

4D Write®

Version 6.5
Language Reference
Windows and Mac OS Versions



4D Write
by
ACI S.A.

4D Write Language Reference ***Version 6.5 for Windows® and Mac™ OS***

Copyright © 1985-1999 ACI SA/ACI US, Inc.
All rights reserved

The Software described in this manual is governed by the grant of license in the ACI 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 ACI Product Line License Agreement.

4D Write, 4th Dimension, 4D, the 4D logo, 4D Server, ACI, and the ACI logo are registered trademarks of ACI SA.

Microsoft and Windows are registered trademarks of Microsoft Corporation.

Apple, Macintosh, Mac, Power Macintosh, Laser Writer, Image Writer, ResEdit, and QuickTime are trademarks or registered trademarks of Apple Computer, Inc. All other referenced trade names are trademarks or registered trademarks of their respective holders.

IMPORTANT LICENSE INFORMATION

Use of this Software is subject to the ACI Product Line License Agreement, which is provided in electronic form with the Software. Please read the ACI Product Line License Agreement carefully before completely installing or using the Software.

Contents

1. 4D Write, Introduction to the language 11

Introduction	13
Multi-platform Document Management	14
Language Conventions in this Manual	16
Commands in the Method Editor	17
Documents in 4D Write Areas	18
4D Write menu items	21
Referring to Characters	22

2. WR Area Control 23

WR AUTO SCROLL	25
WR EXECUTE COMMAND	26
WR GET COMMAND INFO	27
WR Get doc property	29
WR LOCK COMMAND	30
WR ON COMMAND	31
WR REDRAW	32
WR SET DOC PROPERTY	33
WR UPDATE MODE	34

3. WR Area Options 35

WR Build preview	37
WR GET AREA PROPERTY	38
WR GET CURSOR POSITION	40
WR Get frame	41
WR SET AREA PROPERTY	42
WR SET CURSOR POSITION	44
WR SET FRAME	45
WR TEXT ACCESS	46

4. WR Areas 47

WR Area to blob	49
WR Area to picture	50
WR BLOB TO AREA	51
WR DELETE OFFSCREEN AREA	52
WR New offscreen area	53
WR PICTURE TO AREA	54

5. WR Database Objects 55

WR GET REFERENCE	57
WR INSERT EXPRESSION	59
WR INSERT FIELD	61
WR Insert picture area	62

6. WR Documents 63

WR GET DOCUMENT INFO	65
WR LOCK DOCUMENT	66
WR OPEN DOCUMENT	67
WR SAVE DOCUMENT	69
WR SET DOCUMENT INFO	71

7. WR Picture Control 73

WR DELETE PICTURE IN PAGE	75
WR GET PICTURE IN PAGE INFO	76
WR GET PICTURE SIZE	78
WR Get selected picture	79
WR Get styled text	80
WR INSERT PICTURE	81
WR SET PICTURE IN PAGE INFO	83
WR SET PICTURE SIZE	85

8. WR Printing _____ 87

WR INSERT PAGE NUMBER	89
WR PRINT	90
WR PRINT MERGE	91

9. WR Rulers _____ 93

WR ADD STYLESHEET TAB	95
WR ADD TAB	96
WR DELETE TAB	98
WR GET TAB	99
WR SET TAB	100

10. WR Style Sheet _____ 101

WR APPLY STYLESHEET	103
WR Create stylesheet	104
WR DELETE STYLESHEET	105
WR DELETE STYLESHEET TAB	106
WR Get stylesheet font	107
WR GET STYLESHEET INFO	108
WR GET STYLESHEET TAB	109
WR Get stylesheet text prop	110
WR SELECT PICTURE IN PAGE	111
WR SET STYLESHEET FONT	112
WR SET STYLESHEET INFO	113
WR SET STYLESHEET TAB	115
WR SET STYLESHEET TEXT PROP	117
WR UPDATE STYLESHEET	118

11. WR Text Manipulation _____ 119

WR BACKSPACE	121
WR DELETE SELECTION	122
WR Direct find	123

WR Find	124
WR Get font	125
WR GET PARAGRAPHS	126
WR Get selected text	127
WR GET SELECTION	128
WR Get text	129
WR Get text property	130
WR GET WORDS	132
WR INSERT STYLED TEXT	134
WR INSERT TEXT	135
WR Mouse to Selection	136
WR REPLACE	137
WR SELECT	138
WR SET FONT	140
WR SET SELECTION	141
WR SET TEXT PROPERTY	142

12. WR Utilities 143

WR COLOR TO RGB	145
WR Count	146
WR Error number	147
WR Error text	148
WR FONTS TO ARRAY	149
WR ON ERROR	150
WR ON EVENT	151
WR RGB to color	154

13. WR Obsolete Commands 157

WR O Area to picture	159
WR O AUTO SAVE	160
WR O CHANGE STYLE	161
WR O COMPUTE NOW	162
WR O Count Stylesheet	163
WR O CREATE STYLESHEET	164
WR O DELETE STYLESHEET	165
WR O DISPLAY MENUBAR	166

WR O DISPLAY RULER	167
WR O DISPLAY SCROLLBARS	168
WR O DO COMMAND	169
WR O EXPERT COMMAND	170
WR O Find	171
WR O Font name	172
WR O Font number	173
WR O GET ATTRIBUTES	174
WR O GET MARGINS	176
WR O Get pack options	177
WR O Get page	179
WR O GET PICTURE	180
WR O GET PREFERENCES	181
WR O GET RULER	182
WR O Get ScrollBars	184
WR O GET STYLESHEET	185
WR O GET TABS	187
WR O INSERT HYPHEN	189
WR O INSERT PICTURE	190
WR O Is Hyphen	191
WR O LINE SPACING	192
WR O MENU STATUS	193
WR O ModuleInfo	194
WR O MOVE PICTURE	195
WR O ON MENU	196
WR O OPTIONS	197
WR O Page number	199
WR O PICTURE TO AREA	200
WR O Picture to offscreen area	201
WR O REMOVE HYPHEN	202
WR O Replace	203
WR O RESIZE PICTURE	204
WR O Save to picture	205
WR O SET ATTRIBUTES	206
WR O SET MARGINS	208
WR O SET PACK OPTIONS	209
WR O SET PREFERENCES	211
WR O SET STYLESHEET	212
WR O SET TABS	214
WR O STATISTICS	215

WR O STRUCTURE ACCESS	216
WR O TEXT ALIGNMENT	217

14. Appendixes 219

Appendix A: Shortcuts	221
Appendix B: Menu Item Numbers	223
Appendix C: Error Codes	228
Appendix D: 4D Write Constants	230

Command Index 239

1 4D Write, Introduction to the language

4D Write is a plug-in that adds word processing commands and capabilities to 4th Dimension. With these commands, you can automate tasks typically done manually on a document, such as:

- Execute menu commands
- Open and save documents
- Set the margins of a document
- Set display attributes

All 4D Write commands added to 4th Dimension are preceded by the letters WR. This distinguishes these commands from those of 4th Dimension or any other plug-ins.

4D Write documentation

The documentation available for 4D Write consists of two manuals, 4D Write User reference and 4D Write Language Reference. The purpose of this manual (4D Write Language Reference) is to describe the use of the programming language of 4D Write. For more information about how to use 4D Write, please refer to the 4D Write User Reference manual.

Multi-platform Document Management 4D Write, Introduction to the language

version 6.5

4D Write, like 4th Dimension and 4D Server, is a multi-platform program. So, a database created under Mac OS and that uses 4D Write can be run under Windows with no modifications, and vice versa. This is possible only if you use the correct versions of the software.

File Equivalents on Mac OS and Windows

The following table indicates the file equivalents of 4D Write documents on Mac OS and Windows.

Document	Mac OS		Windows
	Type	Creator	Extension
4D Write 6.5 document	4WR7	4DW7	4W7
RTF	TEXT	4DW7	RTF
Windows Text only	TEXT	4DW7	TXT
Mac OS text only	TEXT	4DW7	TXT
Unicode text document	TEXT	4DW7	TXT
HTML document		4DW7	TXT

Documents

The following rules must be acknowledged:

- Under Mac OS, 4D Write uses the type and creator to recognize documents. For example, type 4WR7, creator 4DW7 = 4D Write document. The complete access path includes the disk name, folder names, and document name, each separated by a colon (:). For example, MyDisk:Folder1:Folder2:Mydatabase.
- Under Windows, 4D Write uses the file name extension to recognize documents. For example, .4W7 = 4D Write document. The complete access path includes the disk letter, directory names, and document name, each separated by a backslash (\). For example, D:\Directory1\Directory2\Mydatabase.

- A 4D Write document created under Mac OS and copied onto Windows can be opened directly, provided that it has been saved with its file name extension. For example, the MyDoc document saved as MyDoc.4W7, copied onto a PC volume, can be opened with no further handling.
- A 4D Write document created under Windows and copied onto Mac OS or Power Macintosh can be opened with no further handling.

Templates

To share templates between Mac OS and Windows clients, regardless of the server platform, the procedure is as follows.

The process is transparent for users. If the server is a Windows machine, the name of the template file should be AreaName_.4WT and if the server is a Macintosh machine the name of the template file should be AreaName_.

Description

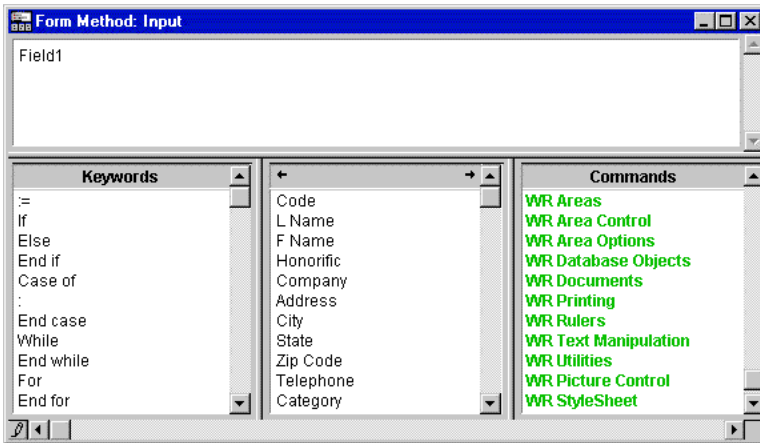
In this manual, 4D Write commands are printed in all uppercase letters using a special font, for example: **WR ON COMMAND**. 4D Write functions are shown with an initial capital letter, for example: **WR Get** styled text.

When 4D Write commands or functions appear in methods or object methods, they are displayed in a bold italic typeface to differentiate them from built-in 4th Dimension commands and functions. Non-italic bold text indicates 4th Dimension language terms.

```
QUERY([Templates];[Templates]ID=vNumber) ` 4th Dimension command  
If (Records in selection ([Templates])=1)  
    WR PICTURE TO AREA (Area;[Templates]Doc) ` 4D Write command  
End if
```

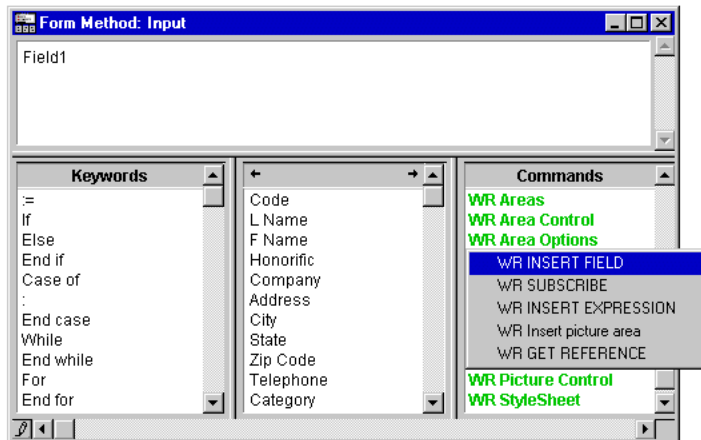
In some examples in this manual, a line of code may be continued on a second or third line due to space limitations. However, when you type these examples, keep those lines of code on a single line—do not press the Return key and cause a break in flow.

The 4D Write commands are grouped into “themes” in the Method editor. The themes are located at the end of the Routines list.



If you have more than one plug-in installed, the command themes appear in the order in which the plug-ins were installed.

You can place a 4D Write command in a method just as you do any 4th Dimension command. You can either type it directly into the method or choose the command from the pop-up menu in the Routines list.



You can use a 4D Write command in any type of method—global, table, form, or database. The commands are especially useful in object methods activated by objects on the same form as the document area.

There are three types of areas available to you in 4th Dimension:

- External areas in forms
- External windows
- Offscreen areas

To use a 4D Write document, you either create an external area on a form or open an external window. You create an external area by drawing the area on a form in the Design environment. You open an external window either by choosing 4D Write from the Windows menu in the User environment or by executing the External window function.

In addition to creating visible areas, you can create invisible offscreen areas. For more information, refer to 4D Write Offscreen Areas in this chapter.

4D Write Area ID Number and Variable

4D Write uses variables to store the location of 4D Write areas, external windows, and offscreen areas. You reference the area on which you want to perform an operation by passing the variable containing the area's ID number as a parameter to the command or function.

In the command descriptions that follow this introduction, the Area parameter refers to the variable identifying the document area.

There are two types of Area variables:

- External object names

When you create and name a 4D Write area, 4th Dimension automatically recognizes the name of the 4D Write area as a variable referring to the area. For example, you would refer to the Letter area by specifying "Letter" as for the Area parameter.

- Variables you create for an external window or offscreen area

When you create an external window or offscreen area using the External window or WR New offscreen area functions, you can store the area ID number returned by the function in a variable. You can then use the variable to refer to the external window or offscreen area in other commands and functions. To store the value in a variable, you place the variable name and the assignment operator (:=) to the left of the function in the line of code.

Most 4D Write commands require you to specify an area before they can be executed.

4D Write External Areas

When you want a 4D Write document to appear in a 4th Dimension form, you must create an external area on the form and assign it a unique name, specifying the external area type as 4D Write.

4th Dimension allows you to save this document with the record.

You will probably most often use the external area to store a document or to use it instead of a text field if formatting is important.

4D Write External Window Areas

4th Dimension allows you to create a 4D Write document in an independent area called an external window. External windows are useful when you want the user to have access to a word processor at any time to write letters, memos or other documents.

Issuing the 4th Dimension function, External window, from a method opens a specified window and returns an area ID in a long integer variable. You can reference this variable whenever you want to issue a 4D Write command to affect the external window.

For example:

```
vWrite:=Open external window (50; 50; 350; 450; 8; "Merge Letter"; "_4D Write" )
```

For more information about the Create external window command, please refer to its definition in the 4D Language Reference manual.

4D Write Offscreen Areas

An offscreen area is stored in memory and is not visible to the programmer or user. You can use an offscreen area to modify a document before a user views it or to save the document so a user can revert to the original, if necessary.

WR New offscreen area and WR Picture to offscreen area are the two functions used to create an offscreen area. Refer to Chapter 9 for a complete explanation of these Area functions. Remember to delete the offscreen area after you are done with it to free the memory it uses.

When placed in a global method, the following code creates an offscreen area for saving the document.

```
QUERY([Employee];[Employee]ID=vID)
If (Records in selection([Employee]=1)
Area:=WR New offscreen area
$res:=WR Picture to area(Area;[Employee]Review_)
`Store the review in the offscreen area
MODIFY RECORD([Employee])
`Modify the employees record
WR DELETE OFFSCREEN AREA(Area)
`Free the memory used by the offscreen area
End if
```

Using a button on a form, you can allow a user to revert to the original saved document.

You can create a button on the input form and assign it the following code:

```
Review:=WR Area to picture(Area)
`Places the offscreen area that contains the original document into the external
`area contained in the Review form.
```

You can procedurally gain access to a 4D Write menu and select a menu item. In a method, you can determine the status of a menu or menu item. Each menu item is referenced by a unique integer. See Appendix B for a listing of menu item integers.

The menu item integers are generally based on the location of the menu and menu item. The menus are numbered from left to right in ascending order. For example, File = 100 and Edit = 200. Likewise, menu items are numbered in ascending order from top to bottom. Therefore, the New menu item is numbered 101, because it is the first item on the first menu, File.

The numbers for these menu items always remain the same, even in future versions of 4D Write which may have new menu items. Any new menu items will use different numbers, even if placed between current menu items. This placement will invalidate the general rule of numbering menu items, but the menu references you use in methods will remain accurate, so you will not need to update them.

A character in a document is referred to by its sequential number. Commands that refer to characters enable you specify either a single character or a range of characters. For example, you can specify a word, a sentence, or whole blocks of text to be selected.

You use the `WR GET SELECTION` command to determine the positions of selected characters in a 4D Write area. The command uses the `First` and `Last` parameters to refer to the range of selected characters. The `First` parameter is always one less than the first character selected. The `Last` parameter is equal to the last character selected.

Example

For example, the following expression returns into the variables `$First` and `$Last` the positions of the selected text in `Area`:

```
WR GET SELECTION (Area; $First; $Last)
```

To select text in a 4D Write area, you need to reference characters. In most cases, you must first select text before using a command to manipulate it. For more information, refer to the “Text Manipulation” section.

2 WR Area Control

WR AUTO SCROLL

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

This command does not require any parameters

Description

The command WR AUTO SCROLL scrolls Area until the selected text is visible. This command is useful when modifications are made through 4D Write commands and the user needs to view the resulting changes.

See Also

No reference.

WR EXECUTE COMMAND (area; cmdNumber)

Parameter	Type		Description
area	Longint	→	4D Write area
cmdNumber	Longint	→	Number of the command to execute

Description

The WR EXECUTE COMMAND causes the action associated with a 4D Write menu command or toolbar button to be executed. The most common use for this command is to execute a command after the user has chosen that command and your code intercepted the user's choice through the WR ON COMMAND command.

Note : The list of commands and their references are available in Appendix B. You can either pass a constant or a value.

See also

WR ON COMMAND.

WR GET COMMAND INFO (area; commandNumber; numValue; stringValue; name; status)

Parameter	Type		Description
area	Longint	→	4D Write area
commandNumber	Longint	→	Number of the command to process
numValue	Longint	←	Command status
stringValue	String	←	Selected text value
name	String	←	Command name or text of the Tip
status	Integer	←	0=disabled 1=enabled

Description

The WR GET COMMAND INFO command allows you to get the status of the menu or toolbar command whose number is passed in commandNumber.

Note: The list of commands and their references is available in appendix D (theme WR Commands). You can either pass a value or a constant.

numValue returns the status of the command. numValue will equal to 0 if the command is dimmed or 1 if it is enabled or 2 if the command is partly enabled (for instance the Bold command when text is selected some of which is in bold face and some is not).

stringValue contains a text that varies and is specific to each command.

name contains the name of the command. This is either the text of the menu command or the text of the tip displayed for that command.

status returns the status of the menu command. status will equal to 0 if the menu command is dimmed and 1 if it is enabled.

Example

The following example applies to the Bold command (Constant: wr cmd gras , Value : 502)

⇒ *WR GET COMMAND INFO*(area;wr cmd bold;numValue;stringValue;name;status)

numValue=1 if text is in bold

numValue=0 if text is not in bold

numValue=2 if only part of the selection is in bold

The following example applies to the Font drop-down list (Constant: wr cmd font dropdown , Value : 1002)

⇒ *WR GET COMMAND INFO*(area;wr cmd font dropdown;numValue;stringValue;name;status)

stringValue="Arial"

See also

No reference.

WR Get doc property (area; property) → value

Parameter	Type		Description
area	Longint	→	4D Write area
property	Integer	→	Number of the property to read
Function result	value	←	Value for the tested property

Description

The WR Get doc property command allows you to get the attributes of the document currently opened in the 4D Write area referenced by area.

- property

For some properties, value returns 1 (True) or 0 (False). An example is property 2 ([wr view ruler](#)).

For other properties, value returns a number expressed in the current default unit. An example is property 21 ([wr paper width](#)).

Note : property can be set using constants.

The list of document properties is available in appendix D. You can either pass the constant or the value.

See also

No reference.

WR LOCK COMMAND (area; cmdNumber; locked)

Parameter	Type		Description
area	Longint	→	4D Write area
cmdNumber	Longint	→	Number of the command to process
locked	Integer	→	0=enables the execution 1=does not enable the execution

Description

The WR LOCK COMMAND command allows you to prevent the user from being able to execute the command whose number is passed in cmdNumber. Those can either be a menu command or a palette command. This command affects the user's access to the indicated command only in the 4D Write area referenced by area. Access to the command is unaffected in other 4D Write areas.

Notes:

- Even if a command is locked, your code can still execute it using the WR EXECUTE COMMAND command.
- WR ON COMMAND will not be called if the user tries to select a command that is disabled.

- locked

If locked equals 1, the command will not be disabled.

Si locked equals 0, the command will be enabled.

Note : The list of constants is available in appendix D.

See also

No reference.

WR ON COMMAND (area; 4DMethod)

Parameter	Type		Description
area	Longint	→	4D Write area
4DMethod	Alpha	→	Replacement method

Description

The WR ON COMMAND command executes the method passed as 4DMethod when a 4D Write command is invoked by the user, by the selection of a menu command or by a click on a button. If area equals zero, 4DMethod will apply to each 4D Write area until the database is closed or until the following call to WR ON COMMAND is made: WR ON COMMAND(0; "").

4DMethod receives two parameters:

- \$1
\$1 is Longint that represents area.
- \$2
\$2 is a Longint that designates the command number.

When planning to use a compiled database, it is necessary to declare both \$1 and \$2 as Longints, even if you do not use them.

If you want the initial command to be executed you need to include the following in the called method: WR ON COMMAND(\$1;\$2).

WR REDRAW (area)

Parameter	Type		Description
area	Longint	→	4D Write area

Description

The command *WR REDRAW* causes Area to be redrawn. This command is useful when you have disabled screen updating with the *WR UPDATE MODE* command and now want to redraw a 4D Write area to show how previously executed code has modified the area.

