

4D Plug-ins 2004

Upgrade
Windows®/Mac™ OS



4D Plug-ins 2004 Upgrade

Copyright© 1985 - 2004 4D SA / 4D, Inc.
All rights reserved.

The software and the manual are copyrighted and may not be reproduced in whole or in part except for the personal licensee's use and solely in accordance with the contractual terms. This includes copying the electronic media, archiving, or using the software in any manner other than that provided for in the Software license Agreement.

4D, 4D Draw, 4D Write, 4D Insider, 4th Dimension®, 4D Server, 4D Backup and the 4th Dimension and 4D logos are registered trademarks of 4D S.A.

Windows, Windows NT, Windows XP and Microsoft are registered trademarks of Microsoft Corporation.

Apple, Macintosh, MacOS, MacOS X, QuickTime are trademarks or registered trademarks of Apple Computer Inc.

Mac2Win Software Copyright © 1990-2004, 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 (ey@cryptsoft.com)
4th Dimension includes software written by Tim Hudson (tjh@cryptsoft.com).

Spellchecker © Copyright SYNAPSE Développement, Toulouse, France, 1994-2004.

ACROBAT © Copyright 1987-2004, Secret Commercial Adobe Systems Inc. All rights reserved. ACROBAT is a registered trademark of Adobe Systems Inc.

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

Contents

Chapter 1	Introduction.5
Chapter 2	4D Write 2004.7
	Support of drag and drop	7
	Default drag and drop	7
	Configuring 4D objects for drag and drop	8
	Programmed management of drag and drop	9
	WR SET AREA PROPERTY, WR GET AREA PROPERTY	10
	WR ON EVENT	10
	New commands	10
	WR GET DRAG SOURCE	11
	WR GET DROP TARGET.	11
	WR GET DROP INFO	12
	New print options	12
	WR SET PRINT OPTION.	13
	WR GET PRINT OPTION	16
	Support of right mouse click	18
	WR ON EVENT	19
	Zoom for pages	19
	WR SET AREA PROPERTY, WR GET AREA PROPERTY	20
	Options for pictures pasted into pages	20
	WR INSERT PICTURE	21
	WR SET PICTURE IN PAGE INFO	22
	WR GET PICTURE IN PAGE INFO.	22
	Spellchecking	23
	WR SET AREA PROPERTY, WR GET AREA PROPERTY	24
Chapter 3	4D View 2004	25
	Sorting columns	25
	PV SET AREA PROPERTY, PV Get area property	26
	PV SORT COLUMN	26
	New pv on column sort event.	27
	Automatic word wrap	29
	Resizing events.	30

Chapter 4	4D Draw 2004	31
	New command	31
	DR GET MOUSE	31
	New global preferences.	32
	DR SET GLOBAL PREFERENCES	33
	DR GET GLOBAL PREFERENCES.	34
	DR SET ENTERABLE	34
	Management of printing in a single print job.	35
	DR EXPORT MODE	35
Chapter 5	4D Pack 2004	37
	4D_Pack: Utilities.	37
	AP Add table and fields	37
	4D_Pack: Printing	39
	AP Print settings to BLOB	39
	AP BLOB to print settings	40

1

Introduction

This manual describes the new features and modifications made to the 2004 versions of the 4th Dimension environment plug-ins. These new features concern the following 4D plug-ins:

- 4D Write
- 4D View
- 4D Draw
- 4D Pack

Keep in mind that the installation of plug-ins has been modified in 4th Dimension 2004. This is covered in the “Application Architecture” chapter of the 4th Dimension 2004 *Upgrade* manual.

2

4D Write 2004

Version 2004 of 4D Write includes the following new features and modifications:


- Support of drag and drop,
- New print options,
- Support of right mouse click,
- Zoom for pages,
- New options for pictures pasted into pages,
- On-the-fly spellchecking.

Support of drag and drop

4D Write 2004 now lets you carry out drag-and-drop operations within the same 4D Write area, between two different 4D Write areas or between a 4D Write area and a 4th Dimension area.

Drag and drop can be used in default mode or programmed using the new commands of the “WR Drag and Drop” theme.

Default drag and drop

By default, 4D Write offers automatic handling of drag and drop, based on the moving or copying of text or pictures. A selection of text or a picture can be moved using the mouse. Under Windows, the cursor changes to  in order to indicate that a drag and drop is being carried out. Under MacOS, a shadow follows the object being moved. An insertion point indicates the destination of the moved object:

Triggers are **automatically** called by the engine

Triggers are called automatically by the engine

When a picture is inserted in a 4D Write area using drag and drop, it is automatically pasted into the text.

Note Data are *moved* when the drag and drop is carried out within the same or between two 4D Write areas, i.e. they are removed from the original area. If you only want to *copy* the data, hold down the **Ctrl** (Windows) or **Command** (MacOS) key during the operation.

With this type of drag and drop, no specific programming is required; you just need to apply the appropriate “Draggable” and “Droppable” properties when you want to drag and drop inside 4D forms (see below).

Configuring 4D objects for drag and drop

You can drag and drop data between 4D Write areas and 4th Dimension objects.

