

4D Write

ランゲージリファレンス
Windows® and Mac™ OS



ACI

4D Write ランゲージリファレンス Windows® and Mac™ OS

Copyright© 1985 - 2000 ACI SA

All rights reserved.

このマニュアルに記載されている事項は、将来予告なしに変更されることがあり、いかなる変更に関しても ACI SA は一切の責任を負いかねます。このマニュアルで説明されるソフトウェアは、本製品に同梱の License Agreement (使用許諾契約書) のもとでのみ使用することができます。

ソフトウェアおよびマニュアルの一部または全部を、ライセンス保持者がこの契約条件を許諾した上での個人使用目的以外に、いかなる目的であれ、電子的、機械的、またどのような形であっても、無断で複製、配布することはできません。

4th Dimension、4D Server、4D、4D ロゴ、ACI ロゴ、およびその他の ACI 製品の名称は、ACI SA の商標または登録商標です。

Microsoft と Windows は Microsoft Corporation 社の登録商標です。

Apple, Macintosh, Mac, Power Macintosh, Laser Writer, Image Writer, ResEdit, QuickTime は Apple Computer Inc. の登録商標または商標です。

その他、記載されている会社名、製品名は、各社の登録商標または商標です。

注意

このソフトウェアの使用に際し、本製品に同梱の License Agreement (使用許諾契約書) に同意する必要があります。ソフトウェアを使用する前に、License Agreement を注意深くお読みください。

第1章	4D Write ランゲージの紹介	9
	はじめに	9
	マルチプラットフォームドキュメントの管理	10
	表記方法について	11
	メソッドエディタ内のコマンド	12
	4D Write エリア内のドキュメント	13
	4D Write メニューアイテム	15
	文字の参照	16
第2章	エリア制御コマンド	17
	WR SCROLL TO SELECTION	17
	WR EXECUTE COMMAND	18
	WR GET COMMAND INFO	19
	WR Get doc property	21
	WR LOCK COMMAND	22
	WR ON COMMAND	24
	WR REDRAW	25
	WR SET DOC PROPERTY	26
	WR UPDATE MODE	29
第3章	エリアオプションコマンド	31
	WR Build preview	31
	WR GET AREA PROPERTY	33
	WR GET CURSOR POSITION	35
	WR Get frame	37
	WR SET AREA PROPERTY	38
	WR SET CURSOR POSITION	40
	WR SET FRAME	42
	WR TEXT ACCESS	43

第4章	エリアコマンド	45
	WR Area to blob	45
	WR Area to picture	46
	WR BLOB TO AREA	48
	WR DELETE OFFSCREEN AREA	49
	WR New offscreen area	50
	WR PICTURE TO AREA	51
第5章	データベースオブジェクトコマンド	53
	WR GET REFERENCE	53
	WR INSERT EXPRESSION	55
	WR INSERT FIELD	56
	WR Insert picture area	57
第6章	ドキュメントコマンド	59
	WR GET DOCUMENT INFO	59
	WR LOCK DOCUMENT	61
	WR OPEN DOCUMENT	62
	WR SAVE DOCUMENT	64
	WR SET DOCUMENT INFO	66
第7章	ピクチャ制御コマンド	69
	WR DELETE PICTURE IN PAGE	69
	WR GET PICTURE IN PAGE INFO	70
	WR GET PICTURE SIZE	72
	WR Get selected picture	73
	WR INSERT PICTURE	74
	WR SELECT PICTURE IN PAGE	76
	WR SET PICTURE IN PAGE INFO	77
	WR SET PICTURE SIZE	80
第8章	プリントコマンド	81
	WR INSERT PAGE NUMBER	81
	WR PRINT	83
	WR PRINT MERGE	84

第9章	ルーラーコマンド	85
	WR ADD TAB	85
	WR DELETE TAB	87
	WR GET TAB	88
	WR SET TAB	90
第10章	スタイルシートコマンド	93
	WR ADD STYLESHEET TAB	93
	WR APPLY STYLESHEET	95
	WR Create stylesheet	96
	WR DELETE STYLESHEET	98
	WR DELETE STYLESHEET TAB	99
	WR Get stylesheet font	100
	WR GET STYLESHEET INFO	101
	WR GET STYLESHEET TAB	103
	WR Get stylesheet text prop	105
	WR SET STYLESHEET FONT	107
	WR SET STYLESHEET INFO	108
	WR SET STYLESHEET TAB	110
	WR SET STYLESHEET TEXT PROP	112
	WR UPDATE STYLESHEET	114
第11章	テキスト操作コマンド	115
	WR BACKSPACE	115
	WR DELETE SELECTION	116
	WR Direct find	117
	WR Find	119
	WR Get font	121
	WR GET PARAGRAPHS	122
	WR Get selected text	123
	WR GET SELECTION	124
	WR Get styled text	125
	WR Get text	126
	WR Get text property	127
	WR GET WORD	130
	WR INSERT STYLED TEXT	132
	WR INSERT TEXT	133
	WR Mouse to Selection	134

	WR Replace	135
	WR SELECT	137
	WR SET FONT	139
	WR SET SELECTION	140
	WR SET TEXT PROPERTY	141
第 12 章	ユーティリティコマンド	143
	WR COLOR TO RGB	143
	WR Count	144
	WR Error number	146
	WR Error text	147
	WR FONT TO ARRAY	148
	WR ON ERROR	149
	WR ON EVENT	150
	WR RGB to color	154
第 13 章	旧コマンド	155
	WR O Area to picture	155
	WR O AUTO SAVE	156
	WR O CHANGE STYLE	157
	WR O COMPUTE NOW	158
	WR O Count stylesheet	159
	WR O CREATE STYLESHEET	160
	WR O DELETE STYLESHEET	161
	WR O DISPLAY MENUBAR	162
	WR O DISPLAY RULER	163
	WR O DISPLAY SCROLLBARS	164
	WR O DO COMMAND	165
	WR O EXPERT COMMAND	166
	WR O Find	167
	WR O Font name	168
	WR O Font number	169
	WR O GET ATTRIBUTES	170
	WR O GET MARGINS	172
	WR O Get pack options	173
	WR O Get page	175
	WR O GET PICTURE	176
	WR O GET PREFERENCES	177

WR O GET RULER	178
WR O Get ScrollBars	180
WR O GET STYLESHEET	181
WR O GET TAB	183
WR O INSERT HYPHEN	185
WR O INSERT PICTURE	186
WR O Is Hyphen	187
WR O LINE SPACING	188
WR O MENU STATUS	189
WR O MOVE PICTURE	190
WR O ON MENU	191
WR O OPTION	192
WR O Page number	194
WR O PICTURE TO AREA	195
WR O Picture to offscreen area	196
WR O REMOVE HYPHEN	197
WR O Replace	198
WR O RESIZE PICTURE	199
WR O Save to picture	200
WR O SET ATTRIBUTES	201
WR O SET MARGINS	203
WR O SET PACK OPTIONS	204
WR O SET PREFERENCES	206
WR O SET STYLESHEET	207
WR O SET TABS	209
WR O STATISTICS	210
WR O STRUCTURE ACCESS	211
WR O TEXT ALIGNMENT	212
第14章 付録	213
付録A：ショートカット	213
付録B：メニューアイテム番号	215
付録C：エラーコード	221
付録D：4D Write 定数	223
コマンド索引	233