Example

The following example turns off screen updates, calls the *Reformat* global method that reformats area, and then redraws area without turning screen updating back on.

```
WR UPDATE MODE (area;0)  
  `Turn off screen updating  
Reformat (area)  
  `area can be passed to a method  
WR REDRAW (area)  
  `Redraw to display changes
```

See Also

No reference.

WR SET DOC PROPERTY (area; property; value)

Parameter	Type		Description
area	Longint	→	4D Write area
property	Integer	→	Number of the property to set
value	Number	→	Value for the selected property

Description

The WR SET DOC PROPERTY command allows you to modify the document attributes in the 4D Write area referenced by area.

- The meaning given to the value parameter depends on the property value used. If the value for property is constant property wr view ruler or 2, values for value can either be 1 (displayed) or 0 (hidden). If the value for property is constant property wr column spacing or 25, value will be a number expressed in the current default unit.

Note: property and value can be set using constants.

The list of text properties and their references are available in the Appendix D, in the WR Text properties theme.

The four following properties are only available in read-only mode, you cannot set them using the WR SET DOC PROPERTY command.

Constant	Values
wr dead left margin	39
wr dead top margin	40
wr printable width	41
wr printable height	42

Be aware that if you set the paper size in code, 4D Write will set the dead margins to 0 (paper size that was set).

This allows you to set document margins without having to deal with dead margins. This is useful when your documents are not destined to be printed (for example if they are used to create pictures).

See also

No reference.

WR UPDATE MODE (area; mode)

Parameter	Type		Description
area	Longint	→	4D Write area
mode	Integer	→	0= No update 1= Update

Description

The command WR UPDATE MODE allows the designer to enable and disable screen updating in area. If mode equals 0, screen updating is off. If mode equals 1, screen updating is on. This command only affects screen updates caused by 4D Write commands. User actions in Area will continue to update the screen correctly.

When screen updating is turned off, 4D Write commands execute faster. For example, if you intend to execute a series of modifications to a 4D Write area, turn off updating before beginning the modifications and then turn updating on when you are finished. The commands execute faster as well as the screen redraw.

Example

The following example turns off screen updating, calls the Reformat global method that makes several modifications, and then turns screen updating back on:

```
WR UPDATE MODE (area;0)
Reformat (Area)
WR UPDATE MODE (area;1)
```

See Also

No reference.

3 WR Area Options

WR Build preview (area; page) → Picture

Parameter	Type		Description
area	Longint	→	4D Write area
page	Longint	→	Number of the page to pass as a picture
Function result	Picture	←	Picture of the page

Description

The WR Build preview command converts the picture whose number is passed in page into a picture. The page number takes into account the page numbering as it was defined in the preferences dialog.

The picture can be stored, for instance, in a 4D picture field or in a 4D picture variable. The picture is the same size as the page. You can set the size of the picture by using the WR SET DOC PROPERTY command and by passing a value for wr paper width and wr paper height.

Note : unlike when you use WR Area to picture, the picture does not contains any 4D Write data

The returned picture is a vector-based picture. A picture that was created on Windows cannot be directly displayed on Mac OS since it uses the EMF format. On the other hand, the opposite is possible, since 4D can display Mac OS pictures. If you want your Windows pictures to be displayed on Mac OS, you need to convert the picture into a bitmap by using the following code:

```
myPicture:=myPicture|myPicture.
```

Unlike EMF (Windows only), Pict and bitmap picture types are not platform dependent.

See also

WR SET DOC PROPERTY.

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

Parameter	Type		Description
area	Longint	→	4D Write area
option	Integer	←	Option number
value	Integer	←	0 or 1 depending on the option
stringValue	String	←	Property string depending on the case

Description

The WR GET AREA PROPERTY command allows you to read various options for the 4D Write area referenced by area.

The options that can be read are the following :

N°	Options (constants are underlined)	Values	Alpha effect
0	Confirm dialog <u>wr_confirm_dialog</u>	0 - 1 -	No dialog Dialog
1	Picture preview <u>wr_save_preview</u>	0 - 1 -	No preview Preview
2	Saving Redo <u>wr_allow_undo</u>	0 - 1 -	No buffer Actions are stored
3	Dirty bit except if area = 0 <u>wr_modified</u>	0 - 1 -	False True
4	Variable size printing <u>wr_fixed_print_size</u> (unless area = 0)	0 - 1 -	Variable size Fixed size
5	Field conversion dialog (if area = 0) <u>wr_convert_dialog</u>	0 - 1 -	no dialog dialog
6	Button title when area is minimized <u>wr_minimized_button_title</u>	0 - 1 Title	Default title
7	4D Write Window title (when going to full screen or in external window) <u>wr_window_title</u>	0 - 1 Title	area name
8	Minimum area width before switching to button <u>wr_minimum_width</u>	XX -	in pixels
9	Minimum area height before switching to button <u>wr_minimum_height</u>	XX -	in pixels
10	Saving the templates on the server in C/S <u>wr_save_template_on_server</u>	0 - 1 -	on client on server
11	Loading templates from server in C/S <u>wr_load_template_on_server</u>	0 - 1 -	on client on server

The list of text properties and their references are available in the Appendix D, in the WR Options theme. You can either pass the value or the constant.

See also

No reference.

WR GET CURSOR POSITION (area; page; column; line; position)

Parameter	Type		Description
area	Longint	→	4D Write area
page	Longint	←	Number of the page where the selection is
column	Longint	←	Number of the column where the selection is
line	Longint	←	Number of the line in the column
position	Longint	←	Position of the selection in the current line

Description

The WR GET CURSOR POSITION command returns the position of the selection in the 4D Write area referenced by area.

- page

Page is between the number of the first page and the number of the last page of the document. These numbers take into account the custom page numbering, if any.

- column

This value is between 1 and the total number of columns.

- line

This value is between 1 and the total number of lines in the column.

- position

This value is between 1 and the total number characters in the line

If the selection contains several characters, the position of the first character is returned. You can later go back to this location, using the WR SET CURSOR POSITION command with the same parameters.

You can use WR Get frame to determine which area the cursor is in.

See also

WR Get frame, WR SET CURSOR POSITION.

WR Get frame (area) → Longint

Parameter	Type		Description
area	Longint	→	4D write area
Function result	Longint	←	Page area in which the cursor is

Description

The WR Get frame command returns a number which represents which page area the insertion point or the current selection is in.

The following values can be returned:

Value	Location	Constants
0	text area	<u>wr text frame</u>
1	right header	<u>wr right header</u>
2	right footer	<u>wr right footer</u>
3	left header	<u>wr left header</u>
4	left footer	<u>wr left footer</u>
5	first header	<u>wr first header</u>
6	first footer	<u>wr first footer</u>

You can enter these values by number or by using a predefined constant (as shown).

Note : The list of values is also available in appendix D.

See also

No reference.

WR SET AREA PROPERTY (area; option; value; alphaValue)

Parameter	Type		Description
area	Longint	→	4D Write area
option	Integer	→	Option number
value	Integer	→	0 or 1, depending on the option
alphaValue	String	→	String for the property, depending on the option

Description

The WR SET AREA PROPERTY command allows you to modify the properties of the 4D Write area referenced by area.

If area equals 0, the WR SET AREA PROPERTY command will apply to each active 4D Write area. In this case, it is recommended that your code should call this command in the On Startup database method.

The options that can be set are the following:

N°	Options (constants are underlined)	Values	Alpha effect
0	Confirm dialog <u>wr confirm dialog</u>	0 - 1 -	No dialog Dialog
1	Picture preview <u>wr save preview</u>	0 - 1 -	No preview Preview
2	Saving Redo <u>wr allow undo</u>	0 - 1 -	No buffer Actions are stored
3	Dirty bit except if area = 0 <u>wr modified</u>	0 - 1 -	False True
4	Variable size printing <u>wr fixed print size</u> (unless area = 0)	0 - 1 -	Variable size Fixed size
5	Field conversion dialog (if area = 0) <u>wr convert dialog</u>	0 - 1 -	no dialog dialog
6	Button title when area is minimized <u>wr minimized button title</u>	0 - 1 Title	Default title
7	4D Write Window title (when going to full screen or in external window) <u>wr window title</u>	0 - 1 Title	area name
8	Minimum area width before switching to button <u>wr minimum width</u>	XX -	in pixels
9	Minimum area height before switching to button <u>wr minimum height</u>	XX -	in pixels
10	Saving the templates on the server in C/S	0 -	on client

Note : option can be set using constants.

The list of text properties and their references are available in the Appendix D, in the WR Options theme.

See also

No reference.

WR SET CURSOR POSITION (area; page; column; line; position)

Parameter	Type		Description
area	Longint	→	4D Write area
page	Longint	→	Page number
column	Longint	→	Column number
line	Longint	→	Line number
position	Longint	→	Horizontal position of the cursor in the line

Description

The WR SET CURSOR POSITION command moves the insertion point to a new position specified by page, column , line and position .

- page

The value for page must be between the first and the last page numbers of the document. The page number must take into account the page numbering as it was defined in the preferences dialog.

- column

The value for column must be between 1 and the total number of columns.

- line

The value for line must be contained between 1 and the total number of lines of the column (or page, if there is only one column).

- position

La valeur doit être comprise entre 1 et le nombre de caractères dans la ligne. Pour placer le curseur en première position sur la ligne, mettez position à 1.

If you want to place the cursor in an area other than the body area, you need to use the WR SET FRAME command before using the WR SET CURSOR command.

See also

No reference.

WR SET FRAME (area; frame)

Parameter	Type		Description
area	Longint	→	4D Write area
frame	Integer	→	Frame number

Description

The WR SET FRAME command places the insertion point at its previous location in the part of the 4D Write area area indicated by the frame parameter. This position was previously memorized by 4D Write. If the Normal view mode is selected and the insertion point is placed in an header or footer area, 4D Write automatically switches to Page view mode.

You can pass in frame the following values or constants:

Value	Constants
0	<u>wr text frame</u>
1	<u>wr right header</u>
2	<u>wr right footer</u>
3	<u>wr left header</u>
4	<u>wr left footer</u>
5	<u>wr first header</u>
6	<u>wr first footer</u>

Values 3 and 4 are to be used when you use different headers and footers for left and right pages.

Values 5 and 6 are to be used when you use different headers and footers for the first page.

Note : The list of values is also available in appendix D.

See also

No reference.

WR TEXT ACCESS (area; mode)

Parameter	Type		Description
area	Longint	→	4D Write area
mode	Integer	→	0=Allow access 1=Restrict access

Description

The command WR TEXT ACCESS enables you to control access to the text in Area. If mode equals 0, 4D Write allows modifications to area. If mode equals 1, 4D Write displays area in read-only mode.

When an area is displayed in read-only mode, the menus, rulers, and Zoom box are not present. The text can be seen and scrolled but not modified. When access to a formerly restricted area is changed, you must call WR DISPLAY RULER and WR DISPLAY MENUBAR to display the ruler and menu bar.

Example

The following example is the form method of the form that contains area. It sets area to read-only when the form is loaded.

```
If (Form event=On load)
  WR TEXT ACCESS (area;1)
End if
```

See Also

No reference.

4 WR Areas

WR Area to blob (area{; savedDoc}) → BLOB

Parameter	Type		Description
area	Longint	→	4D Write area
savedDoc	Integer	→	1=if document is not saved, no dialog 0=if document is not saved, the dialog is displayed
Function result	BLOB	←	Contents of area

Description

The WR Area to blob command places the contents of the area referenced by area into a BLOB field or variable. WR Area to blob returns a Blob that can be assigned to a BLOB field or a BLOB variable.

savedDoc

- If savedDoc equals 0, and the document has been modified since it was last saved, a dialog will be displayed asking the user if they wish to save the document.
- If savedDoc equals 1, the document will be considered as saved and the user will not be prompted to save it.

If savedDoc is omitted, default settings will be applied.

See also

No reference.

WR Area to picture (area; savedDoc; preview) → Picture

Parameter	Type		Description
area	Longint	→	4D Write area
savedDoc	Integer	→	1=if document is not saved, no dialog 0=if document is not saved, the dialog is
displayed preview	Integer	→	1= the picture is created 0=the picture is not created
Function result	Picture	←	Picture of the contents of area

Description

The WR Area to picture command allows you to place the contents of the areareferenced by area in a picture field or variable. Passing a 4D Write area to the WR Area to picture command returns a picture that can later be assigned to a picture field or a picture variable.

savedDoc

- If savedDoc equals 0, and the document has been modified since it was last saved, a dialog will be displayed asking the user if they wish to save the document.
- If savedDoc equals 1, the document will be considered as saved and the user will not be prompted to save it.

preview

- If preview equals 0, no picture preview will be created.
- If preview equals 1, a picture preview will be created.

Note: If no picture preview is created, the picture cannot be displayed.

If optional parameters are omitted, the default settings for area will be applied.

See also

No reference.

WR BLOB TO AREA (area; blob)

Parameter	Type		Description
area	Longint	→	4D Write area
blob	Blob	→	Variable or field that contains 4D Write data

Description

The WR BLOB TO AREA command loads into the 4D Write area area the contents of blob. The contents of the BLOB are assumed to be 4D Write data.

The contents of the Blob can either be data that was automatically saved from a 4D Write area associated by name with a BLOB, or data that was saved using the WR Area to blob command.

WR DELETE OFFSCREEN AREA (area)

Parameter	Type		Description
area	Longint	→	4D Write area

Description

The command **WR DELETE OFFSCREEN AREA** deletes a 4D Write area that was created with **WR New offscreen area** or **WR Picture to offscreen area** and frees the memory used by the offscreen area.

area must be an offscreen area and not an area on a form or in a window. Issue the **WR DELETE OFFSCREEN AREA** command when you no longer need the offscreen area.

Example

The following example illustrates the need to pair every call to **WR New offscreen area** with a corresponding call to **WR DELETE OFFSCREEN AREA**.

```
NewArea=WR New offscreen area
  `Create a new offscreen area
  `Do Something
WR DELETE OFFSCREEN AREA (NewArea)
  `Remove the offscreen area
```

See Also

No reference.

WR New offscreen area → Longint

Parameter	Type		Description
This command does not require any parameters			
Function result	Longint	←	Reference of 4D Write area

Description

The command **WR New offscreen area** reserves space in memory for a 4D Write area that is invisible to you and the user. This function also returns a value that can be used to access the invisible area. The value returned by **WR New offscreen area** can be used in any 4D Write command that requires a 4D Write area. Remember to delete the offscreen area created by this function when you are finished with it.

Example

The following example creates a temporary offscreen area, prints it and the deletes it.

```
⇒ Temporary:=WR New offscreen area
   WR INSERT TEXT(Temporary;MyText)
   WR PRINT(Temporary;0)
   WR DELETE OFFSCREEN AREA(Temporary)
```

See Also

No reference.

WR PICTURE TO AREA (area; picture)

Parameter	Type		Description
area	Longint	→	4D Write area
picture	Picture	→	Field or variable

Description

The WR PICTURE TO AREA command allows you to read a picture variable or a picture field that contains a 4D Write document and to open it in the 4D Write area referenced by area. area can either be an area currently displayed or an offscreen area.

This command allows you, for instance, to read 4D Write documents that were saved in different tables.

Note: This command also reads the 4D Write version 6.0.x file format.

See also

No reference.

5 WR Database Objects

WR GET REFERENCE (area; table; field; name; type)

Parameter	Type		Description
area	Longint	→	4D Write area
table	Integer	←	Receives table number
field	Integer	←	Receives field number
name	String	←	Receives reference name
type	Integer	←	Receives reference type

Description

The command WR GET REFERENCE gets information about the selected reference in the 4D Write area.

Information about the selected reference is returned in the table, field, name, and type parameters. table contains the table number. field contains the field number if the reference is a field. name contains the text of an expression or the name of the hot link if the reference is an expression or a hot link. Values returned in table, field, and name depend on the value in type. If the selected object is not a reference, type returns 0.

Type	Description
1	The reference is a field. <i>Table</i> indicates the Table number. <i>Field</i> indicates the field number. <i>Name</i> is empty.
2	The reference is an expression. <i>Table</i> and <i>Field</i> contain the value 0. <i>Name</i> contains the name of the variable or expression.
3	The reference is a hot link. <i>Table</i> and <i>Field</i> contain the value 0. <i>Name</i> contains the name of the hot link.

Example

This example determines if the user selected an object that is a reference. It also tells the user if the selected object is a field, hot link, or an expression.

```
WR GET REFERENCE (Letter;$Table;$Field;$Name;$Type)
Case of
: ($Type=0)           `Text or nothing
  ALERT("selected text or nothing")
: ($Type=1)
  ALERT("Selected the field "+Fieldname($Table;$Field))
: ($Type=2)
  ALERT("Selected the expression named "+$Name)
```

```
: ($Type=3)
  ALERT("Selected the hot link named "+$Name)
End case
```

WR INSERT EXPRESSION (area; expression)

Parameter	Type		Description
area	Longint	→	4D Write area
expression	String	→	Expression to insert

Description

The command WR INSERT EXPRESSION inserts a reference to Expression into area, replacing any currently selected text.

Expression must be a valid 4th Dimension expression that returns a value. expression can be a 4th Dimension variable, function, or statement that returns a value. expression can also be an external function or a user-defined function (global method).

If expression returns a value that includes carriage returns and tabs, 4D Write formats the text according to the ruler of the paragraph in which expression resides.

The following two-part example shows a reference to a 4th Dimension global method inserted into a 4D Write area. The global method finds a customer's related invoices and concatenates the invoice numbers and amounts.

Example

The following two-part example shows a reference to a 4th Dimension global method inserted into a 4D Write area. The global method finds a customer's related invoices and concatenates the invoice numbers and amounts.

```

`Global method SHOW INVOICES
$Tab:=Char (9)
$CR:=Char (13)
RELATE MANY ([Customers])
FIRST RECORD ([Invoices])
$0:=""
For ($i;1;Records in selection ([Invoices]))
    $0:=$0+[Invoices]Number+$Tab+String([Invoices]Amount;"###,##0.00")
    +$CR           `This fragment continues the line above.
NEXT RECORD ([Invoices])
End for

```

The second part of this example shows the insertion of the SHOW INVOICES global method into Area. When 4D Write displays or prints Area, each invoice will appear in a separate line.

WR INSERT EXPRESSION (area;"SHOW INVOICES")

See Also

No reference.

WR INSERT FIELD (area; table; field)

Parameter	Type		Description
area	Longint	→	4D Write area
table	Integer	→	Table number
field	Integer	→	Field number

Description

The command **WR INSERT FIELD** inserts a reference to a field into area, replacing any selected text. The field is described by the table and field numbers.

Unlike **WR INSERT EXPRESSION**, **WR INSERT FIELD** inserts a dynamic reference. Changes to the field are immediately reflected in the document.

See Also

No reference.

WR Insert picture area (area; picture; where) → Longint

Parameter	Type		Description
area	Longint	→	4D Write area
picture	Picture	→	4D Write area picture to insert
where	Integer	→	1=Document end 0=Insertion point
Function result	Longint	←	Error code

Description

The command WR Insert picture area inserts into area the 4D Write document in Picture.

where describes the position at which the new text will be inserted.

If where equals 1, the text will be inserted at the end of the document.

If where equals 0, the text will be inserted at the current insertion point or will replace any currently selected text.

WR Insert picture area returns a long integer containing an error code.

If the insertion is successful, the value returned is 0. See Appendix C for error codes.

Example

The following example adds the signature of the sender to the end of the document:

```
QUERY([Sender]; [Sender]Name=[Letter]Sender)
ErrorNum:=WR Insert picture area(area; [Sender]Signature_; 1)
```

6 WR Documents

WR GET DOCUMENT INFO (area; string; subject; author; company; notes; creationDate; creationTime; modifDate; modifTime; lock)

Parameter	Type		Description
area	Longint	→	4D Write area
string	String	←	Title of the document
subject	String	←	Subject of the document
author	String	←	Author of the document
company	String	←	Company name
notes	String	←	Document notes
creationDate	Date	←	Creation date
creationTime	Time	←	Creation time
modifDate	Date	←	Last modification date
modifTime	Time	←	Last modification time
lock	Integer	←	0=unlocked 1=locked

Description

The WR GET DOCUMENT INFO command allows you to retrieve document information as displayed in the Document information dialog. The Document information dialog is displayed by selecting Document information from the Tools menu.

Some of this information such as the document subject, the author's name, the company name and the notes can be set using the WR SET DOCUMENT INFO command.

lock can be set using the WR LOCK DOCUMENT command. It is a logical lock that prevents the user from modifying the document. It affects user operations such as Paste, Cut, text entry, modify or replace attributes. The user can still browse the document, copy text, perform some character searches or print the document.

creationDate, creationTime, modifDate, modifTime are automatically updated by 4D Write when the document is saved.

See also

No reference.

WR LOCK DOCUMENT (area; status)

Parameter	Type		Description
area	Longint	→	4D Write area
status	Integer	→	0=unlocked 1=locked

Description

The WR LOCK DOCUMENT prevents users from modifying the 4D Write area referenced by area.. Once the document is locked, users cannot paste text, cut text, enter or modify text. Scrolling, copying, searching and printing the document are still possible.

To determine the lock status of the current document, you can use the WR GET DOCUMENT INFO command. This information is also displayed in the Document information dialog. You can access that dialog by selecting Document information from the Tools menu.

status

- If status equals 1, the document will be locked
- If status equals 0, the document will be unlocked

See also

No reference.

WR OPEN DOCUMENT (area; document{; type})

Parameter	Type		Description
area	Longint	→	4D Write area
document	String	→	Name of the document to open
		←	
type	String	→	Type of the document to open (4 characters)
		←	Type of the open document (4 characters)

Description

The WR OPEN DOCUMENT opens the document specified by document and places it in the 4D Write area referenced by area.

document is the complete access path to the document file. On Windows in addition to passing the access path you must include the file extension.