All types of 4th Dimension fields (except for BLOBs and subtables) and variables (except for BLOBs) can be dropped into 4D Write areas and vice versa. They will be inserted automatically into the 4D Write area as text or pictures according to their original type.

Warning: To drag textual data from a 4th Dimension field or variable into a 4D Write area, you must hold down the **Alt** (Windows) or **Option** (MacOS) key during the operation.

Keep in mind that it is not possible to drag and drop a selection of text from a 4th Dimension area into 4D Write, only the entire contents of the object can be copied.

- In 4th Dimension, if you want to drag and drop objects between a 4D Write area and a 4th Dimension object, the “Draggable” property has to be selected for each object that must be dragged and dropped.
- If the 4D Write area is included in a form, the “Droppable” property has to be selected for the area if it must receive 4D objects or elements coming from other 4D Write areas.
The “Draggable” property must be selected if the elements of the area will need to be dragged.
- For external windows of 4D Write, drag and drop is enabled by default. You must use the WR SET AREA PROPERTY command to control drag and drop.

Programmed management of drag and drop

The default drag and drop of 4D Write lets you set up intuitive interfaces and in general contributes to better ergonomics.

However, in certain cases, you may want to customize these mechanisms, more particularly for:

- Using drag and drop from other form objects (hierarchical lists, scrollable areas, etc.)
- Controlling the effect of a drag and drop, for example when copying the dragged data to several different locations.

In this case, you must combine the commands for managing drag and drop in 4th Dimension with those of 4D Write.

First of all, you need to be sure that the On Drag Over and/or On Drop form events have been checked for the objects used.

You can set the drag and drop properties for the 4D Write area using the WR GET AREA PROPERTY and WR SET AREA PROPERTY commands. If the 4D Write area is included in a form, you can use the On Drag Over and/or On Drop form events of the included area object; if it is an external window, you must manage the events specifically using the WR ON EVENT command.

If you want to control the type of 4D objects being moved precisely, you must use the 4th Dimension DRAG AND DROP PROPERTIES command. 4th Dimension commands let you carry out any type of action in response to a drag and drop.

In the case of a drag and drop between two 4D Write areas, you can find out the area from which the data have been dragged using the new WR GET DRAG SOURCE command.

You can find out the area into which the 4D object has been dropped using the new WR GET DROP TARGET command as well as the exact position of the insertion point when the object was dropped (new WR GET DROP INFO command): area (header, footer, body) and location of cursor.

WR SET AREA PROPERTY, WR GET AREA PROPERTY

WR SET AREA PROPERTY (area; option; value{; stringValue})

WR GET AREA PROPERTY (area; option; value{; stringValue})

The *option* parameter accepts two new constants concerning drag and drop, which are available in the “WR Area properties” theme. These constants let you set and get the drag and drop authorizations in the area by programming. Keep in mind that the values set using this parameter replace any properties that have been set in the form.

The possible values are as follows:

Constants

Values

wr allow drag (14)

0 = No value can be dragged from the area (even if drag and drop is allowed)
1 = It is possible to drag a selection from the area

wr allow drop (15)

0 = No value can be dropped into the area (even if drag and drop is allowed)
1 = It is possible to drop a selection in the area

WR ON EVENT

WR ON EVENT (area; event; method)

The *event* parameter accepts two new constants concerning drag and drop, which are found in the “WR Events” theme:

- wr on drag (11)
- wr on drop (12)

These constants let you manage drag and drop events occurring more particularly in the 4D Write external windows by programming.

New commands

Three new commands, located in the “WR Drag and Drop” theme, let you meet the specific needs related to the programmed management of drag and drop in 4D Write.

WR GET DRAG SOURCE

WR GET DRAG SOURCE(area; source)

Parameters	Type	Description
area	Longint	→ 4D Write area
source	Pointer	← Pointer to source object of drag and drop

The WR GET DRAG SOURCE command returns a pointer to the 4D field, 4D variable or the reference of the 4D Write area, which is the source of the drag and drop.

This command must be called within a wr on drag event. If the drag and drop originates from a 4D object, you can use the DRAG AND DROP PROPERTIES command to get additional information about the type of object being moved.

WR GET DROP TARGET

WR GET DROP TARGET(area; target)

Parameters	Type	Description
area	Longint	→ 4D Write area
target	Pointer	← Pointer to target object of drag and drop

The WR GET DROP TARGET command returns a pointer to the 4D field, 4D variable or the reference of the 4D Write area, in which the drop took place.

This command must be called within a wr on drop event. If the drop takes place in a 4D Write area, you can use the [WR GET DROP INFO](#) command to get additional information about the area and location of the drop. If the drop takes place in a 4D object, you must use 4th Dimension commands such as Drop position in order to manage the operation.

WR GET DROP INFO WR GET DROP INFO(area; frame; cursor)

Parameters	Type	Description
area	Longint	→ 4D Write area
frame	Longint	← Part of document
cursor	Longint	← Position in text

The WR GET DROP INFO command returns information specifying the exact location where the dragged data were dropped. This command must be called within a wr on drop event.