はじめに

4D Write は、4th Dimension にワードプロセッシング用のコマンドと機能を付加するプラグインです。これらのコマンドを利用することによって、(下記に挙げたような) 定型的な手動作業を自動化することが可能になります：

メニューコマンドの実行

ドキュメントを開く / ドキュメントの保存

ドキュメントのマージン設定

表示属性の設定

4th Dimension に追加されるすべての 4D Write コマンドには、先頭に WR という文字がついています。これによって、4th Dimension や 4D Write 以外のプラグインのコマンドと区別しています。

4D Write のドキュメント

4D Write のドキュメントは、『4D Write ユーザリファレンス』と『4D Write ランゲージリファレンス』という2つのマニュアルから成り立っています。本マニュアル(4D Write ランゲージリファレンス)は、4D Write のプログラミングランゲージの使用方法について述べています。4D Write の使い方に関する情報は、『4D Write ユーザリファレンス』を参照してください。

マルチプラットフォームドキュメントの管理

4D Writeは、4th Dimensionや4D Serverと同様に、マルチプラットフォームプログラムです。つまり、Mac OS上で作成された4D Writeを利用しているデータベースを、何の修正も加えずにWindows上で稼働させること、またはその逆も可能であるということです。ただし、これが可能であるのは同一バージョンのソフトウェアをお使いの場合に限られます。

Mac OS と Windows のファイル対応

次の表は、Mac OSとWindowsでの4D Writeドキュメントのファイル対応を示しています。

ドキュメント	Mac OS		Windows
	タイプ	クリエータ	
4D Write 6.5 ドキュメント	4WR7	4DW7	4W7
RTF	TEXT	4DW7	RTF
Windowsテキストのみ	TEXT	4DW7	TXT
Mac OSテキストのみ	TEXT	4DW7	TXT
Unicodeテキストドキュメント	TEXT	4DW7	TXT
HTMLドキュメント		4DW7	TXT

ドキュメント

下記のルールについては、認識しておく必要があります：

Mac OS上では、4D Writeはドキュメントを見分けるためにタイプとクリエータを利用します。例えば、タイプが4WR7でクリエータが4DW7であれば、4D Writeドキュメントです。完全なアクセスパスは、それぞれがコロンの(:)で区切られたディスク名、フォルダ名、ドキュメント名を含みます。例えば、“MyMac:フォルダ1:フォルダ2:MyDatabase” というようなものです。

Windows上では、4D Writeはドキュメントを見分けるためにファイル名の拡張子を利用します。例えば、“.4WR7”であれば、4D Writeドキュメントです。完全なアクセスパスは、それぞれが円マーク(¥)で区切られたディスク名、フォルダ名、ドキュメント名を含みます。例えば、“D:¥フォルダ1¥フォルダ2¥MyDatabase” というようなものです。

Mac OS上で作成され、Windows上にコピーされた4D Writeドキュメントは、ファイル名が拡張子付きで保存されているので、そのまま開くことができます。例えば、MyDocドキュメントは、PCのボリュームにコピーされた際に、MyDoc.4W7として保存されるので、何も手を加えずに開くことができます。

Windows上で作成され、Mac OS上またはPower Macintosh上にコピーされた4D Writeドキュメントは、何も手を加えないで開くことができます。

テンプレート

Mac OSとWindowsのクライアント間でサーバのプラットフォームに関係なくテンプレートを共有するための手順を次に示します。

サーバがWindowsマシンであれば、テンプレートファイルの名前は“エリア名_4WT”にしてください。サーバがMacintoshマシンであれば、テンプレートファイルの名前は“エリア名_”にしてください。

表記方法について

説明

本マニュアルでは、4D Writeコマンドは特殊フォントを使用した大文字で印刷されていません（例：**WR ON COMMAND**）。4D Write関数は、先頭文字のみが大文字で表示されています（例：**WR Get styled text**）。

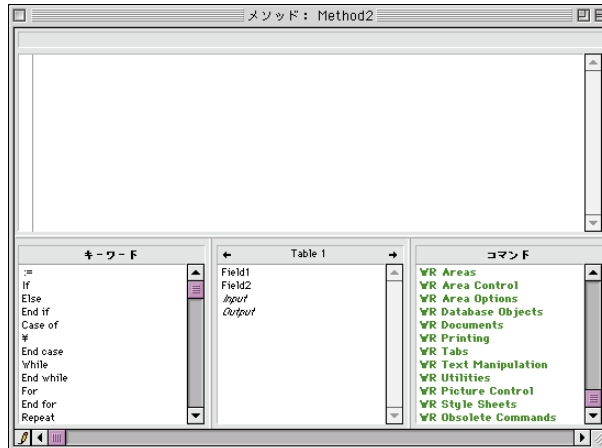
4D Writeコマンドや4D Write関数がメソッドやオブジェクトメソッド中に表示される際には、4th Dimensionの標準のコマンドや関数と区別するために、太字の斜体（ボールドイタリック）の書体で表示されます。斜体でない太字のテキストは、4th Dimensionのランゲージの単語であることを示します。

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

本マニュアルのいくつかの例題において、余白の都合でコードの行が2行目または3行目へと続くものがあります。しかし、これらの例題を入力する場合には、そのコードは1行で記述してください（途中で改行を入力しないでください）。

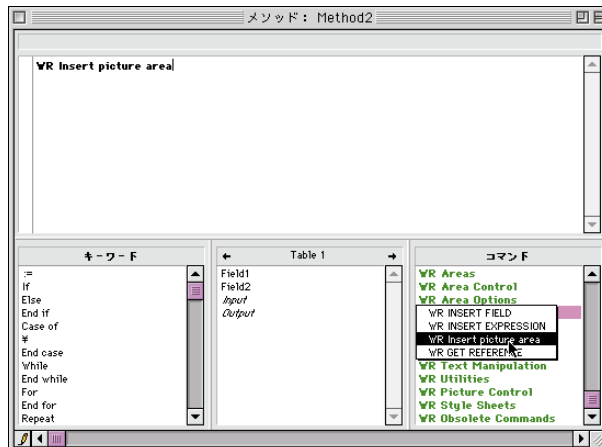
メソッドエディタ内のコマンド

4D Writeのコマンドは、メソッドエディタ内ではテーマ別にグループ化され、コマンドリストの最後に配置されています。



さらに別のプラグインをインストールすると、そのコマンドテーマは、プラグインがインストールされた順番で表示されます。

4th Dimension コマンドを配置するのと同じように、4D Write コマンドをメソッド中に配置することができます。メソッド内に直接入力すること、及び、コマンドリストのポップアップメニューから選択することができます。



4D Write コマンドは、すべてのメソッドタイプ、すなわちグローバル、テーブル、フォーム、データベースメソッドで利用することができます。その中でも4D Write コマンドは、4D Write のドキュメントエリアと同じ場所にあるフォーム上のオブジェクトでアクティブ化されたオブジェクトメソッド内において、特に効果的です。

4D Write エリア内のドキュメント

4D Write ドキュメントは、下記の4th Dimensionの3つのエリアで使用可能です。

フォーム内のプラグインエリア

プラグインウインドウ

オフスクリーンエリア

4D Write ドキュメントを利用するには、フォーム上にプラグインエリアを作成するか、またはプラグインウインドウを開きます。プラグインエリアを作成するには、「デザイン」モードでフォーム上にエリアを描画します。プラグインウインドウを開くには、「ユーザ」モードの「ウインドウ」メニューから「4D Write」を選択するか、または**Open external window**関数を実行します。