Examples:

- on Windows, you must include the “\” character between directories: “D:\directory1\directory2\file.4W7”).
- on Mac OS, you must include the “:” character between folders: "MacintoshHD:Folder:Document".

If document is an empty string, WR OPEN DOCUMENT displays the standard Open file dialog.

When the Open button of the Open file dialog is clicked the OK system variable is set to 1, and the Document variable will be assigned the complete access path of the file the user selects.

In this case the type parameter returns the type selected by the user in the type drop-down list or the document type if no type was selected by the user.

If the user clicks the Cancel button, document returns an empty string and the OK system variable is set to 0.

The file formats supported by this command are:

Type	Files
4WR7	4D Write 6.5
4WR6	4D Write version 6
4WT7	Template 4D Write 6.5
RTF	rtf file
ASCW	Windows text file
ASCM	Mac OS text file
ASCU	Unicode text file
HTML	HTML text file

Example

The following example opens a file located in the database's directory.

- ⇒ *WR OPEN DOCUMENT*(area;"HD:Folder:database folder:File") 'On Mac OS
- ⇒ *WR OPEN DOCUMENT*(area;"D:\directory\Basedirectory\file.4W7") 'On Windows

See also

WR SAVE DOCUMENT.

WR SAVE DOCUMENT (area; document; type)

Parameter	Type		Description
area	Longint	→	4D Write area
document	String	→	Name of the document to create
type	String	→	Type of document

Description

The WR SAVE DOCUMENT saves the document located in the 4D Write area referenced by area, using the access path passed in document.

document is the complete access path to the document file. On Windows, in addition to passing the access path you must include the file extension, in order to determine the file type.

Examples:

- on Windows, you must include the “\” character between directories: “D:\directory1\directory2\file.4W7”).
- on Mac OS, you must include the “:” character between folders: "MacintoshHD:Folder:Document".

If document is an empty string, WR SAVE DOCUMENT displays the standard Save file dialog.

When the Save (Mac OS) or OK (Windows) button of the Save file dialog is clicked the OK system variable is set to 1, and the document variable will be assigned the complete access path of the file the user selects.

In this case the type parameter returns the type selected by the user in the type drop-down list or the document type if no type was selected by the user.

If the user clicks the Cancel button, document returns an empty string and the OK system variable is set to 0.

File formats can be selected from the Type drop-down list (on Windows) or from the type pop-up menu in the Save file dialog.

The file format available are:

Type	Files
4WR7	4D Write 6.5
4WR6	4D Write version 6
4WT7	Template 4D Write 6.5
RTF	rtf file
ASCW	Windows text file
ASCM	Mac OS text file
ASCU	Unicode text file
ASCN	Mac OS text file on Mac OS and Windows text file on Windows

Note: These types are virtual types, they neither correspond to Mac OS file types , nor to windows extensions.

See also

WR OPEN DOCUMENT.

WR SET DOCUMENT INFO (area; title; subject; author; company; comment)

Parameter	Type		Description
area	Longint	→	4D Write area
title	Alpha	→	Title of the document
subject	Alpha	→	Document subject
author	Alpha	→	Author of the document
company	Alpha	→	Company name
comment	Text	→	Comment

Description

The WR SET DOCUMENT INFO stores in the document the information that is passed in the parameters. From a user standpoint, the information is displayed in the Document information dialog. You can access that dialog by selecting Document information in the Tools menu.

To manage the document lock status, refer to WR LOCK DOCUMENT.

See also

No reference.

7 WR Picture Control

WR DELETE PICTURE IN PAGE (area; pictureNumber)

Parameter	Type		Description
area	Longint	→	4D Write area
pictureNumber	Longint	→	Picture number

Description

The WR DELETE PICTURE IN PAGE command deletes the picture whose number is passed in pictureNumber from the 4D Write area referenced by area. . For the WR DELETE PICTURE IN PAGE command to operate properly, the picture must be located in the page, rather than in the text stream. To delete a picture in the text stream, select it and call WR DELETE SELECTION.

You can retrieve a tyhe number of pictures in an area by using, WR count(area;13). When deleting a picture, 1 is substracted from each of the following picture numbers. You can also retrieve the picture number using the WR Get selected picture command.

Example

(1) The following example deletes all the pictures located in the page for the specified area.

```

$NbOccurrence :=WR Count (area;13)
For ($i;1;$NbOccurrence)
  It is always the first picture that is deleted
  WR DELETE PICTURE IN PAGE (area;1)
End for

```

See also

No reference.

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

Parameter	Type		Description
area	Longint	→	4D Write area
pictureNumber	Longint	→	Picture number
page	Longint	←	Picture location
behind	Integer	←	0= picture is in front of the text, 1= Picture is behind the text
firstPage	Integer	←	1= the picture is not in the first page, 0=the picture appears in all the pages
horizPos	Number	←	Horizontal position in the page
verticalPos	Number	←	Vertical position in the page
width	Number	←	Current width of the picture
height	Number	←	Current height of the picture
origWidth	Number	←	Original width of the picture
origHeight	Number	←	Original height of the picture

Description

The WR GET PICTURE IN PAGE INFO returns information about the picture whose number was passed in pictureNumber, as it currently appears in the 4D Write area referenced by area.

Warning: this command should not be used with pictures that are part of the text flow.

- page allows you to know in which page the picture is displayed.
If page is greater than -1, the picture is displayed in the page whose number was returned. This value takes into account the page numbering as it is currently defined.
If page is equal to -1, the picture is displayed in all the pages
If page is equal to -2, the picture is displayed in all the right pages
If page is equal to -3, the picture is displayed in all the left pages

- behind
If behind is equal to 0, the picture is in front of the text.
If behind is equal to 1, the picture is behind the text.

- firstPage
If firstPage is equal to 0, the picture is displayed on all pages.
If firstPage is equal to 1, the picture is displayed on all pages except the first page.

horizPos and vertPos return the coordinates of the picture's upper left corner in relation

`origWidth` and `origHeight` return the original dimensions of the picture before any modification. If the picture was not resized, `origWidth` and `origHeight` return the same values as `width` and `height`. Those values are expressed in the current default units for the document.

Note: It may be convenient to change the current unit to pixels for some computations.

See also

No reference.

WR GET PICTURE SIZE (area; width; height; origWidth; origHeight)

Parameter	Type		Description
area	Longint	→	4D Write area
width	Number	←	Current width of the picture
height	Number	←	Current height of the picture
origWidth	Number	←	Width of the original picture
origHeight	Number	←	Height of the original picture

Description

The WR GET PICTURE SIZE command allows you to retrieve information about the size of a selected picture. That picture must be located in the text flow. To get size information about a picture embedded in a page, use the WR GET PICTURE IN PAGE INFO command. For the WR GET PICTURE SIZE command to operate properly, the picture has to be the only element of the selection.

`height` is the picture height. It is expressed in the current default units for the document.

`width` is the picture width. It is expressed in the current default units for the document.

`origHeight` and `origWidth` are respectively the original height and width before the picture was resized. If `origHeight` and `origWidth` are identical to `height` and `width` the picture has not been resized. `origHeight` and `origWidth` are expressed in the current document unit.

Note : If you want to select a picture, you can use the WR SELECT command.

See also

No reference.

WR Get selected picture (area; status) → Picture

Parameter	Type		Description
area	Longint	→	4D Write area
status	Integer	←	Picture status
Function result	Picture	←	Selected picture

Description

The WR Get selected picture command returns a copy of the picture currently selected in the 4D Write area referenced by area.

The status parameter can return any of the following values:

- If status = -1, no picture is selected.
- If status = 0, the selected picture is in the text flow.
- If status > 0, the selected picture is in the page.

status can help you identify the picture when using WR GET PICTURE IN PAGE INFO, WR SET PICTURE IN PAGE INFO or WR DELETE PICTURE IN PAGE.

See also

No reference.

WR Get styled text (area) → BLOB

Parameter	Type		Description
area	Longint	→	4D Write area
Function result	BLOB	←	Formatted text

Description

The WR Get styled text command returns the selected text in the 4D Write area referenced by area a BLOB field or variable. The structure of the BLOB returned represents the selected text with both character and paragraph formatting included, although without style sheets.

Text that is returned using the WR Get styled text command can be placed into another 4D Write document using the WR INSERT STYLED TEXT command. The page layout of the 4D Write document into which the styled text is inserted will not be affected by the insertion.

By using the WR Get styled and the WR INSERT STYLED TEXT commands you can simulate a Copy/Paste operation while using a BLOB as a buffer instead of the clipboard.

Warning: The BLOB returned by WR Get styled text cannot be used with the WR BLOB TO AREA car command since it does not include all the elements of a 4D Write are.

See also

No reference.

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

Parameter	Type		Description
area	Longint	→	4D Write area
picture	Picture	→	Picture to insert
destination	Longint	→	Location of the insertion
horizPos	Number	→	Horizontal position in the page
verticalPos	Number	→	Vertical position in the page
behind	Integer	→	0=picture above the text 1=picture in background
firstPage	Integer	→	1=Picture is not in first page, 0=picture is displayed on all pages

Description

The WR INSERT PICTURE command inserts a picture in the 4D Write area referenced by area at the location specified by destination, horizPos and verticalPos .

picture can either be a picture field or a picture variable.

The destination optional parameter allows you to define where the picture will be inserted.

- If you want the picture to be inserted into the text flow, pass 0 in destination or omit the parameter. In this case the other parameters will not be used and the picture will either be inserted at the location of the insertion point or will replace the current selection.
- If you want to insert the picture into the page, use one of the following options:
 - If destination is greater than 0, the picture will be displayed in the page whose number is destination. The value of destination must take into account the page numbering as it is defined in the Preferences dialog.
 - If destination equals -1, the picture will be under the text and it will be displayed in all the pages
 - If destination equals -2, the picture will be under the text and it will be displayed in all the right pages
 - If destination equals -3, the picture will be under the text and it will be displayed in all the left pages
 - If destination equals -4, the picture will be under the text and it will be visible in the page containing the insertion point.

The horizPos and verticalPos optional parameters are expressed in the current default unit for the document. Those two parameters set the coordinates of the picture's upper left corner in relation to the upper left corner of the page.

The `behind` optional parameter allows you to define whether the picture will be behind or in front of the text.

- If `behind` equals 1, the picture will be behind the text. In this case it is necessary to pay attention to the text and paragraph background attributes. Selecting "None" will allow you to see the picture behind the text.
- If `behind` equals 0, the picture will be in front of the text.

The `firstPage` optional parameter applies only if `destination` equals -1, -2 or -3.

- If `firstPage` equals 1, the picture will not be displayed in the first page.
- If `firstPage` equals 0, the picture will also be displayed in the first page.

See also

No reference.

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

Parameter	Type		Description
area	Longint	→	4D Write area
pictureNumber	Longint	→	Picture number
page	Longint	→	Location of the picture
behind	Integer	→	0=picture is under the text 1=picture is over tthe text
firstPage	Integer	→	1=image is not present in first page otherwise 0
horizontPos	Number	→	Horizontal position in page
verticalPos	Number	→	Vertical position in page
width	Number	→	Current picture width
height	Number	→	Current picture height

Description

The WR SET PICTURE IN PAGE INFO command allows you to modify the properties of the picture whose number was passed in pictureNumber.

Warning : this command is not to be used for pictures that are inserted in the text flow .

page allows you to define what page the picture is to be displayed in. To do so, pass the page number in page. This number should take into account the page numbering as it is set in the Preferences dialog.

- If page equals -1, the picture will be displayed in all pages
- If page equals -2, the picture will be displayed in all right pages
- If page equals -3, the picture will be displayed in all left pages
- If page equals -4, the previous value is not modified.

- behind

If behind equals 0, the picture will appear above the text

If behind equals 1, the picture will appear behind the text. The text will then have a transparent background unless a background color was previously selected for it.

- firstPage

If firstPage equals 0, the picture will be displayed on all pages.

If firstPage equals 1, the picture will be displayed on all pages except the first page.

horizontPos and verticalPos allow you to set the horizontal and vertical coordinates of the upper left corner of the picture in relation to the upper left corner of the physical page. The value for horizontPos can be between 0 and the total page width. In this case, the

Note: when pasting a picture in the User environment, the printer margins are taken into account.

width and height allow you to set the new dimensions of the picture. Values are expressed in the current default units for the document.

Note: Passing -1 in the following parameters will not modify their initial value: behind, firstPage, horizontPos, verticalPos, width, height.

See also

No reference.

WR SET PICTURE SIZE (area; width; height)

Parameter	Type		Description
area	Longint	→	4D Write area
width	Number	→	New picture width
height	Number	→	New picture height

Description

The WR SET PICTURE SIZE command allows you to modify the size of the selected picture in the 4D Write area referenced by area.

This command has no effect on background pictures. To resize background pictures, use the WR SET PICTURE IN PAGE INFO command.

width and height are expressed in the current default units for the document. The values given must be within the page or within the column, when using multiple columns.

To use pixels as a unit, you can temporarily change the current default unit for the document and set it back after calling WR SET PICTURE SIZE.

See also

No reference.

8 WR Printing

WR INSERT PAGE NUMBER (area; format)

Parameter	Type		Description
area	Longint	→	4D Write area
format	Integer	→	Format type

Description

The WR INSERT PAGE NUMBER command allows you to insert at the cursor location a reference that displays the current page number. This reference can be placed in the main text, footer or header area. You can use the WR SET FRAME command to place the cursor in whichever area you choose.

format allows you to choose the display format for the page number. These formats are identical to the formats available in the Insert page number dialog.

Format Type	Value
123	0
abc	1
ABC	2
Roman characters (lower case)	3
Roman characters (upper case)	4

See also

No reference.

WR PRINT (area; mode; nbCopies)

Parameter	Type		Description
area	Longint	→	4D Write area
mode	Integer	→	0=Values 1=References
nbCopies	Integer	→	Number of copies to be printed

Description

The command WR PRINT prints the document contained in area. This command is the procedural equivalent of choosing Print from the File menu.

WR PRINT prints area one time. Use WR PRINT MERGE if you want to print area once for each record in a selection.

If mode equals 1, referenced elements appear between left and right double angle brackets (« ») in your 4D Write area. If mode equals 0, the values of the referenced elements will be printed in the 4D Write area.

The nbCopies parameter controls the number of copies to be printed.

Example

The following example is the script for a button used on the form that contains area. If you click on this button, area will be printed.

```
WR PRINT (area; 0;1)
```

See Also

No reference.

 WR PRINT MERGE (area; table; display)

Parameter	Type		Description
area	Longint	→	4D Write area
table	Integer	→	File number
display	Integer	→	Display/suppress the print settings dialog box

Description

The command **WR PRINT MERGE** prints the document contained in Area once for each record in the selection of table. table is the number of the merging table. If table equals 0, **WR PRINT MERGE** displays the standard Create Merge Selection dialog box, allowing you to specify the table and change the selection of records for that table.

If display equals 0, the Print Settings dialog box does not appear. If display equals 1, the Print Settings dialog box appears.

Example

The following example prints a letter for each record in the [Clients] table. The letter is stored in a record of the [Letters] table.

```

ALL RECORDS (Clients)
  `Select all clients
QUERY ([Letters];[Letters]Ref="Expedite")
  `Look for Expedite template
Temp:=WR Picture to offscreen area ([Letters]Doc_)
  `Place template in offscreen area
WR PRINT MERGE (Temp;3)
  `Merge the template with the selection in table 3
WR DELETE OFFSCREEN AREA (Temp)
  `Delete the offscreen area
  
```


9 WR Rulers

WR ADD STYLESHEET TAB (area; styleSheetNumber; location; justification; fillCharacter)

Parameter	Type		Description
area	Longint	→	4D Write area
styleSheetNumber	Longint	→	Stylesheet number
location	Number	→	Tab location
justification	Integer	→	Justification value for the tabulation
fillCharacter	Alpha	→	Selected fill character

Description

The **WR ADD STYLESHEET TAB** command allows you to add a new tab stop to the list of tab stops that the parameter `parameter styleSheetNumber` refers to. Using the **WR ADD STYLESHEET TAB** command, you can set the tab position, its type and its fill character.

If there is already tab stop at position, it will be replaced by the tab stop you just defined.

Note : Text that uses the style sheet you want to modify will not be updated unless you call the **WR UPDATE STYLESHEET** command to update text that uses that style sheet.

position is the distance from the left margin (expressed in the document's default units).

The justification optional parameter determines the tab stop type. You can either use the value or one of the following predefined constants.

Value	Constant	Text alignment
1	<u>wr left tab</u>	Left aligned
2	<u>wr centered tab</u>	Centered
3	<u>wr right tab</u>	Right aligned
4	<u>wr decimal tab</u>	Decimal
5	<u>wr vertical separator tab</u>	Vertical separator

If justification is omitted a left aligned tab is created.

Note : The list of constants is available in Appendix D.

The fillCharacter optional parameter can be any character whose ASCII code is between 33 and 127. This character will be added using the same font as the tab stop.

If fillCharacter is omitted or if you pass an empty string, no fill character will be inserted.

See also

No reference.

WR ADD TAB (area; position; justification; fillCharacter)

Parameter	Type		Description
area	Longint	→	4D Write area
position	Number	→	Tab location
justification	Integer	→	Justification value
fillCharacter	Alpha	→	Selected fill character

Description

The WR ADD TAB allows you to add a new tab at the location passed in position, measured from the left margin of the document. It also allows you to define the fill character and the justification of the new tab stop.

This tab stop will be added to all the paragraphs of the selection. If a tab stop already exist at this location, it will be replaced by the one you just created.

position is the distance from the left margin (expressed in the document's default unit).

The justification optional parameter determines the tab stop type. You can either use the value or one of the following predefined constants.

Value	Constant	Text alignment
1	<u>wr left tab</u>	Left aligned
2	<u>wr centered tab</u>	Centered
3	<u>wr right tab</u>	Right aligned
4	<u>wr decimal tab</u>	Decimal
5	<u>wr vertical separator tab</u>	Vertical separator

If justification is omitted a left aligned tab is created.

Note : The list of constants is available in Appendix D.

The fillCharacter optional parameter can be any character whose ASCII code is between 33 and 127. This character will be added using the same font as the tab stop.

If fillCharacter is omitted or if you pass an empty string, no fill character will be inserted.

Example

(1) The following example create a left tab stop, 50 units away from the left margin with a dot as fill character.

WR ADD TAB (area;50;1;".")

ou

WR ADD TAB (area;50;wr left tab;"")

WR DELETE TAB (area; tabNum)

Parameter	Type		Description
area	Longint	→	4D Write area
tabNum	Longint	→	Tabulation number

Description

The WR DELETE TAB deletes the tab whose number (counting left-to-right) is passed in tabNum from the 4D Write area referenced by area. . If other tabs are located at the same position, they too will be deleted.

Note: If the selection consists of several paragraphs, the tab numbering applies to the last selected paragraph.

See also

No reference.

WR GET TAB (area; tabNumber; position; alignment; fillCharacter)

Parameter	Type		Description
area	Longint	→	4D Write area
tabNumber	Longint	→	Tab number
position	Number	←	Tab position
alignment	Integer	←	Justification value for the tab
fillCharacter	String	←	Fill character

Description

The WR GET TAB command returns the position, the alignment and the fill character for the tab whose number was passed in tabNumber and in the current ruler of area. The current ruler is the ruler in which the insertion point appears, or the last ruler when several paragraphs are selected.

- tabNumber

To know the number of tabs in the paragraph, you can use: WR Get text properties(area;45;Uniform), which will return the number of tab stops. You can then loop through the tab numbers to retrieve all the parameters of the current ruler.

- position

position is the distance from the left document margin to the tab stop, expressed in the current default units of the document.

- alignment

alignment is the alignment type of the tab.

Value	Text alignment
1	Left alignment
2	Centered
3	Right alignment
4	Decimal
5	Vertical separator

- fillCharacter can be any character whose ASCII code is contained between 33 and 127. If fillCharacter is an empty string, then there is no fill character in the tab stop setting.

See also

No reference.

WR SET TAB (area; tabNumber; position; alignment; fillCharacter)

Parameter	Type		Description
area	Longint	→	4D Write area
tabNumber	Longint	→	Tabulation number
position	Number	→	New tabulation position
alignment	Integer	→	New value for the tabulation justification
fillCharacter	String	→	New character selected as fill character

Description

The WR SET TAB allows you to set the parameters of the tab stop whose number was passed in tabNumber (tabs are counted left to right). The WR SET TAB command will move the tab stop to position and will set the fill character as well as the alignment of the tab stop.

The selected tab stop will be modified for all the paragraphs of the current selection. If a tab stop already exists at the new location it will be replaced by the tab stop you just modified.

position is the distance from the left margin. position is expressed in the current default unit for the document.

alignment specifies the alignment for the tab stop. You can either use the value or the constant.

Value	Constant	Text alignment
1	<u>wr left tab</u>	Left alignment
2	<u>wr centered tab</u>	Centered
3	<u>wr right tab</u>	Right alignment
4	<u>wr decimal tab</u>	Decimal
5	<u>wr vertical separator tab</u>	Vertical separator

Note: The list of text properties and their references are available in the Appendix D, in the WR Tabs theme.

fillCharacter can be any character whose ASCII code is contained between 33 and 127. This character is displayed in the same font as the modified tab stop.

See also

No reference.

10 WR Style Sheet

WR APPLY STYLESHEET (area; styleSheetNumber)

Parameter	Type		Description
area	Longint	→	4D Write area
styleSheetNumber	Longint	→	Stylesheet number

Description

The WR APPLY STYLESHEET command applies to the current selection in the 4D Write area designated by area the style sheet whose number is passed in styleSheetNumber. The formats of the style sheet will then be applied to the selection and the selection will appear as using that style sheet (when the cursor is in the text, the style sheet will be displayed in the style sheet drop-down list from the Style toolbar).