The *frame* parameter returns the part of the document in which the data were dropped. You can compare the value received to the constants of the “WR Frames” theme:

Constant	Type	Value
wr text frame	Longint	0
wr right header	Longint	1
wr left header	Longint	3
wr right footer	Longint	2
wr left footer	Longint	4
wr first header	Longint	5
wr first footer	Longint	6

The *cursor* parameter returns the location of the insertion cursor among the characters in *area*.

New print options

4D Write 2004 lets you handle new print options for a specified area using two new commands located in the “WR Printing” theme: WR SET PRINT OPTION and WR GET PRINT OPTION.

It is possible to employ 4th Dimension commands used for setting and getting the current printer. Changing the printer does not modify the print options (except if a certain option is not available on the new printer).

WR SET PRINT OPTION

WR SET PRINT OPTION(area;option;value1 {; value2{; value3}})

Parameters	Type	Description
area	Longint	→ 4D Write area
option	Longint	→ Option number
value1	Number	→ Value 1 of the option
value2	Number	→ Value 2 of the option
value3	String	→ Value 3 of the option

The WR SET PRINT OPTION command is used to modify the value of a print option by programming for the 4D Write area designated by the *area* parameter. Each option defined using this command will remain applied to the 4D Write area until this area is erased. Options that are usually saved with 4D Write documents (such as orientation) are also saved.

The current print parameters of 4th Dimension and those of other 4D Write areas are not modified.

The *option* parameter lets you specify the option to be modified. You can pass either a value or one of the following predefined constants, located in the new “WR Print options” theme.

Pass the new value(s) of the specified *option* in the *value1* and (optionally) *value2* and *value3* parameters. The number and nature of the values to be passed depends on the type of option specified.

For more information about the options and their values, refer to the following table:

Option	4D Write constant	<i>value1</i>	<i>value2</i>	<i>value3</i>
1 - Paper	<u>wr paper option</u>	Height 0	Width 0	- Name
2 - Orientation	<u>wr orientation option</u>	1=Portrait, 2=Landscape	-	-
3 - Scale	<u>wr scale option</u>	Number (%)	-	-
4 - No. of copies	<u>wr number of copies option</u>	Number	-	-
5 - Paper source	<u>wr paper source option</u>	Windows only Index (number)	-	-
6 - Pages from	<u>wr pages from option</u>	Number (1=default)	-	-
7 - Pages to	<u>wr pages to option</u>	Number (-1=default, end of document)	-	-

Option	4D Write constant	value1	value2	value3
8 - Color, B/W	<u>wr color option</u>	1=B/W, 2=Color	-	-
9 - Destination	<u>wr destination option</u>	1=Printer, 2=File (PC)/PS (Mac), 3=PDF (Mac)	0	- Access path
11 - Double sided	<u>wr double sided option</u>	Windows only 0=Single-sided (standard) 1=Double-sided	- Binding: 0=Left (default), 1=Top	- -
12 - Spooler document name	<u>wr spooler document name option</u>	0	0	Name of document

- **wr paper option (1):** The list of all the names of available paper can be obtained using the 4D command PRINT OPTION VALUES.
You can either pass the name of the paper in *value3* (and, in this case, pass 0 in *value 1* and *value2*), or pass the paper height in *value1* and its width in *value2*. The width and height must be expressed in screen pixels.
- **wr orientation option (2):** You can pass either 1 (Portrait) or 2 (Landscape) in *value1*.
- **wr scale option (3):** Pass a percentage in *value1*. Be careful, some printers do not allow you to modify the scale. If you pass an invalid value, the property is reset to 100% at the time of printing.
- **wr number of copies option (4):** Pass the number of copies to be printed in *value1*.
- **wr paper source option (5):** Pass the number corresponding to the index, in the array of trays returned by the 4D command PRINT OPTION VALUES, of the paper tray to be used.

Note This option can only be used under Windows.

- **wr pages from option (6):** Pass the number of the page where you want printing to start in *value1*. The default value is 1.
- **wr pages to option (7):** Pass the number of the last page that you want to be printed in *value1*. If you pass -1, the entire document will be printed (-1 is equivalent to passing the last page of the document).

- **wr color option (8):** In *value1*, pass the code specifying the mode for handling color: 1=Black and white (monochrome), 2=Color.

Note This option can only be used under Windows.

- **wr destination option (9):** In *value1*, pass the code specifying the type of print destination: 1=Printer, 2=File (PC)/PS (Mac), 3= PDF file (MacOS only).

Always pass 0 in *value2*.

If *value1* is different from 1, pass the access path for the resulting document in *value3*. This path will be used until another path is specified. If a file with the same name already exists at the destination location, it will be replaced. Under Windows only: if you pass an empty string in *value3* or omit this parameter, a file saving dialog appears at the time of printing. If the operation fails, the Printer (1) settings are applied.

- **wr double sided option (11):** You can either pass 0 (Single-sided or standard), or 1 (Double-sided) in *value1*.

If *value1* equals 1, you can define the binding to be applied using *value2*: 0=Left binding (default value), 1=Top binding.

Note This option can only be used under Windows.

- **wr spooler document name option (12):** In *value3*, pass the name of the print document that must appear in the list of spooler documents. Pass 0 in *value1* and *value2*.