可視エリアを作成することに加えて、不可視オフスクリーンエリアを作成することもできます。詳細は、後述の「4D Write オフスクリーンエリア」の節を参照してください。

4D Write エリアの ID 番号と変数

4D Write は、4D Write エリア、プラグインウインドウおよびオフスクリーンエリアの位置を保持するために変数を使います。コマンドまたは関数に対して、エリア番号を代入した変数を引数として渡し、操作対象のエリアを参照します。

後述のコマンドの説明において、<エリア> はドキュメントエリアを特定する変数として参照されます。

次の2つのタイプの変数があります：

プラグインオブジェクト名

4D Write エリアを作成し、名前を付ける際に、エリアを参照するための変数として4D Write エリアの名前を4th Dimension が自動的に認識します。例えば、Letter という名前のエリアを処理の対象にする場合には、<エリア> に “Letter” を指定します。

プラグインウインドウまたはオフスクリーンエリアのために作成した変数

Open external window または **WR New offscreen area** を使ってプラグインウインドウやオフスクリーンエリアを作成した場合、関数からエリアID番号が返されるので、変数に受け取ることができます。この変数を利用すれば、他のコマンドや関数でプラグインウインドウやオフスクリーンエリアに対して処理を行うことができます。値を変数に受け取るには、コード内の行の関数の左側に変数名と代入記号(=)を記述します。

ほとんどの4D Write コマンドは、実行するためにエリアを指定する必要があります。

4D Write **プラグインエリア**

4D Write ドキュメントを 4th Dimension のフォーム内に表示するためには、フォーム上にプラグインエリアを作成し、他と重複しない名前を割り当て、プラグインエリアのタイプとして 4D Write を指定します。

4th Dimension では、このドキュメントをレコードと一緒に保存することができます。

プラグインエリアは、ドキュメントを保存するためによく使用されます。また書式が重要な場合は、テキストフィールドの代わりにプラグインエリアを使用することもあります。

4D Write **プラグインウィンドウエリア**

4th Dimension では、プラグインウィンドウと呼ばれる独立したエリアに 4D Write ドキュメントを作成することができます。プラグインウィンドウは、ワードプロセッサとして手紙やメモ、またはその他のドキュメントにユーザがアクセスする場合に便利です。

メソッドから 4th Dimension 関数、プラグインウィンドウを実行すると、指定されたウィンドウを開くと同時に倍長整数の変数にエリア ID を返します。この変数を利用すれば、4D Write コマンドを実行したい時にはいつでもプラグインウィンドウを指定することができます。

例えば：

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

Open external window については、『4th Dimension ランゲージリファレンス』マニュアルを参照してください。

4D Write **オフスクリーンエリア**

オフスクリーンエリアはメモリ上に格納され、プログラマやユーザは見ることができませんが、ドキュメントを修正するためにオフスクリーンエリアを使うことができます。ユーザが見る前やドキュメントを保存する前に、必要であればユーザが元の内容に戻すこともできます。

WR New Offscreen area と **WR Picture to offscreen area** は、オフスクリーンエリアを作成する際に用います。これらのエリア関数についての詳細は、第 9 章「ルーラーコマンド」を参照してください。使用しているメモリを解放するために、これらのコマンドの使用後にオフスクリーンエリアを消去することを忘れないでください。

ドキュメントを保存するためにオフスクリーンエリアを作成する場合は、下記のコードをグローバルメソッドで記述してください。

```
QUERY([Employee];[Employee]ID=vID)
If (Records in selection([Employee]=1)
  エリア:=WR New offscreen area
  $res:=WR Picture to area(エリア;[Employee]Review_)
  `調査票フォームをオフスクリーンエリアに代入する
MODIFY RECORD([Employee])
  `従業員レコードを修正
WR DELETE OFFSCREEN AREA(Area)
  `オフスクリーンエリアで使用したメモリを解放する
End if
```

フォーム上のボタンを使って、保存されている元のドキュメントに復帰させることも可能です。

入力フォーム上にボタンを作成し、次のコードを記述してください：

```
Review:=WR Area to picture(Area)
  `Review フォーム内にあるプラグインエリアに元のドキュメントを含んでいる
  `オフスクリーンエリアを配置する。
```

4D Write メニューアイテム

プロシージャで4D Writeメニューにアクセスし、任意のメニューアイテムを選択することができます。メソッド中でメニューまたはメニューアイテムの状態を決定することができます。各メニューアイテムは、それぞれ重複しない異なる整数値が割り振られています。それぞれのメニューアイテムと対応する整数値の一覧は、第14章「付録」を参照してください。

メニューアイテムに割り振られた整数値は、一般的にはメニューとメニューアイテムの位置が基本となっています。メニューは、左から右の順に昇順で番号が付けられます。例えば、「ファイル」メニューは100、「編集」メニューは200となります。同様に、メニューアイテムは上から下の順に昇順で番号が付けられています。したがって、「新規」というメニューアイテムは、最初のメニューである「ファイル」メニューの最初のアイテムなので、101という番号が付けられています。

これらのメニューアイテムの番号は、4D Writeの新しいバージョンで新しいメニューアイテムが追加された場合でも変わりません。新規のメニューアイテムは、現行のメニューアイテムの間に配置されたとしても、別の番号を使用します。この番号付けは、一般的なルールから外れますが、メソッド内で使用してるメニューの参照番号は確実に残るので、参照番号を更新しなくて済みます。

文字の参照

ドキュメント内の文字は、一連番号で参照されます。文字を参照するコマンドは、1文字または一連の文字を指定することができます。例えば、単語、一文、選択されたテキストのブロック全体を指定することができます。

WR GET SELECTIONを使用して、4D Writeエリア内の選択された文字の位置を決めます。このコマンドは、<先頭>と<最終>を使用して、選択された一連の文字列を参照します。<先頭>は、常に選択された文字の先頭よりも1小さくなります。<最終>は、選択した最後の文字の位置と同じになります。

使用例

例えば、下記の表記により、<エリア>内のテキストの選択されている位置を変数<\$First>と変数<\$Last>に返します。

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

4D Writeエリア内のテキストを選択するには、文字の参照が必要です。ほとんどの場合、コマンドを使用して操作を行う前に、先にテキストを選択する必要があります。詳細は、第11章「テキスト操作コマンド」を参照してください。

WR SCROLL TO SELECTION

WR SCROLL TO SELECTION (エリア)

引数	タイプ	説明
エリア	整数	4D Write エリア

説明

WR SCROLL TO SELECTION は、選択されたテキストが表示されるまでエリアをスクロールします。このコマンドは、4D Write コマンドを利用して修正を行い、その変更結果を見る必要がある場合に有効です。

例題

WR Get font 関数、及び **WR SET CURSOR POSITION** コマンドの例を参照してください。

参照

なし

WR EXECUTE COMMAND

WR EXECUTE COMMAND(エリア; コマンド番号)