If styleSheetNumber does not correspond to any style sheet, an error will be returned.

WR Create stylesheet (area; name{; applyTo{; shortCut{}}) → Longint

Parameter	Type		Description
area	Longint	→	4D Write area
name	Alpha	→	Stylesheet name
applyTo	Longint	→	0=characters 1=paragraphs
shortCut	String	→	One character
Function result	Longint	←	Stylesheet reference number

Description

The WR Create stylesheet creates a new style sheet and returns the number that was assigned to it. The features of the new style sheet are set by the parameters name, applyTo and shortCut. You can modify the style sheet by using the WR SET STYLESHEET TEXT PROP, WR SET STYLESHEET FONT, WR SET STYLESHEET TAB and the style sheet reference number .

- name : the length of a style sheet name is limited to 31 characters.

The applyTo optional parameter

- If applyTo equals 0, the style sheet will be a character stylesheet.
 - If applyTo equals 1, the style sheet will be a paragraph stylesheet.
- If applyTo is omitted, the style sheet will be a character style sheet.

- The shortCut optional parameter allows you to assign a keyboard shortcut to the style sheet. It only accepts one character. To use the shortcut you will need to press the key passed in this parameter with the Ctrl key (on Windows) or the Command key (on Mac OS). It is recommended that you use a number in order to avoid any conflict with the standard 4D Write keyboard shortcuts.

If shortCut is omitted or if it is an empty character string no shortcut will be assigned to the style sheet.

See also

No reference.

WR DELETE STYLESHEET (area; stylesheetNum)

Parameter	Type		Description
area	Longint	→	4D Write area
stylesheetNum	Longint	→	Stylesheet number

Description

The WR DELETE STYLESHEET command deletes the style sheet whose number was passed in styleSheetNum from the 4D Write area referenced by area. .

Warning: system style sheets cannot be deleted. You can use the WR GET STYLESHEET INFO command to determine if the style sheet is protected from deletion.

See also

No reference.

WR DELETE STYLESHEET TAB (area; stylesheetNumber; tabNumber)

Parameter	Type		Description
area	Longint	→	4D Write area
stylesheetNumber	Longint	→	Stylesheet number
tabNumber	Longint	→	Number of the tabulation to delete

Description

The WR DELETE STYLESHEET TAB command deletes the tab stop whose number was passed in tabNumber from the styleSheetNumber style sheet, in the 4D Write area referenced by area. DStyle sheets are numbered from top to bottom, as listed in the style sheet dialog. This command has no effect on the selected text, even if it currently uses the styleSheetNumber style sheet.

To update the text that uses the modified style sheet, you need to use the WR UPDATE STYLESHEET command.

See also

No reference.

WR Get stylesheet font (area; stylesheetNumber)

Parameter	Type		Description
area	Longint	→	4D Write area
stylesheetNumber	Longint	→	Stylesheet number

Description

The WR Get stylesheet font command returns the name of the font that was assigned to the style sheet whose number was passed in styleSheetNumber in the 4D Write area referenced by area. Style sheet are numbered from top to bottom as shown in the style sheet dialog. If no font is defined for that style sheet, an empty string is returned.

See also

No reference.

WR GET STYLESHEET INFO (area; stylesheetNumber; name; applyTo; protected; shortcut)

Parameter	Type		Description
area	Longint	→	4D Write area
stylesheetNumber	Longint	→	Stylesheet number
name	String	←	Stylesheet name
applyTo	Integer	←	0=characters, 1=paragraphs
protected	Integer	←	0= non protected, 1= protected
shortcut	String	←	One character

Description

The WR GET STYLESHEET INFO allows you to retrieve information about the style sheet whose number is passed in `styleSheetNumber` and which is contained in the 4D Write area referenced by `area`.

`name` returns the title of the style sheet.

- `applyTo`

If `applyTo` is equal to 0, the style sheet will only apply to characters.

If `applyTo` is equal to -1, the style sheet will only apply to paragraphs.

- `protected`

If `protected` is equal to 0, the style sheet is not protected, thus it is not a system style sheet.

If `protected` is equal to 1, the style sheet is protected, it is therefore a system style sheet and it cannot be deleted.

`shortcut` returns the shortcut assigned to the style sheet, if any. It consists of only one character. When using that shortcut you will need to hold down the Ctrl key (on Windows) or the Command key (on Mac OS) while pressing the shortcut key.

If `shortcut` is an empty string, no shortcut is assigned to `styleSheetNumber`.

See also

No reference.

WR GET STYLESHEET TAB (area; stylesheetNum; tabNumber; position; justification; fillCharacter)

Parameter	Type		Description
area	Longint	→	4D Write area
stylesheetNum	Longint	→	Stylesheet number
tabNumber	Longint	→	Tab number
position	Number	←	Position of the tab
justification	Integer	←	Justification value for the tab
fillCharacter	String	←	Selected fill character

Description

The WR GET STYLESHEET TAB allows you to retrieve the settings of the tab stop whose number was passed in tabNumber and which belongs to the style sheet whose number was passed in styleSheetNumber in the 4D Write area referenced by area.

- tabNumber

To know the number of tabs in the style sheet, you can use: WR GET STYLESHEET INFO(area;styleSheetNumber;wr tab;applyTO), which will return the number of tab stops.

- position

position is the distance from the left document margin to the tab stop, expressed in the current default units of the document.

- alignment

alignment is the alignment type of the tab.

Value	Text alignment
1	Left alignment
2	Centered
3	Right alignment
4	Decimal
5	Vertical separator

- fillCharacter can be any character whose ASCII code is between 33 and 127. If fillCharacter is an empty string, then there is no fill character in the tab stop setting.

See also

No reference.

WR Get stylesheet text prop (area; stylesheetNumber; property; apply)

Parameter	Type		Description
area	Longint	→	4D Write area
stylesheetNumber	Longint	→	Stylesheet number
property	Integer	→	Number of the text property to read
apply	Integer	→	0=the property is not applied 1=the property is applied

Description

The WR Get stylesheet text prop command allows you to know, for area, whether the property passed in property is applied to the selection.

- property

If property = 7 (wr font number Constant), the returned value is an internal number. 4D Write sequentially assigns font numbers to fonts as they are used. This number can only be used by the WR SET STYLESHEET TEXT PROP command. It is recommended that you should use the WR Get stylesheet font and WR SET STYLESHEET FONT whose operation is based on font names.

The property 15 (wr stylesheet number Constant) has not meaning for this function.

If property = 64 (wr tab Constant), WR Get stylesheet text prop returns the number of tab stops set for the style sheet.

For color properties, the returned value will respect the following format (as in 4D and in the former version of 4D Write): 0x00RRVVBB. To separate the RGB values, use the WR COLOR TO RGB command.

If -1 is returned for the properties 11 (wr strikethrough color Constant), 12 (wr underline color Constant), or 13 (wr shadow color Constant), these elements are in the same color as the text.

If -1 is returned for the property 10 (wr text back color Constant), there is no background color selected for the text.

Note : property can be set using constants.

The list of the text properties constants are available in Appendix D.

- If applyTo is equal to 1 , the style sheet takes into account the property and modifies it in the style sheet definition.

WR SELECT PICTURE IN PAGE (area; pictureNum)

Parameter	Type		Description
area	Longint	→	4D Write area
pictureNum	Longint	→	Picture number

Description

The WR SELECT PICTURE IN PAGE command allows you to select the picture whose number is passed in pictureNum. For the command to operate properly, the picture must be located in the page (not in the text flow). If you want to select a picture located in the text flow, you can use the WR SELECT command.

See also

No reference.

WR SET STYLESHEET FONT (area; stylesheetNumber; font)

Parameter	Type		Description
area	Longint	→	4D Write area
stylesheetNumber	Longint	→	Stylesheet number
font	Alpha	→	Font name

Description

The WR SET STYLESHEET FONT allows you to modify the character font for the style sheet whose number is passed in `styleSheetNumber` in the 4D Write document referenced by `area`.

- font

Pass in `font` the name of the font you want to apply. If you want to apply the style sheet to the selection, pass an empty character string in `font`.

See also

No reference.

WR SET STYLESHEET INFO (area; styleSheetNumber; name; applyTo; shortCut)

Parameter	Type		Description
area	Longint	→	4D Write area
styleSheetNumber	Longint	→	Style sheet number
name	Alpha	→	Name of the style sheet
applyTo	Integer	→	0=characters 1=paragraphs
shortCut	Alpha	→	one character "" if no shortcut

Description

The WR SET STYLESHEET INFO command allows you to modify the properties of the style sheet whose reference number is passed in styleSheetNumber and which is contained in the 4D Write document with the reference number area. The style sheet number corresponds to the order of appearance the style sheet when it is either displayed in the Style sheet drop-down list or in the list in the Style sheets dialog.

- name

If name is an empty string, the original name of the style sheet will not be modified.

Warning: two style sheets can both have the same name, however they will always have different reference numbers.

- applyTo

If applyTo equals -1, the current value will remain the same.

If applyTo equals 0, the style sheet applies to characters.

If applyTo equals 1, the style sheet applies to paragraphs.

A paragraph style sheet always apply to all the paragraphs of the selection, even if the first or last paragraphs are partially selected. By default a newly created style sheet is a character style sheet.

- shortCut

The shortCut optional parameter allows you to assign a keyboard shortcut to the style sheet. It only accepts one character. To use the shortcut you will need to press the key passed in this parameter with the Ctrl key (on Windows) or the Command key (on Mac OS). It is recommended that you use a number in order to avoid any conflict with the standard 4D Write keyboard shortcuts.

If shortCut is omitted or if it is an empty character string no shortcut will be assigned to the style sheet.

- styleSheetNumber

If you want the style sheet number to remain identical, you need to call the WR GET STYLESHEET INFO command and use the reference number returned by that command .

See also

WR GET STYLESHEET INFO.

WR SET STYLESHEET TAB (area; stylesheetNumber; tabNumber; position; alignment; fillChar)

Parameter	Type		Description
area	Longint	→	4D Write area
stylesheetNumber	Longint	→	Stylesheet number
tabNumber	Longint	→	Tab number
position	Number	→	New tab position
alignment	Number	→	New value for the tab alignment
fillChar	String	→	Selected fill character

Description

The WR SET STYLESHEET TAB command allows you to modify the parameters of the tab stop whose number was passed in tabNumber (tabs are counted left to right) belonging to the style sheet whose number was passed in styleSheetNumber (style sheets are counted top to bottom as shown in the style sheets dialog). The WR SET STYLESHEET TAB command will move the tab stop to position and will set the fill character as well as the alignment of the tab stop.

This command has no effect on the selected text even if it uses the style sheet being modified.

- If you want to update the text that uses that style sheet, call the WR UPDATE STYLESHEET command after modifying the style sheet definition.
- If you want to immediately apply the new tab properties of the style sheet to both the style sheet and the current selection, use the WR APPLY STYLESHEET command.

If a tab stop already exists at the new location in the style sheet, it will be replaced by the tab stop that is the subject of this command.

position is the distance from the left margin to which you want to move the tab stop. position is expressed in the current default unit for the document.

alignment specifies the type of alignment you want to select for the tab stop. You can either use the value or the constant.

Value	Constant	Text alignment
1	<u>wr left tab</u>	Left alignment
2	<u>wr centered tab</u>	Centered
3	<u>wr right tab</u>	Right alignment
4	<u>wr decimal tab</u>	Decimal
5	<u>wr vertical separator tab</u>	Vertical separator

`fillCharacter` can be any character whose ASCII code is contained between 33 and 127. This character is displayed in the same font as the modified tab stop.

See also

No reference.

WR SET STYLESHEET TEXT PROP (area; styleSheetNumber; property; value; apply)

Parameter	Type		Description
area	Longint	→	4D Write area
styleSheetNumber	Longint	→	Stylesheet number
property	Longint	→	Number of the property to read
value	Number	→	Value for the property chosen
apply	Integer	→	1= apply the value to the property 0=do not apply the value to the property

Description

The WR SET STYLESHEET TEXT PROP command allows you to modify the text attributes of the style sheet whose number is passed in styleSheetNumber.

- If you want all the text that currently uses this style sheet to be updated, call the WR UPDATE STYLESHEET command after modifying the style sheet definition.
- If you want to immediately apply with this command the new text properties of the style sheet to both the style sheet and the current selection, use the WR APPLY STYLESHEET command.
- The meaning given to the value parameter depends on the property value used. If the value for property is constant property wr bold or 0, values for value can either be 1 (True) or 0 (False). If the value for property is constant property wr font size or 8, values for value can be 9, 10, 12...

Note: property and value can be set using constants.

The list of text properties and their references are available in the Appendix D, in the WR Text properties theme.

The list of text properties values and their references are available in the Appendix C, in the WR Text properties values theme. You can either pass the value or the constant.

- apply
Pass 1 in the apply parameter if you want to apply the changes to the property. If you do so, value will define the new value for the property.

Pass 0 in the apply parameter if you do not want to apply the changes to the property. If you do so, value will have no effect.

WR UPDATE STYLESHEET (area; stylesheetNumber)

Parameter	Type		Description
area	Longint	→	4D Write area
stylesheetNumber	Longint	→	Stylesheet number

Description

The WR UPDATE STYLESHEET command updates the displayed formatting of all the text using the style sheet referenced by styleSheetNumber in the 4D Write area referenced by area. After this command is executed, all text formatted with the referenced style will be formatted according to the current definition of that style.

See also

No reference.

11 WR Text Manipulation

WR BACKSPACE (area)

Parameter	Type		Description
area	Longint	→	4D Write area

Description

The command WR BACKSPACE simulates pressing of the Delete or Backspace key. If characters are selected in area, they are deleted. If no characters are selected, WR BACKSPACE acts the same as pressing Delete or Backspace. One character at a time is deleted and the insertion point moves one character to the left.

WR DELETE SELECTION (area)

Parameter	Type		Description
area	Longint	→	4D Write area

Description

The WR DELETE SELECTION allows you to delete the current text selection from the 4D Write area referenced by area.

Using the following statement will have the same effect as using the WR DELETE SELECTION command: WR EXECUTE command(area; wr cmd clear).

Note : The value of the wr cmd clear constant is 6.

If there is no current selection, the command has no effect, unlike the WR BACKSPACE command that would delete the character located before the cursor.

See also

No reference.

WR Direct find (blob; charString; wholeWord; upperCase) → Longint

Parameter	Type		Description
blob	Blob	→	Blob
charString	Alpha	→	Character string to be searched for
wholeWord	Integer	→	0=partial match 1=whole word
upperCase	Integer	→	0=search after the insertion point 1=search the whole document
Function result	Longint	←	Search status

Description

The WR Direct find command allows you to directly search for a character string in a BLOB that contains a 4D Write area. Using this command does not require the BLOB to be opened in a 4D Write area first. This means that this command executes very quickly.

If the character string is found, WR Direct find returns the position of the character string in the text.

If the search was unsuccessful, WR Direct find returns -1.

If blob does not represent the contents of a 4D Write area, WR Direct find returns -2.

wholeWord and upperCase allow you to choose some options for the search:

- If wholeWord equals 1, only the whole word will be searched for. For a string to be found using this option, it must occur between punctuation characters (space, comma and so on). If wholeWord does not equal 1, the character string can either be a whole word or part of a longer word.
- If upperCase equals 1, the search will look for a character string whose case matches the case of the original string .

WR Find (area; charString; wholeWord; upperCase; wrap) → Longint

Parameter	Type		Description
area	Longint	→	4D Write area
charString	Alpha	→	String of characters to be searched for
wholeWord	Integer	→	0=partial match 1=whole word
upperCase	Integer	→	0=ignore uppercase 1=takes uppercase into account
wrap	Integer	→	0=search after the insertion point 1=search the whole document
Function result	Longint	←	Search status

Description

The WR Find allows you to search for a character string in a 4D Write area. You can retrieve the position of the words found using the WR GET WORDS command. You can retrieve the position of the selection found using the WR GET SELECTION command. If the character string is found, WR Find returns 1 and select the first occurrence.

If the search was unsuccessful, WR Find returns 0 and the current selection is not modified. If area does not exist, WR Find returns -1.

wholeWord and upperCase allow you to define some options of the search:

- If wholeWord equals 1, only the whole word will be searched for. For a string to be found using this option, it must occur between punctuation characters (space, comma and so on). If wholeWord does not equal 1, the character string can either be a whole word or part of a longer word.
- If upperCase equals 1, the search will look for a character string whose case matches the case of the original string.

wrap allows you to define whether the search applies to the entire document.

- If wrap equals 1, the search will be performed on the entire document, otherwise the search will be performed from the position of the insertion point or the selection to the end of the document.

WR Get font (area; sameFont)

Parameter	Type		Description
area	Longint	→	4D Write area
sameFont	Longint	←	1 if the font is the same for the entire selection, otherwise 0

Description

The WR Get font command returns the font name of the font applied to the last character of the selection in the 4D Write area referenced by area.

- sameFont
If sameFont = 1, the same font is applied to the whole selection.
If sameFont = 0, other fonts are used in the selection.

See also

No reference.

WR GET PARAGRAPHS (area; beginPara; endPara)

Parameter	Type		Description
area	Longint	→	4D Write area
beginPara	Longint	←	Beginning of the paragraph to return
endPara	Longint	←	End of the paragraph to return

Description

The **WR GET PARAGRAPHS** command returns the position of the first character of the first paragraph of the selection and the position of the carriage return of the last paragraph of the selection, in the 4D Write area referenced by area.

Example :

The following example scans the document and retrieves the position of the first and last character for each paragraph.

```

`Locating the cursor at the beginning of the area
WR SET SELECTION (area;0;0)
`Counting the number of paragraphs in the document
nbPara:=WR Count(Zone;wr_nb_paragraphs)
`Processing paragraphs one by one
For ($i;1;nbPara)
  `retrieving the position of the first and last characters
⇒ WR GET PARAGRAPHS(area;begin;Pos)
  `Relocating after the last processed paragraph
  WR SET SELECTION (area;Pos;Pos)
End for

```

See also

No reference.

WR Get selected text (area) → Text

Parameter	Type		Description
area	Longint	→	4D Write area
Function result	Text	←	Text selected in area

Description

The command WR Get selected text returns the selected text in area.

The maximum number of characters 4th Dimension can store in a field or variable is 32,000. Therefore, WR Get selected text will return a maximum of 32,000 characters. If more than 32,000 characters are selected, this function returns an empty string and an error will be reported by WR Error number.

Example

The following example places the selected text in area into the variable vText.

```
vText:=WR Get selected text (area)
```

See Also

WR GET SELECTION.

WR GET SELECTION (area; first; last)

Parameter	Type		Description
area	Longint	→	4D Write area
first	Longint	←	Receives first character
last	Longint	←	Receives last character

Description

The command WR GET SELECTION returns into the First and Last variables the positions of the selected text in Area.

first is always one less than the first character selected. last is always equal to the last character selected. If first and last are equal, no text is selected and the insertion point is positioned after the character described by first.

Example

The following example sets the margins of the whole document, and retrieves the original selection:

```
⇒  WR GET SELECTION(area;DebSel;FinSel) `re-reading the current selection
    WR EXECUTE COMMAND(area;wr cmd select all) `Select all
    `setting margins
    WR SET TEXT PROPERTY(area;wr left margin;49)
    WR SET TEXT PROPERTY(area;wr first indent;49)
    WR SET TEXT PROPERTY(area;wr right margin;504)
    WR SET SELECTION(area;DebSel;FinSel) `reseting the selection
```

See Also

No reference.

WR Get text (area; first; last) → Text

Parameter	Type		Description
area	Longint	→	4D Write area
first	Longint	→	First character of text
last	Longint	→	Last character of text
Function result	Text	←	Text between first and last characters

Description

The command WR Get text returns the text in Area between the character described by first and the character described by last.

The maximum number of characters 4th Dimension can store in a field or variable is 32,000. Therefore, WR Get text can return a maximum of 32,000 characters. If the difference between last and first is greater than 32,000 characters, and the document has at least last number of characters, WR Get text returns an empty string and an error is reported by WR Error number.

If Last is greater than the number of characters in the document and the difference between last and first is less than 32,000, WR Get text returns all of the characters to the end of the document. If last is less than or equal to first, WR Get text returns an empty string.

WR Get text does not change the selected text in area.

Example

The following example places the first 100 characters of area into the variable vText.

```
vText:=WR Get text (area;0;100)
```

See Also

No reference.

WR Get text property (area; property; sameProperty)

Parameter	Type		Description
area	Longint	→	4D Write area
property	Integer	→	Property number
sameProperty	Integer	←	1 if the whole selection has that property, 0 if part or all of the selection does not have the property

Description

The WR Get text property command allows you to determine whether the property passed in property is used in the current selection of the 4D Write area referenced by area.

- If sameProperty is equal to 1, the property is applied to the whole selection.
 - If sameProperty is equal to 0, the property is not applied to the whole selection.
- The returned value then corresponds to the status of the last character of the selection.

- property

If property = 7 (wr font number Constant), the returned value is an internal number. 4D Write sequentially assigns font numbers to fonts as they are used. This number can only be used by the WR SET TEXT PROPERTY command. It is recommended to use the WR Get stylesheet font and WR SET STYLESHEET FONT as their operation is based on font names.

If property = 15 (wr stylesheet number Constant) WR Get text property returns the style sheet number applied to the beginning of the selection. Styl sheets are numbered from top to bottom as shown in the Style dialog.

If property = 45 (wr tab Constant), WR Get stylesheet text prop returns the number of tab stops of the last paragraph of the selection.

For color properties, the returned value will use the following format (as in 4D and in the former version of 4D Write): 0x00RRVVBB. To separate the RGB values, use the WR COLOR TO RGB.