To use or restore standard operation (using the method name in case of a method, the table name for a record, etc.), pass an empty string in *value3*.

Warning: The name defined by this statement will be used for all the print documents of the session for as long as a new name or an empty string is not passed.

If the value passed for an *option* is invalid or if it is not available on the printer, the command returns an error (that you can intercept using an error-handling method installed by the WR ON ERROR command) and the current value of the option remains unchanged. The system variable OK is set to 1 if the command has been executed correctly; otherwise, it is set to 0.

See also: WR GET PRINT OPTION, PRINT OPTION VALUES (4D)

WR GET PRINT OPTION

WR GET PRINT OPTION(area;option;value1 {; value2 {; value3}))

Parameters	Type	Description
area	Longint	→ 4D Write area
option	Longint	→ Option number
value1	Number	← Value 1 of the option
value2	Number	← Value 2 of the option
value3	String	← Value 3 of the option

The WR GET PRINT OPTION command returns the current value(s) of a print *option*.

The *option* parameter enables you to specify the option to get. You can either pass a value or one of the following predefined constants, located in the new “WR Print options” theme:

Constant	Type	Value
wr paper option	Longint	1
wr orientation option	Longint	2
wr scale option	Longint	3
wr number of copies option	Longint	4
wr paper source option	Longint	5
wr pages from option	Longint	6
wr pages to option	Longint	7
wr color option	Longint	8
wr destination option	Longint	9
wr double-sided option	Longint	11
wr spooler document name option	Longint	12

The command returns, in the *value1* and (optionally) *value2* and *value3* parameters, the current value(s) of the specified *option*. For more information on options and possible values, refer to the description of the [WR SET PRINT OPTION](#) command. Note the following specific features of the WR GET PRINT OPTION command:

- *option* = 1 ([wr paper option](#)): Returns the name of the current paper in *value3* if this parameter is passed (in this case, *value1* and *value2* receive 0). If *value3* is omitted, the command returns the height and width respectively in *value1* and *value2*. Use the PRINT OPTION VALUES command to get the name, height and width of all the paper formats offered by the printer.
- *option* = 2 ([wr orientation option](#)): Returns 1 (Portrait) or 2 (Landscape). If a different orientation option is used, *value1* is set to 0.
- *option* = 5 ([wr paper source option](#)): In *value1*, returns the index, in the array of trays returned by the PRINT OPTION VALUES command, of the paper tray used (*value2* must be omitted).

Note This option can only be used under Windows.

- *option* = 6 and *option* = 7 ([wr pages from option](#) and [wr pages to option](#)): If all the pages are printed, the command returns 1 in *value1* for [wr pages from option](#) and -1 in *value1* for [wr pages to option](#).
- *option* = 8 ([wr color option](#)): Returns a code in *value1* specifying the mode for handling color: 1=Black and white (monochrome), 2=Color.

Note This option can only be used under Windows.

- *option* = 9 ([wr destination option](#)): If the current value is not in the predefined list, *value1* contains -1 and the system variable *OK* is set to 1. If an error occurs, *value1* and the system variable *OK* are set to 0. If *value1* contains a predefined value different from 1, *value3* contains the access path of the printed file. *value2* always contains 0.
- *option* = 11 ([wr double sided option](#)): Returns 0 (Standard or Single-sided, default value) or 1 (Double-sided) in *value1*. If *value1* equals 1, *value2* may return one of the following values: 0=Left binding (default), 1=Top binding.

Note This option can only be used under Windows.

- *option* = 12 (wr spooler document name option): Returns the name of the current print document in *value3*, if it has been defined previously (*value1* and *value2* receive 0). Otherwise, an empty string is returned.

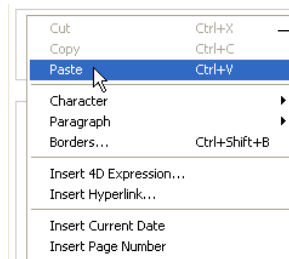
The system variable *OK* is set to 1 if the command has been executed correctly; otherwise, it is set to 0.

See also: [WR SET PRINT OPTION](#), [PRINT OPTION VALUES \(4D\)](#)

Support of right mouse click

Support of the **right click** (Windows and MacOS) and the **Control+click** combination (MacOS) has been modified in 4D Write 2004.

Now, the contents of the contextual menu that appears when the user right clicks in a 4D Write document is in conformity with word processing standards. The actions offered by the menu are different depending on the location of the click: header, footer, text, picture:



Example of contextual menu
(click in a header)

In addition, developers can customize the contents of this menu:

- Contextual menu commands are enabled or disabled at the same time as the corresponding commands in the standard menus when the **WR LOCK COMMAND** command is called.
- The **WR ON EVENT** command can be used to manage right clicks.

WR ON EVENT

WR ON EVENT (area; event; method)

Theme: WR Utilities.

The *event* parameter accepts a new constant concerning the **right click** (or the **Control+click** combination under MacOS), available in the “WR Events” theme:

- wr on right click (4)

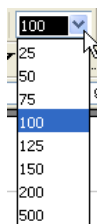
This constant can be used to manage the display of contextual menus in *area* by programming.