引数	タイプ	説明
エリア	倍長整数	4D Write エリア
コマンド番号	倍長整数	実行するコマンド番号

説明

WR EXECUTE COMMANDは、4D Writeメニューコマンドやツールバーに割り当てられている処理を実行します。このコマンドの最も一般的な使い方としては、ユーザがメニューなどから4D Writeコマンドをコールした際に、**WR ON COMMAND**により割り込みをかけ、メソッドを実行した後に、本来ユーザが選択した4D Writeコマンドを実行することがあげられます。

注：このコマンドリストとその説明については、付録Bの「メニューアイテム番号」を参照してください。この引数には、定数と値のどちらでも渡すことができます。

参照

WR ON COMMAND

WR GET COMMAND INFO

WR GET COMMAND INFO (エリア; コマンド番号; 数値戻り値; 文字戻り値; 名前; ステータス)

引数	タイプ	説明
エリア	倍長整数	4D Write エリア
コマンド番号	倍長整数	処理コマンドの番号
数値	倍長整数	コマンドのステータス
文字	文字	選択されたテキストの値
名前	文字	コマンド名またはTipのテキスト
ステータス	整数	0 = 使用不可 1 = 使用可能

説明

WR GET COMMAND INFO は、 <コマンド番号> で指定された番号のメニューやツールバーのコマンドステータスを取得することができます。

注：コマンド一覧およびリファレンスは、付録Dの「WR Commands」を参照してください。この引数には、定数と値のどちらでも渡すことができます。

<数値> は、コマンドのステータスを返します。 <数値> の値は、コマンドが使用不可の場合は0、使用可能である場合は1、部分的に使用可能である場合には2となります（例えば、選択されたテキストのうち、一部が太字表示、他の部分が太字表示でない場合に太字コマンドを使う場合）。

<文字> には、各コマンドで変更または指定されるテキストが返されます。

<名前> には、コマンド名が返されます。これは、メニューコマンドのテキスト、またはそのコマンド用に表示されたTipsのテキストです。

<ステータス> には、メニューコマンドのステータスが返されます。メニューコマンドが使用不可であれば0、使用可能な場合には1の値が<ステータス>に返されます。

例題

次の例は、太字のコマンドが適用されます（定数：wr cmd gras、値：502）

WR GET COMMAND INFO(Area;wr cmd gras;numValue;stringValue;name;status)

<数値> に返る値は以下のようになります。

- numValue=1：テキストが太字表示の場合
- numValue=0：テキストが太字表示でない場合
- numValue=2：選択した部分だけが太字表示の場合

次の例は、「フォント」ドロップダウンリストに適用されます（定数：wr cmd font dropdown、
値：1002）。

```
WR GET COMMAND INFO(Area;wr cmd font dropdown;numValue;stringValue;name;  
                                                                    status)  
stringValue="Osaka"
```

参照

「付録B：メニューアイテム番号」、WR EXECUTE COMMAND

WR Get doc property

WR Get doc property(エリア; プロパティ)

引数	タイプ	説明
エリア	倍長整数	4D Write エリア
プロパティ	整数	読み取り用のプロパティ番号
戻り値	実数	テストされたプロパティの値

説明

WR Get doc property を使用すれば、<エリア> で指定された 4D Write エリア内で開かれているドキュメントの属性を取得することができます。

プロパティ

いくつかのプロパティについては、戻り値<値>に 1 (真) または 0 (偽) が返されます。例えば、プロパティ 2 (wr view ruler) などです。

その他のプロパティについては、現在のデフォルト単位で表現されている番号が返されます。例えば、プロパティ 37 (wr paper width) などです。

注：プロパティは、定数を利用して設定することができます。

ドキュメントのプロパティリストは、付録 D の「WR Document properties」を参照してください。この引数には、定数と値のどちらでも渡すことができます。

参照

WR SET DOC PROPERTY

WR LOCK COMMAND

WR LOCK COMMAND (エリア; コマンド番号; ロック)

引数	タイプ	説明
エリア	倍長整数	4D Write エリア
コマンド番号	倍長整数	処理するコマンド番号
ロック	整数	0 = 実行可能 1 = 実行不可能

説明

WR LOCK COMMANDを使用することで、<コマンド番号>に渡された番号により指定されたコマンドを実行不可能にすることができます。メニューコマンドでもパレットコマンドにも操作可能です。このコマンドは、<エリア>によって指定された4D Write エリア内のみ、表示されているコマンドへのアクセスに影響を及ぼします。他の4D Write エリア内のコマンドへのアクセスには、影響を及ぼしません。

注：

- ・ コマンドがロックされていても、**WR EXECUTE COMMAND**を利用すればコードによってコマンドを実行することができます。
- ・ 使用不可のコマンドを選択した場合には、**WR ON COMMAND**は呼び出されません。
- ・ ロック
 - ・ <ロック>が0の場合、そのコマンドは使用できます。
 - ・ <ロック>が1の場合、そのコマンドは使用できません。

メニュー全体を実行不可能にするには<コマンド番号>に以下の値のうちの1つを渡します。

メニュー	コマンド番号
ファイル	100
編集	200
表示	300
挿入	400
スタイル	500
カラー	600
段落	700
書式	750
ツール	800

実行不可能にしたメニューのコマンドは選択することができませんが、キーボードからの入力、あるいはツールバーのボタンは使用できます。これらのコマンドを完全にロックしたい場合は、各メニューアイテムに対し、**WR LOCK COMMAND**を呼び出します。

注：定数のリストは、付録Dの「4D Write 定数」を参照してください。

例題

1. 以下の例は、デザインモードにアクセスできるユーザをデザイナーだけにしたい場合を示します。

```
If(Current user="デザイナー")  
    WR LOCK COMMAND(Area;wr cmd insert 4D expression;0)  
Else  
    WR LOCK COMMAND(Area;wr cmd insert 4D expression;1)  
End if
```

2. 以下の例は、ユーザ名が「社員」以外の場合、新しいドキュメントを作れないようにします。

```
If(Form event=On load)  
    If (Current user#"社員")  
        WR LOCK COMMAND(Area;wr cmd new;1)  
    End if  
End if
```

参照

「付録B：メニューアイテム番号」、「付録D：4D Write 定数」、WR ON COMMAND

WR ON COMMAND

WR ON COMMAND (エリア; 4D メソッド)

引数	タイプ	説明
エリア	倍長整数	4D Write エリア
4Dメソッド	文字	置き換えメソッド

説明

WR ON COMMAND は、メニューからコマンドを選択する、またはボタンをクリックすることにより、ユーザによって 4D Write コマンドが呼び出された際に、<メソッド名> で指定されたメソッドを実行します。<エリア> が 0 の場合、<メソッド名> で指定されたメソッドはデータベースが閉じられるか、次のような **WR ON COMMAND** が呼び出されるまで、各 4D Write エリアに適用されます：

WR ON COMMAND (0; "")

<メソッド名> で指定されたメソッドは、2つの引数を受け取ります：

\$1：<エリア> を表す倍長整数です。

\$2：コマンド番号を表す倍長整数です。

コンパイルされたデータベースを使用する場合は、これらの変数を使用しない場合でも、\$1 および \$2 の両方を宣言する必要があります。

初期コマンドを実行する場合は、メソッド内に次の記述を行う必要があります：

WR ON COMMAND (\$1;\$2)

WR REDRAW

WR REDRAW(エリア)