If -1 is returned for the properties 11 (wr strikethrough color Constant), 12 (wr underline color Constant), 13 (wr shadow color Constant), these elements are in the same color as the text.

If -1 is returned for the property 10 (wr text back color Constant), there is no background color selected for the text.

Note : property can be set using constants.

The list of the text properties constants is available in Appendix D.

See also

No reference.

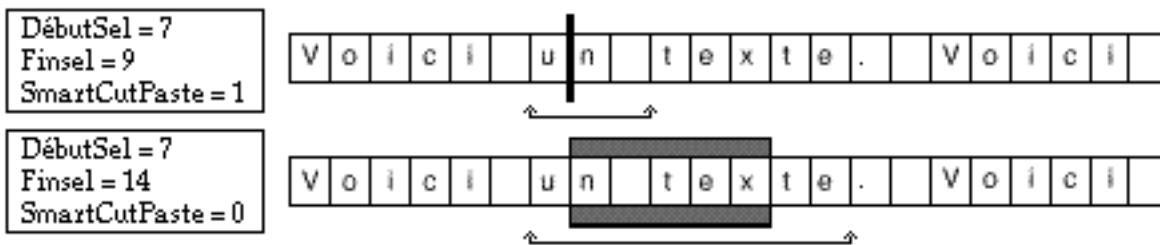
WR GET WORDS (; area; beginSel; endSel; smartCutPaste)

Parameter	Type		Description
area	Longint	→	4D Write area
beginSel	Longint	←	Beginning of the word to return
endSel	Longint	←	End of the word to return
smartCutPaste	Integer	←	1 if the last character is a space, otherwise 0

Description

The WR GET WORDS returns the position of the first character of the first word of the selection and the position of the last character of the last word of the selection. It also specifies if the last character of the selection is a space. If no text is selected, beginSel and endSel returns the first and last character of the word the cursor is in.

This command has no effect on the current selection.



If the selection begins in the middle of a word (or between the last character of a word and the next following space), debSel will return the position of the first character of that word.

If the selection ends in the middle of a word, there are two possible cases:

- If the word is followed by a space, endSel will include the space and smartCutPaste will return 1.
- If the word is not followed by a space, endSel will include the last character of the word and smartCutPaste will return 0.

Example :

The following example scans the document and retrieves the position of the first and last characters for each word.

```
    `Placing the cursor at the beginning of the area  
    WR SET SELECTION (area;0;0)  
    `Counting the number of words in the document  
    nbWords:=WR Count(area;wr nb words)  
    `Processing the words one by one  
    For ($i;1;nbWords)  
        `retrieving the position of the first and last character of the word  
⇒      WR GET WORDS(area;beginning;pos)  
        `relocating after the last processed word  
        WR SET SELECTION (area;Pos;Pos)  
    End for
```

See also

No reference.

WR INSERT STYLED TEXT (area; blob)

Parameter	Type		Description
area	Longint	→	4D Write area
blob	Blob	→	Variable or field

Description

The WR INSERT STYLED TEXT command inserts into the 4D Write area referenced by area the contents of blob. The insertion will either take place at the cursor location or it will replace the current selection. blob can either be a BLOB field or a BLOB variable. It is however mandatory that blob was initially created using the WR Get styled text command.

The internal format used to represent the styled text in blob is platform independent. It can be created using a Mac OS computer and be later inserted in a Windows document, or vice versa.

See also

No reference.

WR INSERT TEXT (area; first)

Parameter	Type		Description
area	Longint	→	4D Write area
first	Longint	→	First character of text

Description

The command **WR INSERT TEXT** inserts **Text** into **area**, replacing any selected characters. If no characters are selected, **Text** is placed at the insertion point. This command can be used in place of **WR INSERT EXPRESSION** or **WR INSERT FIELD** when you do not need automatic referencing.

Example

The following example places the text in the variable **vText** into **area**.

```
WR INSERT TEXT (Area;vText)
```

See Also

No reference.

WR Mouse to Selection (area; posHoriz; posVert; beginSel; endSel) → Integer

Parameter	Type		Description
area	Longint	→	4D Write area
posHoriz	Integer	→	Horizontal position of mouse in area
posVert	Integer	→	Vertical position of mouse in area
beginSel	Longint	→	Returns beginning of selection
endSel	Longint	→	Returns end of selection
Function result	Integer	←	Selection matching the position of the cursor

Description

The command WR Mouse to Selection returns the selection matching the position of the cursor. You can use this function, for example, in conjunction with the Mac OS Drag and Drop manager to find the location of the cursor when the mouse was released, and to paste an object there.

WR REPLACE (area; searchedFor; replaceWith; wholeWord; upperCase; wrap; replaceAll) → Longint

Parameter	Type		Description
area	Longint	→	4D Write area
searchedFor	String	→	Character string to search for
replaceWith	String	→	Replacement character string
wholeWord	Integer	→	Search for whole word
upperCase	Integer	→	Take uppercase characters into account
wrap	Integer	→	0=search from the selection 1=search the whole document
replaceAll	Integer	→	0=replace next 1=replace all
Function result	Longint	←	Number of occurrences replaced

Description

The WR REPLACE command allows you to emulate the Replace command menu of the Edit menu.

If `wholeWord` equals 1, only whole words will be searched for. For a string to be found using this option, it must occur between punctuation characters (space, comma and so on).

If `upperCase` equals 1, the search will look for a character string whose case matches the case of the original string .

If `replaceAll` equals 1, each occurrence of the character string will be replaced. If `replaceAll` does not equal 1, only the first occurrence will be replaced.

`wrap` allows you to define whether the search applies to the entire document. If `wrap` equals 1, the search will be performed on the entire document, otherwise the search will be performed from the position of the selection to the end of the document.

`wr_Replacer` returns the number of occurrences replaced.

See also

No reference.

WR SELECT (area; type; begin{; end})

Parameter	Type		Description
area	Longint	→	4D Write area
type	Integer	→	Type to select
begin	Longint	→	First character
end	Longint	→	Last character. Optional for certain values of
Type			

Description

The command WR SELECT selects text defined by type, begin, and end.

type	Selection	Comments
0	Characters	Selects the characters located between begin and end. In this case, this is the same as using WR SET SELECTION.
1	Reference	Selects the reference located at begin. end must be omitted.
2	Paragraphs	Selects the paragraphs located between begin and end. A picture is considered as a paragraph.
3	Ruler (paragraph attributes)	Selects the paragraphs that use the Nth ruler (counting from the beginning of the text). end must be omitted.
4	Picture	Selects the picture located at begin. end must be omitted.
5	Style (character attributes)	Selects the words that use the Nth style (counting from the beginning of the text). end must be omitted.
6	Word	Selects the word in which the insertion point is located.
7	Page break	Selects the page breaks located between begin and end. If you only want to select one page break, pass the same value in begin and end.
8	Column break	Selects the column breaks located between begin and end. If you only want to select one column break, pass the same value in begin and end.
9	Hyphen	Selects the hyphens located between begin and end. If you only want to select one hyphen, pass the same value in begin and end.
10	Page number	Selects the page numbers located between begin and end. If you only want to select one page number, pass the same value in begin and end.
11	Date and time	Selects the date and time variables located between begin and end. If you only want to select one date and time variable, pass the same value in begin and end.

Example

The following example executes different functions based on the presence or the absence of a Page break:

```
    `Setting the selection
    WR SET SELECTION (area;0;0)
    `Selecting the first page break
⇒ WR SELECT (area;7;1;1)
    `Retrieving the limits of the new selection
    WR GET SELECTION (area;$vlbegin;$vlend)
    If (($vlbegin=0) & ($vlend=0))
        `there is no page break
    Else
        `Do something with the page break
    End if
```

(2) The following example selects the references in the 4D Write area referenced by area and applies to them a style that makes them easy to spot.

```
    NbObjects:=WR Count(area;4)
    `Counting the number of references
    For (i;1;NbObjects)
⇒ WR SELECT(area;1;i)
    `Selecting each reference
    WR GET REFERENCE(area;TableNo;FieldNo;vName;vType)
    WR SET TEXT PROPERTY(area;wr bold;1)
    WR SET TEXT PROPERTY(LaZone;wr text color;wr blue)
    `Applying Blue and Bold to teh selection
    Fin de boucle
```

WR SET FONT (area; font)

Parameter	Type		Description
area	Longint	→	4D Write area
font	Alpha	→	Font name

Description

The WR SET FONT allows you to set the font for the current selection in the 4D Write area referenced by area.

- font
Pass in font the name of the font you want to apply.

See also

No reference.

WR SET SELECTION (area; first; last)

Parameter	Type		Description
area	Longint	→	4D Write area
first	Longint	→	First character
last	Longint	→	Last character

Description

The command **WR SET SELECTION** selects the text in Area described by the numbers first and last. Text is selected from first + 1 characters to last.

If first and last are equal, **WR SET SELECTION** places the insertion point after the character described by first. If last is greater than the length of the text in Area, **WR SET SELECTION** selects the text to the end of the document. If last is less than first, **WR SET SELECTION** does nothing.

Example

The following example selects the first ten characters in Area:

```
WR SET SELECTION (area;0;10)
```

See Also

No reference.

WR SET TEXT PROPERTY (area; property; value)

Parameter	Type		Description
area	Longint	→	4D Write area
property	Integer	→	Number of the text property to set
value	Number	→	Value for the selected property

Description

The **WR SET TEXT PROPERTY** command allows you to modify the text properties of the current selection in the 4D Write area referenced by area.

Be aware that if you pass a style sheet reference using the **WR SET TEXT PROPERTY**, the text will be labelled as using that style sheet but the style sheet attributes will not be applied to the text. The **WR APPLY STYLESHEET** command sets the property and applies it.

- `property` and `value` are to be used jointly.

If the value for `property` is (constant property `wr bold` or 0, values for `value` can either be 1 (True) or 0 (False).

If the value for `property` is constant property `wr font size` or 8, values for `value` can be 9, 10, 12 ...

If you pass an illegal property number, the error 1075 will be generated.

If you pass an illegal value for the selected property, the error 1076 will be generated.

Note: `property` and `value` can be set using constants.

The list of text properties and their references are available in the Appendix D, in the **WR Text properties theme**.

The list of text properties and their references are available in the Appendix D, in the **WR Text properties values theme**. You can either pass the value or the constant.

The list of error codes is available in appendix B.

See also

No reference.

12 WR Utilities

WR COLOR TO RGB (color; red; green; blue)

Parameter	Type		Description
color	Longint	→	Color
red	Longint	←	Receives red value
green	Longint	←	Receives green value
blue	Longint	←	Receives blue value

Description

The command **WR COLOR TO RGB** maps the color defined by **Color** into its three components: red, green, and blue. **color** is an internal number used by 4D Write and can be obtained with the **WR RGB to color** function.

Examples

The following example calculates the closest grey for a given color:

```
WR COLOR TO RGB (Color;Red;Green;Blue)
Blue:=(Blue+Green+Red)/3
Grey:=WR RGB To color (Blue;Blue;Blue)
```

WR Count (area; objectNumber)

Parameter	Type		Description
area	Longint	→	4D Write area
objectNumber	Integer	→	Object number

Description

The WR Count allows you to count the number of occurrences of a specific object in a specific area.

Objects that can be counted are:

Object	Object Number
Characters	0
Words	1
Paragraphs	2
Picture in text flow	3
References	4
Hyphens	5
Page breaks	6
Column breaks	7
Time objects	8
Page numbers	9
Lines	10
Pages	11
Style sheets	12
Images in pages (background)	13

- If objectNumber equals 3, background pictures will be ignored (if you want background pictures to be counted, objectNumber must equal 13).
- If objectNumber equals 12, WR Count returns the number of style sheets, including the standard style sheets (default style sheet).
- If objectNumber equals 13 and if an image is repeated in several pages (as selected in the picture properties dialog), the image counts as one.

WR Error number (area) → Integer

Parameter	Type		Description
area	Longint	→	4D Write area
Function result	Integer	←	Status of the last operation performed in Area by 4D Write

Description

The command WR Error number returns a number that represents the status of the last operation performed in Area by 4D Write. If WR Error number equals 0, the last operation did not cause an error. If WR Error number does not equal 0, then an error occurred during the last operation in area.

Use WR Error text to get a text explanation of the error. If the Debug window is open and an error occurs, you will also receive the error number in the Debug window.

Example

Refer to the example of the WR Error text command.

See Also

No reference.

WR Error text (error) → String

Parameter	Type		Description
error	Integer	→	Number of error
Function result	String	←	Text description of the error specified by Error

Description

The command **WR Error text** returns a text description of the error specified by **error**. You can use this function to receive a description of the error returned by **WR Error number**.

Example

The following example tests for an error and then displays a different error message depending upon whether or not the user is the Designer:

```
$Error:=WR Error number (Area)
If ($Error#0)
  If (Current user="Designer")
    ALERT (WR Error text ($Error))
  Else
    ALERT ("A problem has occurred. Please notify your manager.")
  End if
End if
```

See Also

No reference.

WR FONTS TO ARRAY (fonts)

Parameter	Type		Description
fonts	Alpha array	←	Receives array of available fonts

Description

The command `WR FONTS TO ARRAY` returns the list of available fonts in the fonts array. This list corresponds to that in the Font menu.

fonts should be declared as a String or Text type array.

Example

The code in this example stores the available system fonts in the aFonts array. You can use aFonts as a pop-up menu so users can choose fonts they want.

```
ARRAY STRING (256;aFonts;0)  
WR FONTS TO ARRAY (aFonts)
```

WR ON ERROR (method)

Parameter	Type		Description
method	String	→	Name of method

Description

The command **WR ON ERROR** installs Method to manage 4D Write errors. If method is an empty string, no method is called. After installation, 4D Write calls method when a 4D Write error occurs. When 4D Write calls method, it passes three parameters that can be used to manage the error:

Variable	Description
\$1	A long integer that represents the 4D Write area where the error took place
\$2	An integer that contains the error number
\$3	A text type that contains the text of the error

If you want to compile your database, you must declare the \$1 to \$3 variables using the **C_LONGINT** and **C_TEXT** commands.

Examples

(1) This example shows the installation of an error-handling method.

```
WR ON ERROR ("WRITE ERROR")
```

(2) The following method is **WRITE ERROR**. This method tests \$1 to determine in which area the error occurred. **WRITE ERROR** then presents an alert box with the error number and message.

```
If ($1=area)
  ALERT ("An error occurred in the 4D Write area 'Area'")
End if
ALERT ("Error number "+String($2)+Char(13)+$3)
```

See Also

No reference.

WR ON EVENT (area; event; method)

Parameter	Type		Description
area	Longint	→	4D Write area
event	Longint	→	Event code
method	String	→	Method to execute

Description

The command WR ON EVENT installs the Method as the method to be called whenever the event described by event occurs in area. Events are passed directly to Method before being handled by 4D Write.

If area equals 0, method becomes the default event method for all 4D Write areas until the database is closed. If an area has a specific event method installed, that method is called instead of the default.

The following table lists the possible values for event:

Event	Value
Event method is activated for all events	-1
Key down (including arrow keys, returns, tabs, etc.)	0
Double-click	1
Mouse-click	2
Unused (for compatibility with 4D Calc)	3
Unused (for compatibility with 4D Calc)	4
4D Write area activated or deactivated	5
Unused (for compatibility with 4D Calc)	6
Printing document	7
Ruler modification	8
Dynamic reference modified	9
4D Write area closing	10

When called, method receives six parameters that describe the state of area at the time of the event. You must explicitly type these parameters using compiler directives. The following table describes the parameters received by Method.

Variable	Type	Description
\$1	Long integer	4D Write area
\$2	Integer	Shift key
\$3	Integer	Alt (Windows); Option (Mac OS)
\$4	Integer	Ctrl (Windows); Command key (⌘) (Mac OS)
\$5	Integer	Event type
\$6	Integer	Changes depending on event type

\$1 returns the long integer that is the area ID where the event took place. \$2, \$3, and \$4 describe whether a specific modifier key was depressed at the time of the event. If the value equals 0, the key was not pressed. If the value equals 1, the key was pressed. \$5 returns the event type. \$6 varies depending on the type of event.

Method Variables and the Event Parameter (\$6)

If event equals 0, \$6 returns the ASCII code of the key calling the event.

If event equals 1 or 2, \$6 indicates whether you single or double clicked a reference. If \$6 equals 0, no reference was selected. If \$6 equals 1, a reference was selected. method can be called if you perform one of the following actions:

- Single or double-click a reference
- Alt-click (Windows) or Option-click (Macintosh)
- Shift-Alt-click (Windows) or Shift-Option-click (Macintosh)

On the Macintosh, Option-click and Shift-Option-click typically displays a pop-up menu. In Windows, Alt-click and Shift-Alt-click typically displays a drop-down menu.

If event equals 5, \$6 describes whether or not the area is activated. If \$6 equals 0, the 4D Write area is deactivated. If \$6 equals 1, the 4D Write area is activated.

If event equals 7 and the print job is a mail merge, \$6 indicates the table number for the table used. If the print job is not a mail merge, \$6 equals 0.

If event equals 9, \$6 indicates where margins have been reset in the document. If \$6 equals 0, the margins have been reset in the body. If \$6 equals 1, the margins have been reset in the header. If \$6 equals 2, the margins have been reset in the footer.

To filter characters, you must use method as a function that returns 0 or 1. This enables you to specify characters in the document that 4D Write will ignore.

Initialize \$0 to 1 to make the method trap a particular event. Initialize \$0 to 0 if you do not want to trap for a particular event. For example, if you do not want the character "@" to appear in your document, filter all characters appearing in the document. If the \$6 variable is equal to the ASCII code of the "@" character, you initialize \$0 to 1 and ignore it.

Note: If you filter all characters, operations may slow down considerably since the method will be called for each keystroke.

Example

In the following example, some actions are executed depending on the type of event. Form method:

```
If (Form event=On load)
  WR ON EVENT (Area;0;"ProcName")
  `Call for all key strokes
  WR ON EVENT (Area;5;"ProcName")
  `Check for area status
  DISABLE MENU ITEM(2;1)
  `Disable menu item "Change font"
End if
```

ProcName method:

```
Case of
: ($5=0)
  `Intercept the key strokes
  If ($6=199) | ($6=200)
    `ASCII codes corresponding
    BEEP
    $0:=1
  Else
    `Leave the event to 4D Write
    $0:=0
  End if
: ($5=5)
  `Intercept change in status of area
  If ($6=0)
    `If the area is inactive
    DISABLE MENU ITEM(2;1)
  Else
    ENABLE MENU ITEM(2;1)
  End if
End case
```

WR RGB to color (red; green; blue) → Longint

Parameter	Type		Description
red	Longint	→	Red component Windows: 0 to 255 Mac OS: 0 to 65535
green	Longint	→	Green component Windows: 0 to 255 Mac OS: 0 to 65535
blue	Longint	→	Blue component Windows: 0 to 255 Mac OS: 0 to 65535
Function result	Longint	←	Color

Description

The command `WR RGB to color` returns a compact number that is used by `4D Write to manage colors`. This number represents the three component colors: red, green, and blue. The red, green, and blue parameters are the same values used in your system's color picker.

The following table shows the values for red, green, and blue in commonly used colors.

	Macintosh			Windows		
Color	Red	Green	Blue	Red	Green	Blue
Red	56576	2048	1536	221	8	6
Green	0	32768	4352	0	128	17
Blue	0	0	54272	0	0	212
Cyan	512	43776	59904	2	171	234
Magenta	64512	62208	1280	252	243	5
Yellow	61952	2048	33792	242	8	132

Example

The following example returns a color between two colors:

WR COLOR TO RGB (c1;r1;g1;b1)

WR COLOR TO RGB (c2;r2;g2;b2)

c3:=*WR RGB To color* ((r1+r2)/2;(g1+g2)/2;(b1+b2)/2)

13 WR Obsolete Commands

WR O Area to picture (area)

Parameter	Type		Description
area	Longint	→	4D Write area

Note 6.5: This command was only maintained for compatibility purposes. We recommend using the WR Area to picture command.

Description

The command WR O Area to picture returns a 4th Dimension picture that contains the document in Area. The resulting picture is equivalent to the value that 4D Write automatically stores in a picture field. Use this command to save Area to a field or to retrieve a 4D Write document from an offscreen area.

NOTE: The picture returned by this command is a Picture type. The result of this function must be put in either a 4D picture variable or field.

See Also

No reference.

WR O AUTO SAVE (area)

Parameter	Type		Description
area	Longint	→	4D Write area

Note 6.5: This command was only maintained for compatibility purposes. We recommend using the WR Area to picture command.

Description

The command WR O AUTO SAVE causes the document in area to be saved in a picture field with the same name, if it exists. This command automatically saves a 4D Write area to a picture field. Use WR O AUTO SAVE whenever a 4D Write area is changed procedurally and when the changes will be saved in the corresponding picture field.

See Also

No reference.

WR O CHANGE STYLE (area; delete; add)

Parameter	Type		Description
area	Longint	→	4D Write area
delete	Integer	→	Style to be removed
add	Integer	→	Style to be set

Note 6.5: This command was only maintained for compatibility purposes. We recommend using the WR SET TEXT PROPERTY command.

Description

The command WR O CHANGE STYLE sets the style of the text you select.

delete and add are numbers obtained by adding different style numbers together. add is the style parameter that you want the selected text to have. delete is the style parameter that you do not want in the selected text. Style numbers are presented in the following list.

Style	Value
Plain	0
Bold	1
Italics	2
Underline	4
Outline	8
Shadow	16
Superscript	32
Subscript	64

WR O COMPUTE NOW (area)

Parameter	Type		Description
area	Longint	→	4D Write area

Note 6.5: This command was only maintained for compatibility purposes. We recommend using the WR EXECUTE COMMAND command.