When the *event* parameter is equal to 4 (wr on right click), \$6 indicates the type of contextual menu about to be displayed (according to the location of the click):

- If \$6 equals 1, a type 1 contextual menu (click in header/footer) is displayed.
- If \$6 equals 2, a type 2 contextual menu (click in the text of the body area) is displayed.
- If \$6 equals 3, a type 3 contextual menu (click on a picture of the body area) is displayed.

Zoom for pages

It is now possible to set a zoom value for the display of each 4D Write area. The modification of this value is carried out using a combo box located in the **Standard** tool bar:



You can choose a predefined value for enlarging or reducing the zoom, or enter a custom value in % between 25 and 500.

The WR SET AREA PROPERTY and WR GET AREA PROPERTY commands can be used to manage this function by programming.

WR SET AREA PROPERTY, WR GET AREA PROPERTY

WR SET AREA PROPERTY (area; option; value{; stringValue})

WR GET AREA PROPERTY (area; option; value{; stringValue)

Theme: WR Area Options.

The *option* parameter accepts a new constant, available in the “WR Area properties” theme:

- wr zoom factor (13)

This constant is used to set and get the current zoom factor of the area by programming. The *value* parameter contains the percentage of the zoom (between 25 and 500).

Options for pictures pasted into pages

The options for inserting pictures into pages have been modified in order to improve support of exports in RTF format for documents containing pictures pasted in the page.

Now, the combination of options offered depends on the settings of the headers and footers in the Preferences:

- When the **Different on first page** option is checked in the Preferences, 4D Write lets you paste the picture into all the pages except for the first one.
- When the **Different on left and right pages** option is checked in the Preferences, 4D Write lets you paste the picture into only even- or odd-numbered pages.

Preferences

Headers and footers

Different on first page

Different on left and right pages



Insertion options

Paste Picture in Page

Place Picture

in current page

in all pages

Behind Text

Headers and footers

Different on first page

Different on left and right pages



Paste Picture in Page

Place Picture

in current page

in right pages (except page one)

in left pages (except page one)

Behind Text

Compatibility Note Documents created with a previous version of 4D Write are not converted; they keep their former settings by default.

The WR INSERT PICTURE, WR SET PICTURE IN PAGE INFO and WR GET PICTURE IN PAGE INFO commands have also been modified.

WR INSERT PICTURE WR INSERT PICTURE(area; picture{; destination{; horizPos{; verticalPos{; behind{; firstPage}}}}))

Theme: WR Picture Control.

The *destination* parameter now accepts new values:

- If *destination* is greater than 0, the picture will be visible on the page number *destination*. The value must take the start of numbering into account.
- If *destination* equals -11, the picture will be inserted into the page and will be displayed on the right-hand pages if the even- and odd-numbered headers are different; otherwise, on every page.
- If *destination* equals -12, the picture will be inserted into the page and will be displayed on the left-hand pages only if the even- and odd-numbered headers are different.
- If *destination* equals -4, the picture will be inserted into the page and visible on the current page (the one containing the cursor or the current selection).

For compatibility reasons, the other values offered in the previous version of the command (-1, -2 and -3) are still supported. Their use is nevertheless not recommended. The result obtained will depend on the parameters set for the headers and footers.

In the same way, the *firstPage* parameter is now only taken into account if the *destination* parameter is -1, -2 or -3. In all other cases, it is ignored and should be omitted.

WR SET PICTURE IN PAGE INFO

WR SET PICTURE IN PAGE INFO (area; pictureNumber; page; behind; firstPage; horizPos; verticalPos; width; height)

Theme: WR Picture Control.

The *page* parameter now accepts new values:

- If *page* equals -11, the picture will be visible on the right-hand pages if the even- and odd-numbered headers are different; otherwise, it will be visible on every page.
- If *page* equals -12, the picture will be visible on the left-hand pages if the even- and odd-numbered headers are different.
- If *page* equals -4, the value is not modified.

For compatibility reasons, the other values offered in the previous version of the command (-1, -2 and -3) are still supported. Their use is nevertheless not recommended. The result obtained will depend on the parameters set for the headers and footers.

In addition, the *firstPage* parameter was kept only for compatibility reasons. It is now only taken into account if the *page* parameter equals -1, -2 or -3. Otherwise, it is ignored. In order not to use it, pass -1 in the *firstPage* parameter.

WR GET PICTURE IN PAGE INFO

WR GET PICTURE IN PAGE INFO(area; pictureNumber; page; behind; firstPage; horizPos; verticalPos; width; height)

Theme: WR Picture Control.

The *page* and *firstPage* parameters have been modified in order to take into account the modifications made to the options for inserting pictures into pages. For more information, refer to the description of the [WR SET PICTURE IN PAGE INFO](#) command, page 22.

Spellchecking

4D Write 2004 includes a spellchecking utility, available in several languages.

The former spellchecker management mechanism (based on the CSAPI standard) is no longer supported:

- The **Spellchecking** page of the 4D Write Preferences dialog box has been removed,
- The **Language** command of the **Tools** menu of 4D Write has also been removed.