引数	タイプ	説明
エリア	倍長整数	4D Write エリア

説明

WR REDRAWによって、エリアが再表示されます。このコマンドは、**WR UPDATE MODE**によってスクリーンの更新が行われないように設定されている場合に、実行されたコードによってエリアがどのように変更されたかを確認するために、4D Write エリアを再表示したい場合に便利です。

例題

次の例は、スクリーンの更新をオフにして、<エリア>で指定されたエリアを再フォーマットする **Reformat** というグローバルメソッドを呼び出し、スクリーンの更新をオンに戻さないで <エリア> を再表示させます。

```
WR UPDATE MODE (Area;0)
  `スクリーンの更新をオフにする
Reformat(Area)
  `Area は、メソッドに渡される
WR REDRAW (Area)
  `再表示
```

参照

なし

WR SET DOC PROPERTY

WR SET DOC PROPERTY (エリア;プロパティ;値)

引数	タイプ	説明
エリア	倍長整数	4D Write エリア
プロパティ	整数	設定するプロパティ番号
戻り値	数値	選択されたプロパティの値

説明

この **WR SET DOC PROPERTY** を使用すれば、<エリア>で指定された4D Write エリア内で開かれているドキュメントの属性を変更することができます。

<値>の意味は、使用されたプロパティの値によって変わります。

<プロパティ>が、プロパティ定数である wr view ruler であるか、または2である場合、<値>の値は1 (表示されている) か2 (表示されていない) のどちらかになります。

<プロパティ>が、プロパティ定数である wr column spacing であるか、または25である場合、<値>の値は現在のデフォルト値になります。

注：<プロパティ>および<値>は、定数によって設定することができます。

テキストのプロパティリストは、付録Dの「WR Text properties」を参照してください。

<プロパティ>の値は定数プロパティである wr language であるか、または23である場合、<値>の値は以下ようになります。

言語	値
American English	1033
English Australian	3081
English	2057
Catalan	1027
Danish	1030
Dutch	1043
Finish	1035
French	1036
French Canadian	3084
German	1031
Italian	1040
Norwegian Bokmal	1044
Norwegian Nynorsk	2068
Portuguese Brazil	1046

Portuguese Iberian	2070
Spanish	1034
Swedish	1053
Russian	1049
Czech	1029
Hungarian	1038
Polish	1045

次の4つのプロパティは、読み込み専用モードでのみ有効で、**WR SET DOC PROPERTY** コマンドを使っても設定することはできません。

定数	値
wr dead left margin	39
wr dead top margin	40
wr printable width	41
wr printable height	42

用紙サイズをコードによって設定する場合、4D Writeは余白を0（設定された用紙サイズ）に設定する点に注意してください。

これにより、余白についての処理を行うことなくドキュメントのマージンを設定することができます。ドキュメントを印刷する予定のない場合（例えば、ピクチャを作成するために使用する場合）に有効です。

以下の8つのプロパティ（4D Writeのバージョン6.5.2以降で有効）は、ドキュメントの1ページ目だけを他のページと異なる設定にしたい場合に使用できます。

定数	値
wr view first page header	50
wr view first page footer	51
wr first page top margin	52
wr first page bottom margin	53
wr header 1st page top margin	54
wr header 1st page bottom mg	55
wr footer 1st page top margin	56
wr footer 1st page bottom mg	57

例題

1. 以下の例は、メニューとルーラーのないス画面上に4D Writeエリアを表示させます。

```
If(Form event=On load)  
    WR SET DOC PROPERTY(Area;wr view menubar;0)  
    WR SET DOC PROPERTY(Area;wr view rulers;0)  
End if
```

2. このメソッドでは、スクロールバーを表示または非表示にできます。

```
C_LONGINT(ScrollStatus)  
ScrollStatus:=WR Get doc property(Area;wr Hscrollbar) `定数=7  
ScrollStatus:=ScrollStatus+WR Get doc property(Area;wr Vscrollbar) `定数=8  
If (ScrollStatus>0)  
    CONFIRM("1つ以上のスクロールバーが表示されています。  
                                                    非表示にしますか?")  
    If (OK=1)  
        WR SET DOC PROPERTY(Area;wr Hscrollbar;0)  
        WR SET DOC PROPERTY(Area;wr Vscrollbar;0)  
    End if  
Else  
    CONFIRM("スクロールバーは非表示になっています。表示しますか?")  
    If (OK=1)  
        WR SET DOC PROPERTY(Area;wr Hscrollbar;1)  
        WR SET DOC PROPERTY(Area;wr Vscrollbar;1)  
    End if  
End if
```

参照

WR Get doc property

WR UPDATE MODE

WR UPDATE MODE (エリア; モード)

引数	タイプ	説明
エリア	倍長整数	4D Write エリア
モード	整数	0 = 更新しない 1 = 更新する

説明

WR UPDATE MODE は、<エリア>で指定されたエリアにおけるデザイナーによるスクリーンの更新を可または不可に設定することができます。<モード>が0の場合は、スクリーンの更新はオフになります。<モード>が1の場合は、スクリーンの更新はオンになります。このコマンドは、4D Write コマンドによって生じるスクリーン更新についてのみ影響があります。エリアに対するユーザの操作は、正確にスクリーンを更新し続けます。

スクリーンの更新がオフである場合には、4D Write コマンドの実行速度は速くなります。例えば、4D Write のエリアについて一連の修正処理を実行するのであれば、修正処理の前にスクリーンの更新をオフにして修正処理を実行し、処理終了後にスクリーンの更新をオンに戻します。このコマンドは、スクリーンの再表示と同じくらい高速に実行されません。

例題

次の例は、スクリーンの更新をオフにした後にいくつかの修正処理を行うグローバルメソッドである **Reformat** を呼び出し、スクリーンの更新をオンに戻します。

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

参照

なし

WR Build preview

WR Build preview(エリア; ページ) ピクチャ

引数	タイプ	説明
エリア	倍長整数	4D Write エリア
ページ	倍長整数	ピクチャとして渡されるページ番号
戻り値	ピクチャ	ページのピクチャ

説明

WR Build preview は、`<ページ>` に渡された番号のピクチャを任意のピクチャに変換します。このページ番号は「環境設定」ダイアログで定義されたページの番号付けを考慮しています。

このピクチャは、4Dのピクチャフィールドまたは4Dピクチャ変数に格納することができます。このピクチャはページと同じサイズです。**WR SET DOC PROPERTY** コマンドを使うか、定数の `wr paper width` と `wr paper height` を使うことによって、ピクチャのサイズを調整することができます。

注：このピクチャは、**WR Area to picture** を使用する場合と違って、4D Writeのいかなるデータも含んでいません。

返されるピクチャは、ベクトルベースのピクチャです。Windows上で作られたピクチャはEMFフォーマットを使っているため、Mac OS上で直接表示することはできません。しかし、4DはMac OSピクチャを表示することができるのでその逆は可能です。WindowsのピクチャをMac OS上で表示させる場合は、次のコードを利用してピクチャをビットマップに変換してください。