Description

The command WR O COMPUTE NOW recalculates the variable elements (inserted expressions, markers, etc.) of the 4D Write document's area parameter

WR O Count Stylesheet (area) → Integer

Parameter	Type		Description
area	Longint	→	4D Write area
Function result	Integer	←	Number of style sheets available in area

Note 6.5: This command was only maintained for compatibility purposes. We recommend using the WR Count command.

Description

The command WR O Count Stylesheet returns the number of style sheets available in area.

WR O CREATE STYLESHEET (area; styleNum; name; font; size; style; color)

Parameter	Type		Description
area	Longint	→	4D Write area
styleNum	Integer	→	Creation order number of the style sheet
name	String	→	Style sheet title
font	Integer	→	Font chosen
size	Integer	→	Size of chosen font
style	Integer	→	Style used
color	Longint	→	Color used

Note 6.5: This command was only maintained for compatibility purposes. We recommend using the WR Create Stylesheet command.

Description

The command WR O CREATE STYLESHEET creates a new style and inserts the list at the position specified by StyleNum. Define the new style sheet by assigning values to the name, font, size, style, and color parameters.

name is the name of the style sheet and can have a maximum of 32 characters.

font is the number of the font. Use the WR Font name function to obtain the integer corresponding to a font.

size is the point size of the font and must be between 1 and 127. If you specify a font size large than 127, it will be ignored.

style is expressed as the sum of the styles used. See the WR O SET STYLESHEET command in this section for a list of styles and their corresponding codes.

color is a long integer that represents the color of the text. Use the WR RGB to color function to obtain the long integer corresponding to a color. See the WR O GET STYLESHEET command in this section for a list of commonly used colors and their values.

WR O DELETE STYLESHEET (area; styleNum)

Parameter	Type		Description
area	Longint	→	4D Write area
styleNum	Integer	→	Style number in the style sheet

Note 6.5: This command was only maintained for compatibility purposes. We recommend using the WR DELETE STYLESHEET command.

Description

The command WR O DELETE STYLESHEET deletes the style sheet specified by styleNum.

WR O DISPLAY MENUBAR (area; mode)

Parameter	Type		Description
area	Longint	→	4D Write area
mode	Integer	→	1=Hide 0=Show

Note 6.5: This command was only maintained for compatibility purposes. We recommend using the WR SET DOCUMENT PROPERTY command.

Description

The command WR O DISPLAY MENUBAR displays or hides the menu bar in area. If mode equals 1, the menu bar is hidden. If mode equals 0, the menu bar is visible.

See Also

No reference.

WR O DISPLAY RULER (area; mode)

Parameter	Type		Description
area	Longint	→	4D Write area
mode	Integer	→	1=Hide 0=Show

Note 6.5: This command was only maintained for compatibility purposes. We recommend using the WR SET DOCUMENT PROPERTY command.

Description

The command WR O DISPLAY RULER displays or hides the ruler in area. If mode equals 1, the ruler is hidden. If mode equals 0, the ruler is displayed.

See Also

No reference.

WR O DISPLAY SCROLLBARS (area; displayed)

Parameter	Type		Description
area	Longint	→	4D Write area
displayed	Integer	→	0=Scroll bars hidden 1=Scroll bars displayed

Note 6.5: This command was only maintained for compatibility purposes. We recommend using the WR SET DOCUMENT PROPERTY command.

Description

The command WR O DISPLAY SCROLLBARS allows you to display or hide the 4D Write document's scroll bars.

If the value of displayed parameter equals 1, the scroll bars are shown; if it equals 0, the scroll bars are hidden.

WR O DO COMMAND (area; command; modifiers)

Parameter	Type		Description
area	Longint	→	4D Write area
command	Integer	→	Command number
modifiers	Integer	→	Modifier key(s) pressed

Note 6.5:

- This command was only maintained for compatibility purposes. We recommend using the WR EXECUTE COMMAND command.
- Also, the WR O DO COMMAND command does not work with the following commands:
405 (outline style)
702 (subscribe hotlink)
708 (publish hotlink)

Description

The command WR O DO COMMAND executes the menu item specified by command. The menu item will be executed as if you had chosen it from a 4D Write menu. These values will remain the same even if menu items are changed or moved in future versions of 4D Write.

The values for command are listed in Appendix B.

The following modifier keys can be used, expressed as the sum of key numbers:

Key	Value
Command	256
Shift	512
Caps Lock	1024
Option	2048
Control	4096

See Also

No reference.

WR O EXPERT COMMAND (area; command; status)

Parameter	Type		Description
area	Longint	→	4D Write area
command	Integer	→	Command number
status	Integer	→	0=Enabled 1=Disabled

Note 6.5: This command was only maintained for compatibility purposes. We recommend using the WR LOCK COMMAND command.

Description

The command WR O EXPERT COMMAND enables or disables the menu items. If status equals 0, the menu item is enabled. If status equals 1, the menu item is disabled. The WR O EXPERT COMMAND command cannot enable commands disabled by 4D Write. See Appendix B for a listing of command values.

To disable all items in a menu, pass the menu number followed by 00. For example, to disable the File menu, pass 100 for Command. In the case of the Database menu, if you disable all the menu items, the menu is not displayed.

WR O Find (area; criteria{; where{; method}}) → Integer

Parameter	Type		Description
area	Longint	→	4D Write area
criteria	Text	→	String for which to search
where	Integer	→	0=Partial word 1=Whole word
method	Integer	→	0=Not case sensitive 1=Case sensitive
Function result	Integer	←	0 if criteria is not found, 1 if criteria is found

Note 6.5: This command was only maintained for compatibility purposes. We recommend using the WR Find command.

Description

The command WR O Find searches for criteria in area, and if found, selects it. If criteria is not found, WR O Find returns 0. If criteria is found, WR O Find returns 1. WR O Find always begins searching from the position of the last character in the selected text. Because WR O Find selects criteria, its position can be obtained from the WR GET SELECTION command.

The optional where parameter determines whether criteria can match part of a word. If where equals 0, WR O Find selects criteria even if criteria is part of another word. If where equals 1, WR O Find will select criteria only if criteria is found as a whole word, criteria is surrounded by separator characters such as spaces or punctuation marks.

The optional method parameter determines whether the search will be case sensitive. If method equals 0, WR O Find selects criteria regardless of case. If method equals 1, WR O Find will select criteria only if the cases match.

See Also

No reference.

WR O Font name (fontNumber) → Text

Parameter	Type		Description
fontNumber	Integer	→	Font number
Function result	Text	←	Name of the font whose ID is fontNumber

Note 6.5: This command was only maintained for compatibility purposes. We recommend using the 4D commands.

Description

The command WR O Font name returns the name of the font whose ID is fontNumber. The ID is the same value returned by WR O Font number. If fontNumber does not exist, WR Font name returns an empty string.

See Also

No reference.

WR O Font number (fontName) → Integer

Parameter	Type		Description
fontName	Integer	→	Name of the font
Function result	Integer	←	ID for the font named fontName

Note 6.5: This command was only maintained for compatibility purposes. We recommend using the WR Area to picture command.

Description

The command WR O Font number returns the integer ID for the font named fontName. This ID can be used in the WR O SET ATTRIBUTES command. If fontName does not exist, WR O Font number returns 0.

See Also

No reference.

WR O GET ATTRIBUTES (area; font; size; style; color)

Parameter	Type		Description
area	Longint	→	4D write area
font	Integer	←	Receives font number
size	Integer	←	Receives font size
style	Integer	←	Receives font style
color	Integer	←	Receives font color

Note 6.5: This command was only maintained for compatibility purposes. We recommend using the WR Get text property command.

Description

The command WR O GET ATTRIBUTES returns into the variables font, size, style, and color the attributes of the selected text in Area. If the selected text in area contains more than one font, size, style, or color, WR O GET ATTRIBUTES returns -1 for that attribute.

font is the ID of the font in your system. This is the same value that is returned by WR Font number. You can use the WR Font name function to determine the name of the font.

size is the size in points of the selected text.

style is a composite number that results from the addition of several style numbers. Style numbers are shown in the following list.

Style	Value
Plain	0
Bold	1
Italic	2
Underline	4
Outlined	8
Shadowed	16
Superscript	32
Subscript	64

The styles **Outlined** and **Shadowed** do not exist in Windows. The values returned for these attributes depend on the platform on which they are used in a multi-platform database.

color is a long integer that represents the color of the text. This number can be used in the `WR O SET ATTRIBUTES` command to set other text to the same color.

See Also

No reference.

WR O GET MARGINS (area; left; indent; right)

Parameter	Type		Description
area	Longint	→	4D Write area
left	Integer	←	Receives left margin in points
indent	Integer	←	Receives indent in points
right	Integer	←	Receives right margin in points

Note 6.5: This command was only maintained for compatibility purposes. We recommend using the WR Get text property command.

Description

The command WR O GET MARGINS returns in the left, indent, and right parameters the margin settings of the currently selected paragraph in area. If more than one paragraph is selected, this command returns values for the paragraph where the selection begins.

See Also

No reference.

The styles **Outlined** and **Shadowed** do not exist in Windows. The values returned for these attributes depend on the platform on which they are used in a multi-platform database.

color is a long integer that represents the color of the text. This number can be used in the `WR O SET ATTRIBUTES` command to set other text to the same color.

See Also

No reference.

WR O Get pack options (selector) → Integer

Parameter	Type		Description
selector	Integer	→	Options for templates, menus, or rulers 1=Template saving location 2=Template loading location 3=Menus 4=Rulers 5=Frame 6=Page View 7=reformat Message Suppression
Function result	Integer	←	State of an option

Note 6.5: This command was only maintained for compatibility purposes. We recommend using the WR Document property command.

Description

The command WR O Get pack options returns the state of an option set using WR SET PACK OPTIONS.

If selector equals 1, WR O Get pack options returns the location to which your templates are saved. If the value returned equals 1, the templates are saved on the server. If the value returned equals 0, the templates are saved on the client.

If selector equals 2, WR O Get pack options returns the location from which your templates are loaded. If the value returned equals 1, the templates are loaded from the server. If the value returned equals 0, the templates are loaded from the client.

If selector equals 3, WR O Get pack options returns whether menus are displayed or hidden. If the value returned equals 1, menus are displayed. If the value returned equals 0, menus are hidden.

If selector equals 4, WR O Get pack options returns whether rulers are displayed or hidden. If the value returned equals 1, rulers are displayed. If the value returned equals 0, rulers are hidden.

If Selector equals 5, you can specify whether the frame around the 4D Write area should be displayed or hidden. If value equals 1, the frame is displayed. If value equals 0, the frame is hidden. By default, the frame is displayed (value equals 1).

If selector equals 6, you can specify whether the 4D Write area should be displayed using

WR O Get page (area; position) → Integer

Parameter	Type		Description
area	Longint	→	4D Write area
position	Longint	→	Position of a character in area
Function result position	Integer	←	Page of the character position passed in

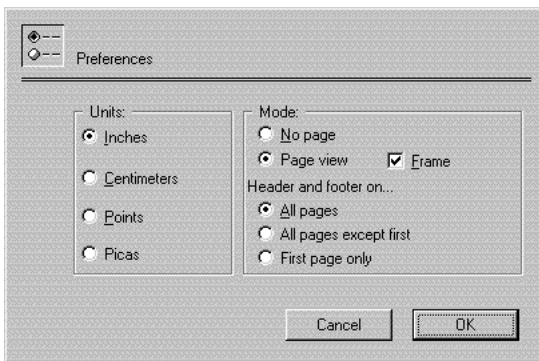
Note 6.5: This command was only maintained for compatibility purposes. We recommend using the WR GET CURSOR INFO command.

Description

The command WR O Get page returns the page of the character position passed in the position parameter. For example, if the 100th character is on page 3, the value in Result in the following line of code will be 3:

```
Result:=WR O Get page (Area;100)
```

This function is useful only after you select the Preferences menu item from the 4D Write Edit menu and set the document to View Page mode. You can also set document mode with the WR SET PREFERENCES command or check it with WR GET PREFERENCES.



WR O GET PICTURE (area; height; width; topLeft)

Parameter	Type		Description
area	Longint	→	4D Write area
height	Integer	←	Receives height of the picture
width	Integer	←	Receives width of the picture
topLeft	Integer	←	Receives position of the picture

Note 6.5: This command was only maintained for compatibility purposes. We recommend using the WR GET PICTURE SIZE command.

Description

The command WR O GET PICTURE returns information for a selected picture in the height, width, and topLeft variables.

height is the height of the picture measured in points. width is the width of the picture also measured in points. topLeft is the horizontal coordinate of the top-left corner of the picture. Distance is calculated from the ruler origin and is expressed in points.

WR O GET PREFERENCES (area; mode; titlePage; units)

Parameter	Type		Description
area	Longint	→	4D Write area
mode	Integer	←	Receives page mode code
titlePage	Integer	←	Receives title page code
units	Integer	←	Receives unit code

Note 6.5: This command was only maintained for compatibility purposes. We recommend using the WR Get document property command.

Description

The command WR O GET PREFERENCES returns in Area the current settings of the document. These settings can be set by either WR O SET PREFERENCES or by selecting the Preferences menu item from the 4D Write Edit menu.

The mode, titlePage, and units parameters can have the following values:

Parameter	Value	Description
<i>Mode</i>	0	Document in No Page mode
	1	Document in Page View mode without frame
	2	Document in Page View mode with frame
<i>TitlePage</i>	0	Header and footer on all pages
	1	Header and footer on first page only
	2	Header and footer on all pages except first page
<i>Units</i>	0	Ruler defined in inches
	1	Ruler defined in centimeters
	2	Ruler defined in points
	3	Ruler is defined in picas





WR O GET RULER (area; leading; alignment)

Parameter	Type		Description
area	Longint	→	4D Write area
leading	Integer	←	Receives code for spacing between lines
alignment	Integer	←	Receives alignment code




Note 6.5: This command was only maintained for compatibility purposes. We recommend using the WR Get text property command.

Description

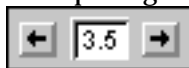
The command WR O GET RULER places into the leading and alignment variables the values you chose for line spacing and alignment for a selection. alignment can have one of four values.

Icon				
Alignment	0	1	2	3

By clicking on one of the following icons, you can set the leading to single, one and one-half, or double spacing.

Icon			
Line Spacing	1	1.5	2

You can specify leading greater than double line spacing by clicking on the left and right line spacing arrows.



leading is an integer value between 0 and 17 that corresponds to one of the 18 possible line spacing positions. For example, a value of 0 results in leading of a single line while a value of 9 yields leading of five and one-half line spaces.

Value	Line Spacing
0	1.0
1	1.5
2	2.0
3	2.5
4	3.0
5	3.5
6	4.0
7	4.5
8	5.0

Value	Line Spacing
9	5.5
10	6.0
11	6.5
12	7.0
13	7.5
14	8.0
15	8.5
16	9.0
17	9.5

See Also

No reference.

WR O Get ScrollBars (area) → Longint

Parameter	Type		Description
area	Longint	→	4D write area
Function result	Longint	←	Scroll bar status

Note 6.5: This command was only maintained for compatibility purposes. We recommend using the WR Get document property command.

Description

The command WR O Get ScrollBars returns a Long integer that describes the status of the 4D Write document's scroll bars.

- If the function returns 1, the scroll bars are displayed.
- If the function returns 0, the scroll bars are hidden.

WR O GET STYLESHEET (area; styleNum; name; font; size; style; color)

Parameter	Type		Description
area	Longint	→	4D Write area
styleNum	Integer	→	Style sheet number
name	String	←	Receives style sheet title
font	Integer	←	Receives font chosen
size	Integer	←	Receives size of chosen font
style	Integer	←	Receives styles used
color	Longint	←	Receives color used

Note 6.5: This command was only maintained for compatibility purposes. We recommend using the WR GET STYLESHEET INFO command.

Description

The command WR O GET STYLESHEET returns in the name, font, size, style, and color parameters the values corresponding to the style sheet specified by StyleNum.

name contains the title of the style sheet and can be a maximum of 32 characters.

font returns the number of the font used in the specified style sheet. Use the WR O Font name function to obtain the name of the font with an ID of font.

size returns the size of the font used in the specified style sheet.

style returns the sum of the styles used. The list of styles and their corresponding codes is presented in the following table.

Style	Value
Plain	0
Bold	1
Italics	2
Underline	4
Outline	8
Shadow	16
Superscript	32
Subscript	64

color returns a long integer that represents the color of the text. Use the WR RGB to color function to obtain the long integer corresponding to a color. The following is a list of commonly used colors and their corresponding values.

	Macintosh			Windows		
Color	Red	Green	Blue	Red	Green	Blue
Black	0	0	0	0	0	0
Red	56576	2048	1536	221	8	6
Green	0	32768	4352	0	128	17
Blue	0	0	54272	0	0	212
Cyan	512	43776	59904	2	171	234
Magenta	64512	62208	1280	252	243	5
Yellow	61952	2048	33792	242	8	132

See Also

No reference.

WR O GET TABS (area; tabs)

Parameter	Type	Description
area	Longint →	4D Write area
tabs	2D Integer array ←	Receives array of tabs

Note 6.5: This command was only maintained for compatibility purposes. We recommend using the WR GET TAB command.

Description

The command WR O GET TABS returns in tabs the tab stops of the selected paragraph in area. If more than one paragraph is selected, WR O GET TABS returns values for the paragraph where the selection begins.

Tabs must be a 3 x 10, two-dimensional integer array. WR O GET TABS will not resize tabs. After you issue WR O GET TABS, the three arrays in tabs will contain the following:

- Location of the tab stops in points
- Types of justification
- Leader (fill) characters

The number of tab stops will be returned in the zero element of tabs{1}.

tabs{1} contains the distance in points from the left side of the page to each tab.

tabs{2} contains a number that describes the justification of each tab. Possible values for elements in tabs{2} are as follows.

Justification	Value
Left	0
Right	1
Decimal	2
Centered	3

tabs{3} contains a number that describes the leader character of each tab. Possible values for elements in tabs{3} are as follows.

Justification	Value
Left	0
Right	1
Decimal	2
Centered	3

Note: The array used in WR O GET TABS must be a 3 X 10, two-dimensional integer array and cannot be resized by the command. You can check the number of tabs in the paragraph by testing the element tabs{1}{0}.

See Also

No reference.

WR O INSERT HYPHEN (area)

Parameter	Type		Description
area	Longint	→	4D Write area

Note 6.5: This command was only maintained for compatibility purposes. We recommend using the WR EXECUTE COMMAND command.

Description

The command WR O INSERT HYPHEN enables you to insert a hyphen at the insertion point. The insertion point can be located before or after the hyphen. The inserted hyphen is then visible only if the word is at the end of a line.

If a hyphen or punctuation (separator) character is already present at the insertion point, WR O INSERT HYPHEN does nothing and returns an error.

WR O INSERT PICTURE (area; picture)

Parameter	Type		Description
area	Longint	→	4D Write area
picture	Picture	→	Picture to insert

Note 6.5: This command was only maintained for compatibility purposes. We recommend using the WR INSERT PICTURE command.

Description

The command WR O INSERT PICTURE inserts a picture at the point of selection.

WR O Is Hyphen (area) → Longint

Parameter	Type		Description
area	Longint	→	4D Write area
Function result	Longint	←	1, there is a hyphen at the insertion point 0, there is no hyphen at the insertion point

Note 6.5: This command was only maintained for compatibility purposes. We recommend using the WR SELECT command.

Description

The command WR O Is Hyphen returns a Long integer that signifies the presence or absence of hyphens at the insertion point. The insertion point can be located before or after the hyphen.

- If the function returns 1, there is a hyphen at the insertion point.
- If the function returns 0, there is no hyphen at the insertion point.

WR O LINE SPACING (area; leading)

Parameter	Type		Description
area	Longint	→	4D Write area
leading	Integer	→	Spacing between lines [0...17]

Note 6.5: This command was only maintained for compatibility purposes. We recommend using the WR SET TEXT PROPERTY command.

Description

The command WR O LINE SPACING sets the line spacing of the selected paragraphs in area to the value described by leading. leading is an integer value between 0 and 17.

See Also

No reference.

WR O MENU STATUS (area; command; checked; status{; name})

Parameter	Type		Description
area	Longint	→	4D Write area
command	Integer	→	Command number
checked	Integer	←	Receives checked status 0 = Not checked 7 or 195 = Checked
status	Integer	←	Receives active status 0 = Inactive 1 = Active
name	Text or String	←	Receives name of menu item

Note 6.5: This command was only maintained for compatibility purposes. We recommend using the WR GET COMMAND INFO command.

Description

The command WR O MENU STATUS determines if the menu item represented by Command is checked or active. See Appendix B for a listing of Command values. After a call to WR O MENU STATUS, the checked and active variables contain values that describe the state of Command in area. If active equals 1, the menu item is enabled. If active equals 0, the item is disabled. If checked equals 0, the item is not checked. If checked does not equal 0, the item is checked with the character described by the ASCII code in checked.

The optional name parameter must be a text or string variable and will return the text of the menu item.

See Also

No reference.

WR O ModuleInfo → Longint

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

This command does not require any parameters

Function result	Longint	←	4D Write version
-----------------	---------	---	------------------

Note 6.5: This command was only maintained for compatibility purposes.

Description

The command WR O ModuleInfo returns a Long integer that represents the type of 4D Write version used. The possible values are:

Value	4D Write version
0	License number not entered, plug-in cannot be used
1	Licensed version
2	Demo version
3	Time-limited version

WR O MOVE PICTURE (area; topLeft)

Parameter	Type		Description
area	Longint	→	4D Write area
topLeft	Integer	→	New picture position

Note 6.5: This command was only maintained for compatibility purposes. We recommend using the alignment settings.

Description

The command WR O MOVE PICTURE moves a selected picture laterally. The selection should not consist of anything except the picture.

topLeft is the horizontal coordinate of the top-left corner of the picture. Distance is calculated from the ruler origin where this value is expressed in points.