The language used for spellchecking is now the current language of the 4D application. The SET DICTIONARY command of the 4th Dimension language lets you change the current dictionary.

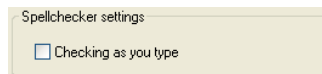
The spelling dialog box is identical to that of previous versions of 4D Write.

Two spellchecking methods are now available:

- Checking triggered using the **Spelling...** command in the **Tools** menu (identical to that of previous versions),
- On-the-fly spellchecking.
This new option causes any words whose spelling seems to be incorrect to be instantly underlined:

Triggers are variables _____ Incorrect spelling detected

To activate this mode, you must check the **Checking as you type** option located in the 4D Write Preferences:



The WR SET AREA PROPERTY and WR GET AREA PROPERTY commands are used to set and get the “on-the-fly spellchecking” mode by programming for a specific area.

**WR SET AREA
PROPERTY, WR GET
AREA PROPERTY**

WR SET AREA PROPERTY (area; option; value{; stringValue})

WR GET AREA PROPERTY (area; option; value{; stringValue)

Theme: WR Area Options.

The *option* parameter accepts a new constant, found in the “WR Area properties” theme:

- wr on the fly spellchecking (16)

This constant is used to set and get the on-the-fly spellchecking option by programming for the *area*, regardless of whether or not the option is checked in the Preferences.

3

4D View 2004

The new features offered in version 2004 of the 4D View plug-in concern the following subjects:

- Management of column sorting by programming,
- Automatic word wrap,
- Resizing events for rows and columns.

Sorting columns

4D View 2004 simplifies the support for sorting data in columns when their header is clicked.

To do this, the following new features are available:

- The PV SET AREA PROPERTY and PV Get area property commands (“PV Area” theme) accept a new area property: pv headers sort.
- A new command, PV SORT COLUMN, has been added to the “PV Cell Manipulation” theme, which allows you to sort a column directly.
- The new constants theme, “PV Header sort”, contains the possible sorting values.
- The new pv on column sort event lets you customize sorts.

PV SET AREA PROPERTY, PV Get area property

PV SET AREA PROPERTY (area; property; value)

PV Get area property (area; property) → Longint

Theme: PV Area

The *property* parameter accepts a new value that is used to set or get the authorization (or not) of standard sorting (managed by 4D View) of columns when their header is clicked. To do this, a new constant is available in the “PV Area properties” theme: pv headers sort.

pv headers sort

Lets you allow or forbid the standard sorting of data when a column header is clicked (dynamic or static data).

Associated values: Constants of the “PV Header sort” theme.

- pv sort not allowed (default value): 4D View does not carry out a standard sort when the user clicks on a column header (the sort can nevertheless be managed by the developer in a customized manner).
- pv sort allowed: 4D View carries out a standard sort when the user clicks on a column header. In this case, a symbol appears in the header in order to indicate the sort order. Successive clicks cause alternating ascending and descending sorts:



Sorting a dynamic column produces a synchronized sort of the other columns so that the records always remain in their initial state.

A sort on a static column only sorts that column.

PV SORT COLUMN

PV SORT COLUMN(area; column; order)

Parameters	Type	Description
area	Longint	→ 4D View area
column	Longint	→ Column number
order	Longint	→ Sort order

Theme: PV Cell Manipulation

The PV SORT COLUMN command is used to carry out a standard sort, in ascending or descending order, on a column set by the *column* parameter in the 4D View *area*.

Sorting a dynamic column produces a synchronized sort of the other columns so that the records always remain in their initial state. A sort on a static column only sorts that column.

To set the *order* parameter, use one of the following constants, located in the “PV Header sort” theme:

Constant	Type	Value
pv ascending sort	Longint	2
pv descending sort	Longint	3

- ▼ With the following statement, the column will be sorted in ascending order; keep in mind that the sort must first have been allowed using the PV SET AREA PROPERTY command. Otherwise, this line of command will do nothing:

PV SORT COLUMN(area; 2; pv ascending sort)

New pv on column sort event

A new event is available in the “PV Event” theme of the constants: pv on column sort.

The pv on column sort event is generated just after a column has been sorted. This way, it can be used to control user actions.

When this event is used with the PV ON EVENT command, the *method* called receives a value indicating the sort order in the \$6 parameter. This value can be compared with the following constants, located in the “PV Header sort” theme:

Constant	Type	Value
pv ascending sort	Longint	2
pv descending sort	Longint	3

- ▼ The user clicks on the column header to carry out a sort. The *PM_Event* method is used to find out which column has been sorted and in what order.

Installation of the method that will be called during the *pv* on column sort event:

```
PV ON EVENT(area;pv on column sort;"PM_Event")
```

Here is the *PM_Event* method:

```
C_BOOLEAN($0)
C_LONGINT($1; $2; $3; $4; $5; $6)
C_STRING(12; $SortOrder)
If ($2=pv on column sort)
  Case of
    :($6=pv ascending sort)
      $SortOrder:="ascending"
    :($6=pv descending sort)
      $SortOrder:="descending"
  End case
  ALERT ("The sort was carried out on the column "+String($4)+" in "
    +$SortOrder+" order")
End if
```

Sorts and double-clicks

A double-click on a column header causes the column to be resized. However, a double-click generates a sequence of two events: *pv* on clicked then *pv* on double clicked.