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

EMF(Windowsのみ)と違って、PICTおよびビットマップのピクチャタイプはプラットフォームに依存しません。

例題

4D Write ドキュメントはBLOBフィールドに保存されます。各ドキュメントの2ページ目だけをプリントしたい場合、プリントフォームに（例えばMyImageという名前の）ピクチャ変数を挿入し、その変数に以下のメソッドを埋め込みます。

```
If (Form event=On Printing Detail)
  WR BLOB TO AREA (NewOffscreen;[MyTable]WriteBlob_)
  MyImage:=WR Build preview (NewOffscreen;2)
End if
```

以下のプロジェクトメソッドを作成し、実行します。

```
QUERY([MyTable])           `プリント範囲を作成
OUTPUT FORM([MyTable];"2ページ目")   `プリントに使用されるフォーム
NewOffscreen:=WR New offscreen area
  `前のメソッドで使用されるオフスクリーンエリアを作成
PRINT SELECTION([MyTable])           `作成した範囲をプリント
WR DELETE OFFSCREEN AREA(NewOffscreen)
  `オフスクリーンエリアを削除
```

参照

WR SET DOC PROPERTY

WR GET AREA PROPERTY

WR GET AREA PROPERTY(エリア;オプション;値;文字値)

引数	タイプ	説明
エリア	倍長整数	4D Write エリア
オプション	整数	オプション番号
値	整数	オプションに依存して、0または1
文字値	文字列	ケースに依存する属性文字列

説明

WR GET AREA PROPERTY を利用すれば、<エリア> で指定された 4D Write エリアに関する様々なオプションを読み込むことができます。

読み取り可能なオプションを下記に示します：

N° オプション(下線付きは定数)	数値	文字	効果
0 確認ダイアログ wr confirm dialog	0 1	- -	ダイアログなし ダイアログ
1 ピクチャのプレビュー wr save preview	0 1	- -	プレビューなし プレビュー
2 操作を保存 wr allow undo	0 1	- -	バッファに溜めない 操作を格納
3 エリアが0の場合を除いて、不正ビット wr modified	0 1	- -	偽(False) 真(True)
4 印刷時の可変サイズ wr fixed print size (エリアが0でない場合)	0 1	- -	可変サイズ 固定サイズ
5 フィールド変換ダイアログ(エリアが0の場合) wr convert dialog	0 1	- -	ダイアログなし ダイアログ
6 エリアを小さくした際のボタンタイトル wr minimized button title	0 1	- タイトル	デフォルトタイトル
7 4D Write のウインドウタイトル (フルスクリーンまたは外部ウインドウ内) wr window title	0 1	- タイトル	エリア名
8 ボタンにスイッチする前のエリアの最小の幅 wr minimum width	XX	-	ピクセル
9 ボタンにスイッチする前のエリアの最小の高さ wr minimum height	XX	-	ピクセル
10 テンプレートを C/S のサーバに保存 wr save template on server	0 1	- -	クライアント上 サーバ上

11 テンプレートをC/Sのサーバからロード	0	-	クライアント上
wr load template on server	1	-	サーバ上

テキストプロパティと、そのリファレンスの一覧は、付録Dの「WR Options」を参照してください。この引数には、定数と値のどちらでも渡すことができます。

参照

「付録D：4D Write 定数」、WR SET AREA PROPERTY

WR GET CURSOR POSITION

WR GET CURSOR POSITION(エリア;ページ;段組;行;位置)

引数	タイプ	説明
エリア	倍長整数	4D Write エリア
ページ	倍長整数	選択部分の属するページ番号
段組	倍長整数	選択部分の段組数
行	倍長整数	段組内の行数
位置	倍長整数	対象となる段組内の選択位置

説明

WR GET CURSOR POSITION は、 <エリア> で指定された 4D Write エリア内の選択された部分の位置を返します。

<ページ>

返されるページは、ドキュメントの最初のページ番号から最後のページ番号までの番号です。これらのページ番号は、必要であればカスタムページ番号も考慮されます。

<段組>

この値は、1 から全段組数までの数値です。

<行>

この値は、1 から段組内の全行数までの数値です。

<位置>

この値は、1 から行内の全文字数までの数値です。

選択された中にいくつかの文字が含まれている場合、最初の文字の位置が返されます。

WR SET CURSOR POSITION でこの値を引数として指定すれば、後でこの位置に戻ることができます。

WR Get frame を使えば、どのエリアにカーソルがあるのかを確認することができます。

例題

テキスト中のカーソルの現在位置を変えずに、ドキュメントのヘッダにロゴを挿入したい場合は、挿入ボタンに下記のメソッドを埋め込みます。

```
C_LONGINT($frame;$Col;$Line;$Pos)
C_REAL($PictWidth;$PictHeight;$OrigWidth;$OrigHeight;$HeadTopMargin)
  `ドキュメントのどのフレームにカーソルがあるか？
$frame:=WR Get frame(Area)
  `現在のカーソルの位置を取得する
WR GET CURSOR POSITION (Area;$Page;$Col;$Line;$Pos)
  `カレントエリアをドキュメントのヘッダに切り替える
WR SET FRAME (Area;wr right header)
  `挿入したいロゴを含むレコードをロードする
ALL RECORDS([Interface])
  `ロゴを挿入
WR INSERT PICTURE(Area,[Interface]Logo;0) `ロゴを選択し、サイズを取得する
WR SELECT(Area;4;1)
WR GET PICTURE SIZE(Area;$PictWidth;$PictHeight;$OrigWidth;$OrigHeight)
  `ヘッダの高さはピクチャに合わせる
$HeadTopMargin:=WR Get doc property(Area;wr header top margin)
WR SET DOC PROPERTY(Area;wr text top margin;$HeadTopMargin+$PictHeight)
WR SET DOC PROPERTY(Area;wr header bottom margin;$PictHeight)
  `カーソルがあるフレームに戻る
WR SET FRAME(Area;$frame)
  `元々の位置にカーソルを配置する
WR SET CURSOR POSITION(Area;$Page;$Col;$Line;$Pos)
```

参照

WR GET CURSOR COORDINATES、WR Get frame、WR SET CURSOR POSITION

WR Get frame

WR Get frame(エリア) 結果

引数	タイプ	説明
エリア	倍長整数	4D Write エリア
戻り値	整数	カーソルのあるページエリア

説明

WR Get frame は、カーソルポイントまたは選択部分が属しているページの番号を返します。

下記は、返される値を示しています：

値	指定位置	定数
0	テキストエリア	wr text frame
1	右のヘッダ	wr right header
2	右のフッタ	wr right footer
3	左のヘッダ	wr left header
4	左のフッタ	wr left footer
5	最初のヘッダ	wr first header
6	最初のフッタ	wr first footer

これらの値を、番号または定数（上記参照）によって入力することができます。

注：この値の一覧については、付録Dの「WR Freams」の項を参照してください。

参照

「付録D：4D Write 定数」、WR SET FRAME

WR SET AREA PROPERTY

WR SET AREA PROPERTY(エリア;オプション;値;文字値)

引数	タイプ	説明
エリア	倍長整数	4D Write エリア
オプション	整数	オプション番号
値	整数	オプションに依存。0または1
文字値	文字列	オプションに依存する属性文字列

説明

WR SET AREA PROPERTY を利用すれば、<エリア>で指定された4D Write エリアに関する様々な属性を変更することができます。<エリア>が0である場合、**WR SET AREA PROPERTY** は、アクティブな4D Write エリアそれぞれに反映されます。このような場合には、「On Startup」データベースメソッドからこのコマンドを呼び出します。