WR O ON MENU (area; method)

Parameter	Type		Description
area	Longint	→	4D Write area
method	String	→	Name of method

Note 6.5: This command was only maintained for compatibility purposes. We recommend using the WR ON COMMAND command.

Description

The command WR O ON MENU installs Method as the method for managing 4D Write menus. If method is an empty string, no method will be called. After a menu method is installed, all menu actions must be managed by method.

If area equals 0, WR O ON MENU is applied to all 4D Write areas until the database is closed. Method receives the following local variables when the method executes:

Variable	Description
\$1	A long integer that represents the 4D Write area where the menu was selected
\$2	A long integer that contains the command number for the selected menu item (see Appendix B)
\$3	A long integer that describes the modifier keys depressed at the time the menu item was selected.

Explicitly type the \$1, \$2, and \$3 variables using compiler directives. The following are the modifier keys used, expressed as the sum of key numbers:

Key	Value
Command	256
Shift	512
Caps Lock	1024
Option	2048
Control	4096

WR O OPTIONS (area; message; confirm; printMode; format; undo)

Parameter	Type		Description
area	Longint	→	4D Write area
message	String	→	Message for empty references
confirm	Integer	→	0=No confirmation 1=Confirm -1=No change
printMode	Integer	→	0=Variable length 1=Fixed size -1=No changes
format	Integer	→	0=4D Write and first page as PICT 1=4D Write -1=No change
undo	Integer	→	1=Disabled 0=Enabled -1=No change

Note 6.5: This command was only maintained for compatibility purposes. We recommend using the WR Area to picture command.

Description

The command WR O OPTIONS sets several functional parameters for Area. If 0 is specified for area instead of an area name, the WR OPTIONS command is applied to all new areas you create, but only affects the message parameter. Areas that already exist will not be affected by this command.

Message is the text that appears when a reference to a 4th Dimension field or expression returns nothing. The text you assign to message can be used until the user quits from 4th Dimension or executes the WR O OPTIONS command again. In either case, message reverts to the default of “«»”. message also appears as “«»” if you did not assign any text to this parameter.

The message “This area has been modified. Do you want to save it?” appears when any command has affected the area since its creation. You can disable the display of this message using the WR O OPTIONS command.

Confirm controls whether or not 4D Write will display confirmation dialog boxes when a 4D Write area is closed but not saved. If confirm equals 0, no confirmation dialog box appears. If confirm equals 1, 4D Write displays the confirmation dialog box. If confirm equals -1, the current setting is not changed.

printMode controls how area is treated when a form that contains area is printed. If printMode equals 0, the text in area is truncated to the size of area. This is similar to a fixed frame included form. If printMode equals 1, area expands to print all of the text, even across pages. This is similar to a variable frame included form. If printMode equals -1, the current setting is not changed.

format specifies the format in which the document will be saved. If format equals 0, area is saved in the 4D Write format and its first page is saved in PICT format. If format equals 1, area can be saved only in the 4D Write format. If format equals -1, the current setting is not changed.

undo enables or disables the Undo menu item in the Edit menu. Use this parameter in offscreen areas; it saves memory and increases 4D Write speed. If undo equals 0, the Undo menu item is enabled. If undo equals 1, the Undo menu item is disabled. If undo equals -1, the setting is not changed.

See Also

No reference.

WR O Page number {(area)} → Integer

Parameter	Type		Description
area	Longint	→	4D Write area
Function result	Integer	←	Number of the page currently being printed

Note 6.5: This command was only maintained for compatibility purposes. We recommend using the WR INSERT PAGE NUMBER command.

Description

The command WR O Page number returns the number of the page that is currently being printed. This function can only be used in the header or footer area of a 4D Write document or in a method called from within the header or footer of a 4D Write document. If WR O Page number is used directly in a 4D Write area, the area name may be omitted.

See Also

No reference.

WR O PICTURE TO AREA (area; picture)

Parameter	Type		Description
area	Longint	→	4D Write area
picture	Picture	→	Picture to open

Note 6.5: This command was only maintained for compatibility purposes. We recommend using the WR PICTURE TO AREA command.

Description

The command WR O PICTURE TO AREA assigns to area the 4D Write document in picture. Use this command to manually load Picture from a field or to place a 4D Write document in an offscreen area.

See Also

No reference.

WR O Picture to offscreen area (picture) → Longint

Parameter	Type		Description
picture	Picture	→	Picture to open
Function result	Longint	←	Area reference number

Note 6.5: This command was only maintained for compatibility purposes. We recommend using the WR Picture to area command.

Description

The command WR O Picture to offscreen area places the document contained in picture into a 4D Write area that is invisible to the user (an offscreen area) and returns a value that can be used to access the new area.

The value returned by WR O Picture to offscreen area can be used in any 4D Write command that requires a 4D Write area. This command is similar to combining WR New offscreen area and WR O PICTURE TO AREA.

NOTE: You should always use WR DELETE OFFSCREEN AREA when you have completed your operation on the offscreen area.

See Also

No reference.

WR O REMOVE HYPHEN (area)

Parameter	Type		Description
area	Longint	→	4D Write area

Note 6.5: This command was only maintained for compatibility purposes. We recommend using the WR SELECT command.

Description

The command WR O REMOVE HYPHEN enables you to remove the hyphen at the insertion point. The insertion point can be located before or after the hyphen.

If a hyphen is not present at the insertion point, WR O REMOVE HYPHEN does nothing and returns an error.

WR O Replace (area; old; new; where; method; mode) → Longint

Parameter	Type		Description
area	Longint	→	4D Write area
old	Text	→	String to search for
new	Text	→	String to replace it with
where	Integer	→	0=Partial word 1=Whole word
method	Integer	→	0=Not case sensitive 1=Whole word
mode	Integer	→	0=Replace next 1=Replace all
Function result	Longint	←	Number of replacements

Note 6.5: This command was only maintained for compatibility purposes. We recommend using the WR Replace command.

Description

The command WR O Replace searches for occurrences of old in area and replaces them with new. This function also returns the number of replacements. WR O Replace begins replacing from the position of the last character in the selected text.

Where determines whether old can match only whole words. If where equals 0, WR O Replace replaces old if old is part of a word. If where equals 1, WR O Replace replaces old only if old is a whole word. A whole word is a word surrounded by separator characters such as spaces or punctuation marks.

method determines whether or not the replacement will be case sensitive. If method equals 0, WR O Replace replaces old regardless of case. If method equals 1, WR O Replace replaces old only if the cases match. mode determines how many replacements will be made.

If mode equals 0, only the next occurrence of old is replaced. If mode equals 1, all occurrences of old are replaced with new, regardless of the position of the selected text.

See Also

No reference.

WR O RESIZE PICTURE (area; newHeight; newWidth)

Parameter	Type		Description
area	Longint	→	4D Write area
newHeight	Integer	→	New height of the picture
newWidth	Integer	→	New width of the picture

Note 6.5: This command was only maintained for compatibility purposes. We recommend using the WR SET PICTURE SIZE command.

Description

The command WR O RESIZE PICTURE modifies the size of a selected picture in the 4D Write area. The selection should consist only of the selected picture. If no picture is selected, then error 1034 is generated.

newHeight is the new height of the picture in points. newWidth is the new width of the picture also in points.

There are 72 dots per inch (dpi) on a typical Macintosh monitor. If your monitor has a resolution of 82 dpi or more, use a case statement to determine the number of dots per inch to use.

If newHeight or newWidth are less than 0, the size of the picture is not modified. If newHeight or newWidth are equal to 0, the picture is removed.

WR O Save to picture (area) → Picture

Parameter	Type		Description
area	Longint	→	4D Write area
Function result	Picture	←	4th Dimension picture

Note 6.5: This command was only maintained for compatibility purposes. We recommend using the WR Area to picture command.

Description

The command WR O Save to picture returns a 4th Dimension picture that contains the document in area. The resulting picture is equivalent to the value that is automatically stored in a picture field by 4D Write. This command is useful when Area is saved manually to a field or on disk. You can also use this command to retrieve a 4D Write document from an offscreen area.

The difference between WR O Area to picture and WR O Save to picture is subtle. WR O Save to picture sets the document-saved flag for Area to TRUE while WR O Area to picture does not. Consequently, closing the window or form that contains area after a call to WR O Save to picture will not cause 4D Write to prompt you to save the document.

NOTE: The result of this function must be put in either a 4th Dimension picture field or a variable.

See Also

No reference.

WR O SET ATTRIBUTES (area; font; size; style; Color)

Parameter	Type		Description
area	Longint	→	4D Write area
font	Integer	→	Font number
size	Integer	→	Font size
style	Integer	→	Font style
Color	Integer	→	Font color

Note 6.5: This command was only maintained for compatibility purposes. We recommend using the WR SET TEXT PROPERTY command.

Description

The command WR O SET ATTRIBUTES sets the attributes of the selected text in area to the values described by font, size, style, and color. If you want to leave one of the attributes unchanged, pass -1 for that attribute.

font is the ID of the font in your system. This value can be obtained by the WR O Font number function.

size is the size of the font expressed in points.

style is the style of the font expressed as a sum of style numbers. The following is a list of style numbers:

Style	Value
Plain	0
Bold	1
Italic	2
Underline	4
Outlined	8
Shadowed	16
Superscript	32
Subscript	64

When styles are set for a selection of text, each style is applied separately and the selection is affected by the following conditions:

- A style is in effect for the entire selection, such as bold, italic, and outlined. WR O SET ATTRIBUTES applies that style number and that style is deselected throughout the selection.

- A style is in effect for only part of the selection or not in effect at all. WR O SET ATTRIBUTES applies that style throughout the selection. To determine if a style is in effect for the entire selection, test the ASCII value of the check mark character for the appropriate menu item. You can determine if a particular style is used or not in the selected text if one of the following is true:

- A style menu item has no check mark character and is therefore not in effect anywhere in the selection.
- A style menu item is checked with the character ◊ (ASCII 215) and is therefore in effect for part of the selection.
- A style menu item is checked with the character ⇒ (ASCII 195) and is therefore in effect for the entire selection.

Note: The styles Outlined and Shadowed do not exist in Windows. The values returned for these attributes depend on the platform on which they are used in a multi-platform database.

color is a long integer that represents the color of the text. The following is a list of commonly used colors and their values:

Color	Macintosh			Windows		
	Red	Green	Blue	Red	Green	Blue
Black	0	0	0	0	0	0
Red	56576	2048	1536	221	8	6
Green	0	32768	4352	0	128	17
Blue	0	0	54272	0	0	212
Cyan	512	43776	59904	2	171	234
Magenta	64512	62208	1280	252	243	5
Yellow	61952	2048	33792	242	8	132

See Also

No reference.

WR O SET MARGINS (area; left; indent; right)

Parameter	Type		Description
area	Longint	→	4D Write area
left	Integer	→	Left margin in points
indent	Integer	→	Indent in points
right	Integer	→	Right margin in points

Note 6.5: This command was only maintained for compatibility purposes. We recommend using the WR SET TEXT PROPERTY command.

Description

The command WR O SET MARGINS sets the margins of the selected paragraphs in area to the values given by left, indent, and right. The values of left, indent, and right are expressed in points from the left side of the document.

To leave one of the margins unchanged, use -1 for that parameter.

See Also

No reference.

 WR O SET PACK OPTIONS (selector; value)

Parameter	Type		Description
selector	Integer	→	Options for templates, menus, or rulers 1=Template saving location 2=Template loading location 3=Menus4=Rulers 5=Frame 6=Page View 7=reformat Message Suppression
value	Integer	→	For templates 1=server 0=client For menus and rulers 1=displayed 0=hidden

Note 6.5: This command was only maintained for compatibility purposes. We recommend using the WR SET DOCUMENT PROPERTY command.

Description

The command WR O SET PACK OPTIONS is used to set options for templates, menus, or rulers. All options set by WR O SET PACK OPTIONS are temporary. To ensure that the options are always in effect, set the options in the Startup method.

If selector equals 1, you can specify where your templates will be saved. If value equals 1, the templates will be saved on the server. If value equals 0, the templates will be saved on the client. By default, templates are saved on the server (value equals 1).

If selector equals 2, you can specify the location from which your templates will be loaded. If value equals 1, the templates will be loaded from the server. If value equals 0, the templates will be loaded from the client. By default, templates are loaded from the server (value equals 1).

If selector equals 3, you can specify whether the menus should be displayed or hidden. If value equals 1, the menus are displayed.

If value equals 0, the menus are hidden. By default, menus are displayed (value equals 1).

If selector equals 4, you can specify whether the rulers should be displayed or hidden. If value equals 1, the rulers are displayed. If value equals 0, the rulers are hidden. By default, the rulers are displayed (value equals 1).

If selector equals 5, you can specify whether the frame around the 4D Write area should be displayed or hidden. If value equals 1, the frame is displayed. If value equals 0, the frame is hidden. By default, the frame is displayed (value equals 1).

If selector equals 6, you can specify whether the 4D Write area is displayed using the Page View mode. If value equals 1, the Page View mode is used. If value equals 0, the No Page View mode is used. By default, the Page View mode is used (value equals 1).

If selector equals 7, you can specify whether 4D Write displays a warning message before reformatting a document when the page setup has changed. If value equals 1, the warning is displayed. If value equals 0, no warning is displayed. By default, the warning is displayed (value equals 1).

WR O SET PACK OPTIONS is especially useful for hiding menus or rulers in an external window. By using the command before creating an external window with the External window function, you can hide the menus or rulers before the window is displayed. If you use the WR O DISPLAY RULER or WR O DISPLAY MENUBAR commands instead, 4th Dimension hides the menus or rulers after the window is displayed, causing an awkward redrawing of the window.

See Also

No reference.

WR O SET PREFERENCES (area; mode; firstPage; units)

Parameter	Type		Description
area	Longint	→	4D Write area
mode	Integer	→	0=No page 1=Page view 2=Page view with frame -1=No changes
firstPage	Integer	→	0=Normal 1=Header and footer on all but first page 2=Header and footer on first page only -1=No changes
units	Integer	→	0=Inches 1=Centimeters 2=Points 3=Picas -1=No changes

Note 6.5: This command was only maintained for compatibility purposes. We recommend using the WR SET DOCUMENT PROPERTY command.

Description

The command WR O SET PREFERENCES defines the options you can use to display the document contained in Area.

To leave an attribute unchanged for any one of the parameters, pass -1 for that attribute.

WR O SET STYLESHEET (area; styleNum; name; font; size; style; color)

Parameter	Type		Description
area	Longint	→	4D Write area
styleNum	Integer	→	Style sheet number
name	String	→	Style sheet title
font	Integer	→	Font chosen
size	Integer	→	Size of chosen font, 0 = No change
style	Integer	→	Style sheet used, 0 = No change
color	Longint	→	Color used

Note 6.5: This command was only maintained for compatibility purposes. We recommend using the WR SET STYLESHEET INFO command.

Description

The command WR O SET STYLESHEET modifies the parameters name, font, size, style, and color of the style sheet specified by styleNum.

name is the title of the style sheet and can be a maximum of 32 characters.

font is the number of the font used in the specified style sheet. Use the WR Font number function to obtain the number of the font in question.

size is the point size of the font used in the specified style sheet. The value must be between 1 and 127. If you specify a font size large than 127, it will be ignored.

style is expressed as a sum of the styles used. The following is a list of styles and their corresponding codes.

Style	Value
Plain	0
Bold	1
Italic	2
Underline	4
Outline	8
Shadow	16
Superscript	32
Subscript	64

Note: To leave a font attribute unchanged, pass -1 for that attribute.

color is a long integer that represents the color of the text. Use the WR RGB to color function to obtain the long integer corresponding to a color. See the WR O GET STYLESHEET command in this section for a list of commonly used colors and their values.

WR O SET TABS (area; old; new; leader; justification)

Parameter	Type		Description
area	Longint	→	4D Write area
old	Integer	→	Old tab in points
new	Integer	→	New tab in points
leader	Integer	→	Leader character [0...3]
justification	Integer	→	Justification [0...3]

Note 6.5: This command was only maintained for compatibility purposes. We recommend using the WR SET TAB command.

Description

The command WR O SET TABS adds, modifies, or deletes tab stops for the selected paragraphs in area. This command moves the tab stop described by old to the tab stop described by new.

If old equals -1, or if there is no tab stop at old, WR O SET TABS creates a new tab at the position specified by new. If new equals -1, WR O SET TABS deletes the tab at the position specified by old. If a tab is created or moved, it is modified by the leader and justification parameters.

leader specifies the fill character to use for the tab. The following are the possible values for leader:

Leader	Value
None	0
Dots	1
Dashes	2
Underlines	3

justification specifies the type of justification the tab will have. The following are the possible values for justification:

Leader	Value
None	0
Dots	1
Dashes	2
Underlines	3

WR O STATISTICS (area; characters; paragraphs; objects; hotlinks; modified; pages)

Parameter	Type		Description
area	Longint	→	4D Write area
characters	Longint	←	Receives character count
paragraphs	Integer	←	Receives paragraph count
objects	Integer	←	Receives object count
hotlinks	Integer	←	Receives hot links count
modified	Integer	←	Receives modified status 0=Document not modified 1=Document modified
pages	Integer	←	Receives number of pages

Note 6.5: This command was only maintained for compatibility purposes. We recommend using the WR Count command.

Description

The command WR O STATISTICS returns the number of characters, paragraphs, and hot links in area along with 4th Dimension objects. This command also returns the number of pages in area.

After WR O STATISTICS is called, each parameter will contain a value as described in the following table:

Parameter	Value Returned
<i>Characters</i>	Total number of characters in <i>Area</i>
<i>Paragraphs</i>	Total number of paragraphs in <i>Area</i>
<i>Objects</i>	Total number of 4 th Dimension objects (fields, expressions, and hot links) in <i>Area</i> .
<i>Hotlinks</i>	Total number of hot links subscribed to in <i>Area</i> .
<i>Modified</i>	0 = Document not modified 1 = Document modified
<i>Pages</i>	Total number of pages in the document

WR O STRUCTURE ACCESS (area; mode)

Parameter	Type		Description
area	Longint	→	4D Write area
mode	Integer	→	0=Allow access 1=Restrict access

Note 6.5: This command was only maintained for compatibility purposes. We recommend using the WR LOCK COMMAND command.

Description

The command WR O STRUCTURE ACCESS controls access to 4th Dimension fields. If mode equals 0, 4D Write enables the Insert Field menu item. If mode equals 1, 4D Write disables the Insert Field menu item in the Database menu and cancels the keyboard shortcut for accessing the table-field pop-up menu. This command enables the Designer to restrict access of specific fields in the database.

To make the table-field pop-up menu available:

- In Windows, hold down the Alt or Shift-Alt keys while pressing the mouse button.
- On the Macintosh, hold down the Option or Shift-Option keys while pressing the mouse button.

See Also

No reference.

WR O TEXT ALIGNMENT (area; alignment)

Parameter	Type		Description
area	Longint	→	4D Write area
alignment	Integer	→	0=Left 1=Center 2=Right 3=Full

Note 6.5: This command was only maintained for compatibility purposes. We recommend using the WR SET TEXT PROPERTY command.

Description

The command WR O TEXT ALIGNMENT sets the alignment of the selected paragraphs in area to the value described by alignment. The following are the possible values for alignment:

Justification§	Value§
Left§	0§
Center§	1§
Right§	2§
Full§	3§

See Also

No reference.

14 Appendixes

Special Keys

In addition to scrolling, some keyboards offer keys that allow you to quickly find locations in a document.

Key	Explanation
Home	Displays the beginning of the document
End	Displays the end of the document
Page Up	Scrolls one page up (does not modify the current selection)
Page Down	Scrolls one page down (does not modify the current selection)
Ctrl (or Command) + left arrow	Moves the insertion point to the beginning of the current word or to the beginning of the previous word if the insertion point was already at the beginning of the current word.
Ctrl (or Command) + right arrow	Moves the insertion point to the end of the current word or to the end of the following word if the insertion point was already at the end of the current word
Ctrl + Delete (Ctrl/Command + Forward Delete on Macintosh)	Deletes the next word located on the right of the cursor.
Ctrl + Backspace (Ctrl/Command + Forward Delete on Macintosh)	Deletes the next word located on the left of the cursor
Shift (in combination with any of the above keys to move the insertion point or view)	Extends the current selection

Click Combinations

In addition to scrolling, 4D Write allows you to use the following click combination.

Combination	Explanation
Single click	Moves the insertion point, deselecting any text that was selected
Double-click	Selects the word that was double-clicked and the following space (if any)
Triple-click	Selects the paragraph
Click in left margin	Selects the line next to the click.
Double-click in left margin	Selects the paragraph next to the click.
Shift + Click	Extends the current selection to the location of the click.

The following table lists the command value for each menu item. These numbers will remain the same, even if menu items are modified or moved in future versions of 4D Write. For more information, refer to the description of the WR EXECUTE COMMAND command. The following codes can also be used by the WR ON COMMAND and the WR GET COMMAND INFO commands. When using those commands you can either pass the menu item number or the constant.

