestamp to GMT.....	46
AP Toolbar installed.....	31