設定可能なオプションを以下に示します：

N° オプション(定数は下線付き)	値		効果
	数値	文字	
0 確認ダイアログ	0	-	ダイアログなし
wr confirm dialog	1	-	ダイアログ
1 ピクチャのプレビュー	0	-	プレビューなし
wr save preview	1	-	プレビュー
2 操作を保存	0	-	バッファに溜めない
wr allow undo	1	-	操作を格納
3 エリアが0の場合を除いて、不正ビット	0	-	偽(False)
wr modified	1	-	真(True)
4 印刷時の可変サイズ	0	-	可変サイズ
wr fixed print size (エリアが0でない場合)	1	-	固定サイズ
5 フィールド変換ダイアログ(エリアが0の場合)	0	-	ダイアログなし
wr convert dialog	1	-	ダイアログ
6 エリアを小さくした際のボタンタイトル	0	-	デフォルトタイトル
wr minimized button title	1	タイトル	
7 4D Write のウィンドウタイトル (フルスクリーンまたは外部ウィンドウ内)	0	-	エリア名
wr window title	1	タイトル	
8 ボタンにスイッチする前のエリアの最小の幅	XX	-	ピクセル
wr minimum width			
9 ボタンにスイッチする前のエリアの最小の高さ	XX		ピクセル
wr minimum height			

10	テンプレートをC/Sのサーバに保存 wr save template on server	0 - 1 -	クライアント上 サーバ上
11	テンプレートをC/Sのサーバからロード wr load template on server	0 - 1 -	クライアント上 サーバ上

注：<オプション>は、定数を使って設定することも可能です。

例題

エリアの自動的なピクチャプレビュー、確認ダイアログの表示、「編集」メニューからの取り消しコマンドを実行不可能にしたいものとします。

WR SET AREA PROPERTY(Area;wr save preview;0)

WR SET AREA PROPERTY(Area;wr confirm dialog;0)

WR SET AREA PROPERTY(Area;wr allow undo;0)

参照

「付録D：4D Write 定数」、WR GET AREA PROPERTY

WR SET CURSOR POSITION

WR SET CURSOR POSITION(エリア;ページ;段組;行;位置)

引数	タイプ	説明
エリア	倍長整数	4D Write エリア
ページ	倍長整数	ページ番号
段組	倍長整数	段組番号
行	倍長整数	行番号
位置	倍長整数	行内のカーソルの水平位置

説明

WR SET CURSOR POSITION は、< ページ >、< 段組 >、< 行 >、< 位置 > で指定された新しい位置にカーソルポイントを移動します。

< ページ >

指定する値は、ドキュメントの最初のページ番号から最後のページ番号までの間の番号を指定する必要があります。このページ番号は「環境設定」ダイアログで定義したページ番号を考慮する必要があります。

< 段組 >

指定する値は、1 から全段組数までの間の数値を指定する必要があります。

< 行 >

指定する値は、1 から段組内の全行数までの間の数値（1つの段組しかない場合はページ）を指定する必要があります。

< 位置 >

指定する値は1から1行の文字数までの間の数値を指定する必要があります。

本文のエリアと別の場所にカーソルを配置したい場合には、**WR SET CURSOR POSITION** を使用する前に **WR SET FRAME** を使用する必要があります。

例題

4ページ目の10行目の行頭に挿入ポイントを移動したいものとします。

```
`ドキュメントのbodyエリアにいることを確認する
If (WR Get frame (Area)#0)
  `違うエリアにいたら body エリアに移動する
  WR SET FRAME (Area;wr_body)
End if
`カーソルの移動
WR SET CURSOR POSITION(Area;10;1;10;1)
`挿入ポイントを表示するためにエリアをスクロールする
WR SCROLL TO SELECTION(Area)
```

参照

WR GET CURSOR POSITION、WR SET FRAME

WR SET FRAME

WR SET FRAME(エリア;フレーム)

引数	タイプ	説明
エリア	倍長整数	4D Write エリア
フレーム	整数	フレーム番号

説明

WR SET FRAMEは、<エリア>によって指定された4D Write エリア内において、<フレーム>によって指示された位置にカーソルポイントを移動します。この位置は、あらかじめ4D Write に記憶されています。通常の表示モードが選択されている場合やカーソルポイントがヘッダエリアまたはフッタエリアのいずれかに配置されている場合、4D Write は自動的に「ページプレビュー」モードに変換します。

<フレーム>には、次の値または定数を渡すことができます：

値	指定位置	定数
0	テキストエリア	wr text frame
1	右のヘッダ	wr right header
2	右のフッタ	wr right footer
3	左のヘッダ	wr left header
4	左のフッタ	wr left footer
5	最初のヘッダ	wr first header
6	最初のフッタ	wr first footer

左ページと右ページに異なったヘッダまたはフッタを使用する際には、3または4という値を利用します。

最初のページに異なったヘッダまたはフッタを使用する際には、5または6という値を利用します。

注：この値の一覧については、付録Dの「WR Frames」を参照してください。

例題

WR GET CURSOR POSITION、WR SET CURSOR POSITION、WR INSERT PAGE NUMBERの例題を参照してください。

参照

WR Get frame

WR TEXT ACCESS

WR TEXT ACCESS(エリア;モード)

引数	タイプ	説明
エリア	倍長整数	4D Write エリア
モード	整数	0 = アクセスを許可 1 = アクセスを制限

説明

WR TEXT ACCESS を利用することによって、エリア内のテキストへのアクセスを制御することができます。 <モード> が0の場合は、4D Writeは<エリア>について修正を許可します。 <モード> が1の場合は、4D Writeは読み取り専用モードで<エリア>を表示します。

エリアが読み取り専用モードで表示されている場合は、メニューやルーラー、およびズームボックスは表示されません。テキストは表示され、スクロールも可能ですが、修正はできません。制限されたエリアへアクセス制限を変更する場合は、**WR DISPLAY RULER** や **WR DISPLAY MENUBAR** を呼び出して、ルーラーとメニューを表示する必要があります。

例題

次の例は、<エリア>で指定されたエリアを含んでいるフォームのフォームメソッドです。フォームがロードされる際に、<エリア>で指定されたエリアを読み取り専用を設定します。

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

参照

WR SET DOC PROPERTY

WR Area to blob

WR Area to blob (エリア{;保存ドキュメント}) 結果

引数	タイプ	説明
エリア	倍長整数	4D Write エリア
保存ドキュメント	整数	0=ドキュメントを保存しない、ダイアログを表示する 1=ドキュメントを保存しない、ダイアログも表示しない
戻り値	BLOB	エリアの内容

説明

WR Area to blob は、<エリア>によって指定されたエリアの内容をBLOBフィールドまたはBLOB変数に代入します。**WR Area to blob** は、BLOBフィールドまたはBLOB変数に指定可能なBLOBを返します。

保存ドキュメント

- ・ <保存ドキュメント> が0の場合、または最後に保存されてからドキュメントが変更されている場合、ドキュメントを保存するかどうかを確認するダイアログが表示されます。
- ・ <保存ドキュメント> が1の場合、ドキュメントは保存済みであると見なされるため、ドキュメントを保存する必要はありません。
- ・ <保存ドキュメント> が省略されている場合は、デフォルトの設定を使用します。