As a result, if sorting has been allowed, a double-click on a header first causes the sorting of the column, then its resizing. If you want a double-click to only cause resizing of the column, you must intercept and remove the *pv* on clicked event, which is generated just before the sort is carried out.

To do this, simply install a method that will be called during the *pv* on clicked event:

```
PV ON EVENT(area;pv on clicked;"PM_Event")
```

Here is the *PM_Event* method:

```
C_BOOLEAN($0)
C_LONGINT($1; $2; $3; $4; $5; $6)
If ($2=pv on clicked)
  $0:=True `The event is ignored and the sort is not carried out
End if
```

Automatic word wrap

4D View now lets you enable an automatic word wrap function when the contents of a cell exceeds its width. By default, when this function is not enabled, the cell contents may run over into the adjacent cells.

To enable word wrapping, a new constant has been added to the “PV Style properties” theme: **pv style automatic word wrap** (33).

The values that can be associated with this constant are pv value on and pv value off.

This constant can be used with the following commands:

PV SET CELL PROPERTY

PV SET RANGE PROPERTY

PV SET STYLE PROPERTY

- ▼ Given the following 4D View area:

	A	B
1		
2	Anomalies observed this year	

After execution of this statement:

PV SET CELL PROPERTY(area;1;2;pv style automatic word wrap;
pv value on)

... the contents of cell 1;2 automatically move to the next line:

	A	B
1		
2	Anomalies observed this year	

If the column is resized, the automatic line breaks will be recalculated if necessary.

Note Forced line breaks can be inserted manually using **Ctrl+Carriage return** (Windows) or **Command+Carriage return** (MacOS).

Resizing events

Two new events related to resizing have been added in the “PV Event” theme of the constants:

pv on column resize (12)

pv on row resize (13)

These events are sent when a column or row is resized by the user.

Note These events are not sent if the columns or rows are resized by programming (using the PV SET COLUMNS WIDTH or PV SET ROWS HEIGHT commands).

4

4D Draw 2004

Version 2004 of the 4D Draw plug-in includes three new features:

- New DR GET MOUSE command,
- New global preferences,
- Management of printing in a single print job.

New command

The “DR Area Control” theme now has a new command that can be used to find out the location of the mouse cursor.

DR GET MOUSE

DR GET MOUSE(area; horizontal; vertical)

Parameters	Type	Description
area	Longint	→ 4D Draw area
horizontal	Number	← Horizontal position of mouse cursor
vertical	Number	← Vertical position of mouse cursor

The DR GET MOUSE command returns the current location of the mouse cursor in the *horizontal* and *vertical* variables. This position is calculated with respect to the origin of *area* and is expressed in base units (inches, points, centimeters). The DR Base to scale command can be used to convert the base units to scale units.

The command returns the location of the mouse cursor, regardless of whether or not the mouse button is pressed. During a dragging or resizing operation, DR GET MOUSE returns the position of the mouse when its button is released.

If the command has been correctly executed, the system variable OK is set to 1; otherwise, it is set to 0.

- ▼ This example describes the installation of a process method displaying the coordinates of the mouse in a separate window when it moves over the 4D Draw area.

The *zDrInfoGetMouse* method launches the process which returns the mouse position:

```
`zDrInfoGetMouse method  
C_LONGINT($NewProc)  
$NewProc:=New process("zDrGetMouse";256;"MyTest";Drawarea)
```

The *zDrGetMouse* method, called using the *\$NewProc* process, displays the X and Y coordinates of the mouse in a window until the user presses the **Shift** key:

```
`zDrGetMouse method  
C_REAL($MouseX;$MouseY)  
$ref:=Open window(50;80;300;125;-Palette window;"Mouse position";  
"CloseBox")  
  
GOTO XY(1;0)  
MESSAGE("Press Shift to exit process")  
GOTO XY(1;1)  
MESSAGE("Values in X / Y")  
Repeat  
  DR GET MOUSE ($1;$MouseX;$MouseY)  
  GOTO XY(1;2)  
  MESSAGE(String($MouseX)+" / "+String($MouseY)+"")  
Until (Shift down)  
CLOSE WINDOW
```

New global preferences

In previous versions of 4th Dimension, the 4D Customizer Plus utility could be used to customize various options of 4D Draw (selection mode, lock keys, etc.). Since this utility was removed from the 4th Dimension 2004 product line, the main settings can now be accessed using the following 4D Draw commands:

- DR SET GLOBAL PREFERENCES and DR GET GLOBAL PREFERENCES (“DR Area Options” theme)
- DR SET ENTERABLE (“DR Area Control” theme).