Menu	Toolbar But.	Command	Reference
File	Yes	New	101
	Yes	Open	102
	Yes	Save	103
	No	Save as...	104
	No	Save as Template	110
	No	Preferences...	105
	No	Page SetUp...	106
	Yes	Print Preview	107
	Yes	Print...	108
	No	Print Merge...	109
	No	Goto Full Window/Return to Form	20
Edit	Yes	Undo Fonction	1
	Yes	Redo Fonction	2
	Yes	Cut	3
	Yes	Copy	4
	Yes	Paste	5
	No	Clear	6
	No	Select All	7
	Yes	Find...	208
	No	Find Next	209
	No	Replace...	210
	No	Replace next	211
	No	Change Case	220
	No	/ lower case	221
	No	/ UPPER CASE	222
	No	/ Title Case	223
	No	/ tOGGLE cASE	224
	No	Show Selection	309
No	Goto Page...	807	

View	No	Normal	302
	No	Page	303
	No	Toolbars	330
	No	/ View Standard Toolbar	331
	No	/ View Format Toolbar	332
	No	/ View Style Toolbar	333
	No	/ View Borders Toolbar	334
	No	View Ruler	311
	No	View Header	312
	No	View Footer	313
	Yes	View References	314
	No	View Pictures	315
	Yes	View Invisibles	316
	No	View Frames	317
	No	View Horizontal Scrollbar	318
	No	View Vertical Scrollbar	319
	No	View MenuBar	310
	No	View Status Bar	320
	Insert	No	Insert Date and Time...
Yes		Insert Current Hour	411
Yes		Insert Current Date	412
No		Insert Page Number...	402
No		Insert Special Character...	409
No		Insert Soft Hyphen	404
No		Insert Non Breaking Space	405
No		Insert Column Break	410
No		Insert Page Break	406
No	Insert 4D Expression...	407	
Style	No	Plain	501
	Yes	Bold	502
	Yes	Italic	503
	No	Shadow	504
	No	StrikeThrough	505
	No	Underline	
	No	/ No Underline	521
	No	/ Single Underline	522
	No	/ Word Underline	523
	No	/ Double Underline	524
	No	/ Hatched Underline	525
	Yes	Button Underline	530
	No	Superscript	506
	No	Subscript	507
	No	Capitals	508
No	Small Capitals	509	

Colors

Text

- / 602 Black Text
- / 603 Red Text
- / 604 Orange Text
- / 605 Yellow Text
- / 606 Green Text
- / 607 Blue Text
- / 608 Violet Text
- / 609 White
- / 610 LightGrey Text
- / 611 MediumGrey Text
- / 612 DarkGrey Text
- / 613 Other Text Color...

Back

- / 628 No Back Color
- / 616 White Back
- / 617 LightRed Back
- / 618 LightOrange Back
- / 619 LightYellow Back
- / 620 LightGreen Back
- / 621 LightBlue Back
- / 622 LightViolet Back
- / 623 LightGrey Back
- / 624 MediumGrey Back
- / 625 DarkGrey Back
- / 626 Black Back
- / 627 Other Back Color...

Strikethrough

- / 632 Automatic Strikethrough Color
- / 633 Black Strikethrough
- / 634 Red Strikethrough
- / 635 Orange Strikethrough
- / 636 Yellow Strikethrough
- / 637 Green Strikethrough
- / 638 Bleu Strikethrough
- / 639 Violet Strikethrough
- / 640 White Strikethrough
- / 641 LightGrey Strikethrough
- / 642 MediumGrey Strikethrough
- / 643 DarkGrey Strikethrough
- / 644 Other Strikethrough Color...

Underline

- / 646 Automatic Underline Color
- / 647 Black Underline
- / 648 Red Underline
- / 649 Orange Underline
- / 650 Yellow Underline
- / 651 Green Underline

/ 652 Blue Underline
/ 653 Violet
/ 654 White Underline
/ 655 LightGrey Underline
/ 656 MediumGrey Underline
/ 657 DarkGrey Underline
/ 658 Other Underline Color...
Shadow
/ 662 LightGrey Shadow
/ 663 MediumGrey Shadow
/ 664 DarkGrey Shadow
/ 665 Other Shadow Color...
Paragraph Back
/ 684 No Back Color
/ 672 White Paragraph Back
/ 673 LightRed Paragraph Back
/ 674 LightOrange Paragraph Back
/ 675 LightYellow Paragraph Back
/ 676 LightGreen Paragraph Back
/ 677 LightBlue Paragraph Back
/ 678 LightViolet Paragraph Back
/ 679 LightGrey Paragraph Back
/ 680 MediumGrey Paragraph Back
/ 681 DarkGrey Paragraph Back
/ 682 Black Paragraph Back
/ 683 Other Paragraph Back Color...
Border
/ 686 Black Border
/ 687 Red Border
/ 688 Orange Border
/ 689 Yellow Border
/ 690 Green Border
/ 691 Blue Border
/ 692 Violet Border
/ 693 White Border
/ 694 LightGrey Border
/ 695 MediumGrey Border
/ 696 DarkGrey Border
/ 697 Other Border Color...

Paragraph	No	Copy Ruler	701
	No	Paste Ruler	702
	No	Bullet ->	
	No	/ No Bullet	1021
	No	/ Black Square	1022
	No	/ White Square	1023
	No	/ Black Circle	1024
	No	/ White Circle	1025
	No	/ Diamonds	1026
	No	/ Clubs	1027
	No	/ Other Bullet...	1028
	Yes	Align Left	711
	Yes	Align Center	712
	Yes	Align Right	713
	Yes	Full Justification	714
	Yes	Single Spaced	721
	Yes	1.5 Line Spaced	722
	Yes	Double Spaced	723
	No	Other Line Spacing	724
	Format	No	Character...
No		Paragraph...	752
No		Tabs...	753
No		Borders...	754
Yes		Left border	1005
Yes		Top border	1006
Yes		Right border	1007
Yes		Bottom border	1008
Yes		All borders	1009
Yes		Borders inside	1010
Yes		No borders	1011
No		Style Sheets...	755
No		Columns...	756
Tools	No	Table Wizard...	408
	No	Spelling...	805
	No	Language...	806
	No	Document Information...	801
	No	Document Statistics...	802
	No	Compute References Now	803
	No	Freeze References	804

The following is a list of error codes returned by 4D Write:

Code	Text Error
1002	Error while printing.
1003	Invalid left margin parameter (too close to the right margin).
1004	Invalid indent parameter (too close to the right margin).
1005	Invalid right margin parameter (too close to the left margin and/or indent).
1006	Invalid tab parameter.
1007	Invalid array parameter: Array is not a valid type or size, or is not an array at all.
1012	The file has not been saved.
1013	Invalid selection (either start < 0 or end < start).
1015	The file has not been read.
1016	Invalid menu or item reference.
1017	This field does not seem to be a 4D Write field.
1022	Invalid area parameter passed to an external command.
1023	Invalid 4D file reference number.
1024	A 4D text variable or field allows a maximum of 32000 characters.
1028	Invalid position passed to WR Select.
1032	This file does not exist.
1034	There is no picture selected.
1035	Invalid size parameter.
1036	Invalid position parameter.
1038	This style does not exist.
1041	Not enough memory to execute this command.
1044	Invalid event type.
1047	Invalid field reference.
1048	Invalid option number.
1051	This path does not exist.
1054	First parameter is invalid.
1055	Second parameter is invalid.
1056	Third parameter is invalid.
1058	Fifth parameter is invalid.
1060	You cannot insert a subfield.
1066	You cannot create more than 256 tab stops.
1067	Invalid tab position.
1068	Invalid tab justification.
1069	You cannot insert a Blob.
1072	There is no hyphen to remove.
1073	Invalid expression.
1074	Invalid Blob.
1075	Text property out of range.
1076	Text property value out of range.
1077	Font not in system.

- 1078 Unknown stylesheet.
- 1079 Document property out of range.
- 1080 Document property value out of range.
- 1081 Selection has changed during printing.
- 1082 Invalid destination number.
- 1083 Invalid picture in page number.
- 1084 Invalid tab number.
- 1085 Page number format out of range.
- 1086 Invalid page number.
- 1087 Invalid column number.
- 1088 Invalid line number.
- 1089 Invalid option number.
- 1090 Invalid statistic number.
- 1091 Invalid frame reference.
- 1092 Invalid command number.
- 1093 Cannot print. Document is already printing.
- 1094 Reserved StyleSheet.
- 1095 Cannot Open File.

WR Text properties

wr bold	0
wr italic	1
wr shadow	2
wr strikethrough	3
wr underline	4
wr superscript or subscript	5
wr capital case	6
wr font number	7
wr font size	8
wr text color	9
wr text back color	10
wr strikethrough color	11
wr underline color	12
wr shadow color	13
wr stylesheet number	15
wr justification	32
wr line spacing	33
wr bullet	34
wr left margin	35
wr first indent	36
wr right margin	37
wr border back color	38
wr border line color	39
wr border line style	40
wr left border	41
wr right border	42
wr top border	43
wr bottom border	44
wr border spacing	45
wr tab	64

WR Text properties values

wr continuous underline	1
wr word underline	2
wr double underline	3
wr hatched underline	4
wr superscript	1
wr subscript	2
wr capitals	1
wr small capitals	2

wr left justified	0
wr centered	1
wr right justified	2
wr full justified	3
wr black square bullet	110
wr white square bullet	111
wr black circle bullet	108
wr white circle bullet	109
wr diamonds bullet	117
wr clubs bullet	118

WR Standard colors

wr automatic	-1
wr white	
16777215	
wr light grey	
13421772	
wr medium grey	
10066329	
wr dark grey	
6710886	
wr black	0
wr red	
16711680	
wr orange	
16750848	
wr yellow	
16770560	
wr green	52249
wr blue	
3381759	
wr violet	
13369599	
wr light red	
16757683	
wr light orange	
16767398	
wr light yellow	
16777164	
wr light green	
11796403	
wr light blue	
11790079	
wr light violet	
16761087	

WR Document properties

wr first page	0
wr view mode	1
wr view rulers	2
wr view frames	3
wr view headers	4
wr view footers	5
wr view pictures	6
wr view Hscrollbar	7
wr view Vscrollbar	8
wr view statusbar	9
wr view menubar	10
wr view standard palette	11
wr view format palette	12
wr view style palette	13
wr view borders palette	14
wr view invisible chars	15
wr view references	16
wr view column separators	17
wr first page header	18
wr opposite headers	19
wr widow orphan	20
wr unit	21
wr default tab	22
wr language	23
wr number of columns	24
wr columns spacing	25
wr binding	26
wr opposite pages	27
wr right first page	28
wr text inside margin	29
wr text outside margin	30
wr text left margin	31
wr text right margin	32
wr text top margin	33
wr text bottom margin	34
wr header top margin	35
wr header bottom margin	36
wr footer top margin	37
wr footer bottom margin	38
wr paper width	39
wr paper height	40
wr dead left margin	41
wr dead top margin	42
wr printable width	43
wr printable height	44

WR Options

wr confirm dialog	0
wr save preview	1
wr allow undo	2
wr modified	3
wr fixed print size	4
wr convert dialog	5
wr minimized button title	6
wr window title	7
wr minimum width	8
wr minimum height	9
wr save template on server	10
wr load template on server	11

WR Events

wr on key	0
wr on double click	1
wr on single click	2
wr on triple click	3
wr on activate	5
wr on printing	7
wr on ruler	8
wr on compute references	9
wr on close	10

WR Frames

wr text frame	0
wr right header	1
wr right footer	2
wr left header	3
wr left footer	4
wr first header	5
wr first footer	6

WR Commands

wr cmd undo	1
wr cmd redo	2
wr cmd cut	3
wr cmd copy	4
wr cmd paste	5
wr cmd clear	6
wr cmd select all	7
wr cmd about	10
wr cmd help	11

wr cmd goto full window	20
wr cmd new	101
wr cmd open	102
wr cmd save	103
wr cmd save as	104
wr cmd save as template	110
wr cmd preferences	105
wr cmd page setup	106
wr cmd print preview	107
wr cmd print	108
wr cmd print merge	109
wr cmd find	208
wr cmd find next	209
wr cmd replace	210
wr cmd replace next	211
wr cmd replace all	212
wr cmd lower case	221
wr cmd upper case	222
wr cmd title case	223
wr cmd toggle case	224
wr cmd view normal	302
wr cmd view page	303
wr cmd show selection	309
wr cmd view menubar	310
wr cmd view ruler	311
wr cmd view header	312
wr cmd view footer	313
wr cmd view references	314
wr cmd view pictures	315
wr cmd view invisibles	316
wr cmd view frames	317
wr cmd view HScrollbar	318
wr cmd view VScrollbar	319
wr cmd status bar	320
wr cmd view standard toolbar	331
wr cmd view format toolbar	332
wr cmd view style toolbar	333
wr cmd view borders toolbar	334
wr cmd insert date and time	401
wr cmd insert page number	402
wr cmd insert soft hyphen	404
wr cmd insert non break space	405
wr cmd insert page break	406
wr cmd insert 4D expression	407
wr cmd table wizard	408
wr cmd insert special char	409
wr cmd insert column break	410
wr cmd insert current hour	411

wr cmd insert current date	412
wr cmd plain	501
wr cmd bold	502
wr cmd italic	503
wr cmd shadow	504
wr cmd strikethrough	505
wr cmd superscript	506
wr cmd subscript	507
wr cmd capitals	508
wr cmd small capitals	509
wr cmd underline button	530
wr cmd no underline	521
wr cmd continuous underline	522
wr cmd word underline	523
wr cmd double underline	524
wr cmd hatched underline	525
wr cmd black text	602
wr cmd red text	603
wr cmd orange text	604
wr cmd yellow text	605
wr cmd green text	606
wr cmd blue text	607
wr cmd violet text	608
wr cmd white text	609
wr cmd light grey text	610
wr cmd medium grey text	611
wr cmd dark grey text	612
wr cmd other text color	613
wr cmd white back	616
wr cmd light red back	617
wr cmd light orange back	618
wr cmd light yellow back	619
wr cmd light green back	620
wr cmd light blue back	621
wr cmd light violet back	622
wr cmd light grey back	623
wr cmd medium grey back	624
wr cmd dark grey back	625
wr cmd black back	626
wr cmd other back color	627
wr cmd no back color	628
wr cmd auto striketh color	632
wr cmd black striketh	633
wr cmd red striketh	634
wr cmd orange striketh	635
wr cmd yellow striketh	636
wr cmd green striketh	637
wr cmd blue striketh	638

wr cmd violet striketh	639
wr cmd white striketh	640
wr cmd light grey striketh	641
wr cmd medium grey striketh	642
wr cmd dark grey striketh	643
wr cmd other striketh color	644
wr cmd auto underline color	646
wr cmd black underline	647
wr cmd red underline	648
wr cmd orange underline	649
wr cmd yellow underline	650
wr cmd green underline	651
wr cmd blue underline	652
wr cmd violet underline	653
wr cmd white underline	654
wr cmd light grey underline	655
wr cmd medium grey underline	656
wr cmd dark grey underline	657
wr cmd other underline color	658
wr cmd light grey shadow	662
wr cmd medium grey shadow	663
wr cmd dark grey shadow	664
wr cmd other shadow color	665
wr cmd white border back	672
wr cmd lgt red border back	673
wr cmd lgt orange border back	674
wr cmd lgt yellow border back	675
wr cmd lgt green border back	676
wr cmd lgt blue border back	677
wr cmd lgt violet border back	678
wr cmd lgt grey border back	679
wr cmd med grey border back	680
wr cmd dark grey border back	681
wr cmd black border back	682
wr cmd other border back color	683
wr cmd no border back color	684
wr cmd black border	686
wr cmd red border	687
wr cmd orange border	688
wr cmd yellow border	689
wr cmd green border	690
wr cmd blue border	691
wr cmd violet border	692
wr cmd white border	693
wr cmd light grey border	694
wr cmd medium grey border	695
wr cmd dark grey border	696
wr cmd other border color	697

wr cmd copy ruler	701
wr cmd paste ruler	702
wr cmd align left	711
wr cmd align center	712
wr cmd align right	713
wr cmd full justification	714
wr cmd single spaced	721
wr cmd 1.5 line spaced	722
wr cmd double spaced	723
wr cmd other line spacing	724
wr cmd character	751
wr cmd paragraph	752
wr cmd tabs	753
wr cmd borders	754
wr cmd stylesheets	755
wr cmd columns	756
wr cmd doc information	801
wr cmd doc statistics	802
wr cmd compute references	803
wr cmd freeze references	804
wr cmd spellcheck	805
wr cmd language	806
wr cmd goto page	807
wr cmd stylesheet dropdown	1000
wr cmd size dropdown	1001
wr cmd font dropdown	1002
wr cmd left border	1005
wr cmd top border	1006
wr cmd right border	1007
wr cmd bottom border	1008
wr cmd all borders	1009
wr cmd borders inside	1010
wr cmd no borders	1011
wr cmd standard bullet	1012
wr cmd left tab	1031
wr cmd centered tab	1032
wr cmd right tab	1033
wr cmd decimal tab	1034
wr cmd vertical separator	1035
wr cmd no bullet	1021
wr cmd black square bullet	1022
wr cmd white square bullet	1023
wr cmd black circle bullet	1024
wr cmd white circle bullet	1025
wr cmd diamonds bullet	1026
wr cmd clubs bullet	1027
wr cmd other bullet	1028

WR Tabs

wr left tab	1
wr centered tab	2
wr right tab	3
wr decimal tab	4
wr vertical separator tab	5

WR Count

wr nb characters	0
wr nb words	1
wr nb paragraphs	2
wr nb pictures in text flow	3
wr nb objects	4
wr nb soft hyphens	5
wr nb page breaks	6
wr nb column breaks	7
wr nb insertions date time	8
wr nb insertions page number	9
wr nb lines	10
wr nb pages	11
wr nb stylesheets	12
wr nb pictures in page	13

Command Index

A

WR ADD STYLESHEET TAB	95
WR ADD TAB	96
WR APPLY STYLESHEET	103
WR Area to blob	49
WR Area to picture	50
WR AUTO SCROLL	25

B

WR BACKSPACE	121
WR BLOB TO AREA	51
WR Build preview	37

C

WR COLOR TO RGB	145
WR Count	146
WR Create stylesheet	104

D

WR DELETE OFFSCREEN AREA	52
WR DELETE PICTURE IN PAGE	75
WR DELETE SELECTION	122
WR DELETE STYLESHEET	105
WR DELETE STYLESHEET TAB	106
WR DELETE TAB	98
WR Direct find	123

E

WR Error number	147
WR Error text	148

WR EXECUTE COMMAND	26
--------------------	----

F

WR Find	124
WR FONTS TO ARRAY	149

G

WR GET AREA PROPERTY	38
WR GET COMMAND INFO	27
WR GET CURSOR POSITION	40
WR Get doc property	29
WR GET DOCUMENT INFO	65
WR Get font	125
WR Get frame	41
WR GET PARAGRAPHS	126
WR GET PICTURE IN PAGE INFO	76
WR GET PICTURE SIZE	78
WR GET REFERENCE	57
WR Get selected picture	79
WR Get selected text	127
WR GET SELECTION	128
WR Get styled text	80
WR Get stylesheet font	107
WR GET STYLESHEET INFO	108
WR GET STYLESHEET TAB	109
WR Get stylesheet text prop	110
WR GET TAB	99
WR Get text	129
WR Get text property	130
WR GET WORDS	132

I

WR INSERT EXPRESSION	59
WR INSERT FIELD	61

WR INSERT PAGE NUMBER	89
WR INSERT PICTURE	81
WR Insert picture area	62
WR INSERT STYLED TEXT	134
WR INSERT TEXT	135

L

WR LOCK COMMAND	30
WR LOCK DOCUMENT	66

M

WR Mouse to Selection	136
-----------------------	-----

N

WR New offscreen area	53
-----------------------	----

O

WR O Area to picture	159
WR O AUTO SAVE	160
WR O CHANGE STYLE	161
WR O COMPUTE NOW	162
WR O Count Stylesheet	163
WR O CREATE STYLESHEET	164
WR O DELETE STYLESHEET	165
WR O DISPLAY MENUBAR	166
WR O DISPLAY RULER	167
WR O DISPLAY SCROLLBARS	168
WR O DO COMMAND	169
WR O EXPERT COMMAND	170
WR O Find	171
WR O Font name	172
WR O Font number	173

WR O GET ATTRIBUTES	174
WR O GET MARGINS	176
WR O Get pack options	177
WR O Get page	179
WR O GET PICTURE	180
WR O GET PREFERENCES	181
WR O GET RULER	182
WR O Get ScrollBars	184
WR O GET STYLESHEET	185
WR O GET TABS	187
WR O INSERT HYPHEN	189
WR O INSERT PICTURE	190
WR O Is Hyphen	191
WR O LINE SPACING	192
WR O MENU STATUS	193
WR O ModuleInfo	194
WR O MOVE PICTURE	195
WR O ON MENU	196
WR O OPTIONS	197
WR O Page number	199
WR O PICTURE TO AREA	200
WR O Picture to offscreen area	201
WR O REMOVE HYPHEN	202
WR O Replace	203
WR O RESIZE PICTURE	204
WR O Save to picture	205
WR O SET ATTRIBUTES	206
WR O SET MARGINS	208
WR O SET PACK OPTIONS	209
WR O SET PREFERENCES	211
WR O SET TABS	214
WR O STATISTICS	215
WR O STRUCTURE ACCESS	216
WR O TEXT ALIGNMENT	217
WR ON COMMAND	31
WR ON ERROR	150
WR ON EVENT	151
WR OPEN DOCUMENT	67

P

WR PICTURE TO AREA	54
WR PRINT	90
WR PRINT MERGE	91

R

WR REDRAW	32
WR REPLACE	137
WR RGB to color	154

S

WR SAVE DOCUMENT	69
WR SELECT	138
WR SELECT PICTURE IN PAGE	111
WR SET AREA PROPERTY	42
WR SET CURSOR POSITION	44
WR SET DOC PROPERTY	33
WR SET DOCUMENT INFO	71
WR SET FONT	140
WR SET FRAME	45
WR SET PICTURE IN PAGE INFO	83
WR SET PICTURE SIZE	85
WR SET SELECTION	141
WR SET STYLESHEET FONT	112
WR SET STYLESHEET INFO	113
WR SET STYLESHEET TAB	115
WR SET STYLESHEET TEXT PROP	117
WR SET TAB	100
WR SET TEXT PROPERTY	142

T

WR TEXT ACCESS _____ 46

U

WR UPDATE MODE _____ 34

WR UPDATE STYLESHEET _____ 118