DR SET GLOBAL PREFERENCES

DR SET GLOBAL PREFERENCES(write; read{; selection{; grid{; horLock{; vertLock}}}))

Parameters	Type	Description
write	Integer	→ Location to which to write
read	Integer	→ Location from which to read
selection	Integer	→ Object selection: 0=No change, 1=Included objects, 2=Touched objects
grid	Integer	→ Default activation of grid: 0=No change, 1=Grid disabled, 2=Grid enabled
horLock	String	→ Horizontal lock key
vertLock	String	→ Vertical lock key

The DR SET GLOBAL PREFERENCES command accepts four additional parameters that are used to access new global preferences:

- **selection:** This parameter modifies the way in which objects can be selected in 4D Draw areas.
 - If you pass 1, only objects that are entirely included within the selection rectangle will be selected (default behavior).
 - If you pass 2, all the 4D Draw objects touched by the selection rectangle will be selected. This behavior is identical to that of the Form editor in 4th Dimension.
 If you do not want to modify this setting, pass 0.
- **grid:** This parameter is used to set the initial state of the grid for all the 4D Draw documents that are opened or created.
 - If you pass 1, the grid is disabled (default behavior).
 - If you pass 2, the grid is enabled.
 If you do not want to modify this setting, pass 0.
- **horLock and vertLock:** These parameters let you designate the horizontal and vertical lock keys that can be used when creating objects. Pressing one of these keys while you are drawing an object locks the object size associated with the key.

By default, the *h* and *v* keys are used respectively for horizontal and vertical locking.

See also: DR GET GLOBAL PREFERENCES

DR GET GLOBAL PREFERENCES

DR GET GLOBAL PREFERENCES(write; read{; selection{; grid{; horLock{; vertLock}}}))

Parameters	Type	Description
write	Integer	← Location to which to write
read	Integer	← Location from which to read
selection	Integer	← Object selection: 1=Included objects, 2=Touched objects
grid	Integer	← Default activation of grid: 1=Grid disabled, 2=Grid enabled
horLock	String	← Horizontal lock key
vertLock	String	← Vertical lock key

The DR GET GLOBAL PREFERENCES command returns four additional parameters that are used to find out the current value of new global preferences.

For more information about these new options, refer to the description of the [DR SET GLOBAL PREFERENCES command, page 33](#).

See also: [DR SET GLOBAL PREFERENCES](#)

DR SET ENTERABLE

DR SET ENTERABLE(area; mode{; buttonMode})

Parameters	Type	Description
area	Integer	→ 4D Draw area
mode	Integer	→ Locking of area
buttonMode	Integer	→ Display as button when area is small: 0 = Yes, 1 = No

The DR SET ENTERABLE command accepts a new parameter that is used to specify whether the 4D Draw *area* can be displayed as a button when its size is too small in the form.

- If you pass 0 in *buttonMode*, the area is displayed as a button if its height is less than 150 points or if its width is less than 300 points. When the user clicks on this button, 4D Draw switches to full page view. This function is enabled by default.
- If you pass 1 in *buttonMode*, the area is never displayed as a button.

Management of printing in a single print job

It is now possible to carry out printing in a single print job with 4D Draw 2004.

By default, when successive calls to the DR PRINT command were made, pages could be inserted between different print jobs. The DR EXPORT MODE command now accepts a new value that is used to enable the “print one job” mode.

DR EXPORT MODE DR EXPORT MODE(area; mode)

Theme: DR Area Control.

The DR EXPORT MODE command can be used to enable “print one job” mode for the current session. When this mode is enabled, 4D Draw stores all the print requests in a temporary file on disk. The documents will only be sent to the printer when “print one job” mode has been disabled (using value 667). This lets you make sure that all the documents will be printed in a single print job.

- To enable “print one job” mode, pass 666 in the *mode* parameter,
- To disable this mode (default behavior), pass 667 in the *mode* parameter.
- ▼ Here is a typical example for enabling “print one job” mode:

```
$Area:=DR New offscreen area
DR EXPORT MODE($Area;666)
ALL RECORDS([MyTable3])
For($i;1;Records in selection([MyTable3]))
  DR AREA TO FIELD($Area;3;4)
  DR PRINT($Area;0)
  NEXT RECORD([MyTable3])
End for
DR EXPORT MODE($Area;667)
DR DELETE OFFSCREEN AREA($Area)
```


5

4D Pack 2004

Version 2004 of the 4D Pack plug-in includes three new commands:

- The AP Add table and fields command, located in the “4D_Pack: Utilities” theme, lets you add a table and fields by programming,
- The AP Print settings to BLOB and AP BLOB to print settings commands (“4D_Pack: Printing” theme) let you store and load all the print settings.

4D_Pack: Utilities

AP Add table and fields

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

Parameters	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)
Function result	Longint	← Number of the table created

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 2004 “new generation” 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.

If it has been correctly 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.

- ▼ This command 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}:=ls Text
```

```
tFieldLengths{1}:=0
```

...

```

$ret:= AP Add table and fields("Employees"; tFieldNames; tFieldTypes;
                                tFieldLengths)

```

4D_Pack: Printing

AP Print settings to BLOB

AP Print settings to BLOB (printSettings) → Longint

Parameters	Type	Description
printSettings	BLOB	→ BLOB which stores the print settings
Function result	Longint	← Error code: 1 = Operation successful 0 = No current printer

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

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

Parameters	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

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](#)