例題

"WriteBlobSave"BLOBフィールドにAreaを保存したいものとします。

```
[Texts]WriteBlobSave:=WR Area to blob(Area;1)
```

参照

WR Area to picture、WR BLOB TO AREA

WR Area to picture

WR Area to picture(エリア;保存ドキュメント;プレビュー) 結果

引数	タイプ	説明
エリア	倍長整数	4D Write エリア
保存ドキュメント	整数	0=ドキュメントを保存しない、ダイアログを表示する 1=ドキュメントを保存しない、ダイアログも表示しない
プレビュー	整数	1=ピクチャを作成する 0=ピクチャを作成しない
戻り値	ピクチャ	エリアの内容のピクチャ

説明

WR Area to pictureは、<エリア>によって指定されたエリアの内容をピクチャフィールドまたはピクチャ変数に代入します。4D Write エリアを**WR Area to picture**に渡すことによって、ピクチャフィールドまたはピクチャ変数に指定可能なピクチャを返します。

保存ドキュメント

- ・ <保存ドキュメント>が0の場合、または最後に保存されてからドキュメントが変更されている場合、ドキュメントを保存するかどうかを確認するダイアログが表示されます。
- ・ <保存ドキュメント>が1の場合、ドキュメントは保存済みであると見なされるため、ドキュメントを保存する必要はありません。

プレビュー

- ・ <プレビュー>が0の場合、ピクチャのプレビューは作成されません。
- ・ <プレビュー>が1の場合、ピクチャのプレビューが作成されます。

注：ピクチャプレビューを作成していない場合、ピクチャは表示されません。

オプション引数が省略されている場合は、デフォルトの設定を使用します。

例題

1. ピクチャは、そのプレビューと同じように、ピクチャフィールド " WritePictSave"の中にAreaを保存したいものとします。

```
[Texts]WritePictSave:=WR Area to picture(Area;1;1)
```

2. [Templates]テーブルのレコード中に現在のテキストの範囲を保存したいものとします。

```
WR EXECUTE COMMAND(Area;wr_cmd_copy) `範囲をコピー  
CREATE RECORD([Templates]) ` [Templates]にレコードを作成  
Tempo:=WR New offscreen area ` オフスクリーンエリアを作成  
WR EXECUTE COMMAND(Tempo;wr_cmd_paste) ` エリアに選択範囲をペースト  
[Templates]Text_:=WR Area to picture(Tempo)  
` [Templates]テキストフィールドに結果を保存  
WR DELETE OFFSCREEN AREA (Tempo) ` テンポラリエリアを削除  
SAVE RECORD([Templates]) ` [Templates]にレコードを保存
```

参照

WR Area to blob、WR PICTURE TO AREA

WR BLOB TO AREA

WR BLOB TO AREA(エリア;blob)

引数	タイプ	説明
エリア	倍長整数	4D Write エリア
blob	Blob	4D Writeのデータを含む変数またはフィールド

説明

WR BLOB TO AREA は、<エリア>によって指定された4D Writeエリアに< blob >の内容を代入します。BLOBの内容は、4D Writeのデータと見なされます。

Blobの内容は、BLOBという名前によって割り当てられた4D Writeエリアから自動的に保存されたデータ、または**WR Area to blob**を使って保存されたデータのどちらにもなり得ます。

例題

1. "[Templates]Reference_"BLOBフィールドに保存された手紙のテンプレートをロードし、カレントテンプレートとして使用したいものとします。

```
QUERY([Templates];[Templates]Texts=Ref)
If(Records in selection([Templates])>0
  WR BLOB TO AREA(Area;[Templates]Reference_)
End if
```

2. "[Templates]TheText_"BLOBフィールドに保存されたテキストをコピーし、スクリーン上のカレントエリアにペーストしたいものとします。この例題は高度な用語システムの作成方法を示しています。

```
Temp:=WR New offscreen area
WR BLOB TO AREA (Temp;[Templates]TheText_) `フィールドの拡張
WR EXECUTE COMMAND(Temp;wr_wmd_select all)
WR EXECUTE COMMAND(Temp;wr_cmd_copy)
WR DELETE OFFSCREEN AREA (Temp) `エリアの削除
WR EXECUTE COMMAND(Area;wr_cmd_paste)`ペーストメニューコマンドの実行
```

注：ピクチャフィールドの中に4D Writeエリアを保存したい場合は、**WR PICTURE TO AREA** コマンドの説明を参照してください。

参照

WR Area to blob

WR DELETE OFFSCREEN AREA

WR DELETE OFFSCREEN AREA(エリア)

引数	タイプ	説明
エリア	倍長整数	4D Write エリア

説明

WR DELETE OFFSCREEN AREA は、**WR New offscreen area** または **WR Picture to offscreen area** によって作成された 4D Write エリアを削除し、オフスクリーンエリアが使用していたメモリを解放します。

<エリア> は、フォーム上やウインドウ内のエリアではなく、オフスクリーンエリアでなければなりません。オフスクリーンエリアが不要になった場合には、**WR DELETE OFFSCREEN AREA** を使用してください。

例題

次の例は、**WR New offscreen area** を呼び出す場合には、対となる **WR DELETE OFFSCREEN AREA** も呼び出す必要があるという例です。

```
NewArea:=WR New offscreen area  
  `新しくオフスクリーンエリアを作成する  
  `何らかの処理を行う  
WR DELETE OFFSCREEN AREA (NewArea)  
  `オフスクリーンエリアを削除する
```

参照

なし

WR New offscreen area

WR New offscreen area 結果

引数	タイプ	説明
		この関数には、引数はありません。
戻り値	倍長整数	4D Writeエリアの参照番号

説明

WR New offscreen areaは、ユーザからは見ることのできない4D Writeエリアのためにメモリを確保します。また、この関数は見ることができないエリアにアクセスするための参照番号を返します。**WR New offscreen area**によって返される値は、4D Writeエリアに対するすべての4D Writeコマンドで使用可能です。

この関数を用いて作成したオフスクリーンエリアは、使い終わったら削除することを忘れないでください。

例題

次の例は、テンポラリのオフスクリーンエリアを作成し、印刷後にそれを削除します。

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

参照

なし

WR PICTURE TO AREA

WR PICTURE TO AREA(エリア;ピクチャ)

引数	タイプ	説明
エリア	倍長整数	4D Write エリア
ピクチャ	ピクチャ	フィールドまたは変数

説明

WR PICTURE TO AREAを使用することにより、4D Writeドキュメントを含んだピクチャ変数またはピクチャフィールドを読み取り、<エリア>によって指定された4D Writeエリアにピクチャ変数またはピクチャフィールドの内容を開くことができます。<エリア>には、表示されているエリアでもオフスクリーンエリアでも指定することができます。

例えば、このコマンドを利用すれば、別のテーブルに保存された4D Writeドキュメントを読み取ることも可能です。

注：このコマンドは、4D Writeのバージョン6.0.xのファイルフォーマットも読み取りません。

例題

1. "[Templates]Reference"ピクチャフィールドに保存した手紙のテンプレートをロードして、カレントテンプレートとして使用したいものとします。

```

QUERY([Templates];[Templates]Reference=Ref)
If(Records in selection([Templates])>0)
    WR PICTURE TO AREA(Area:[Templates]Reference_)
End if
    
```

2. "[Templates]TheText_"ピクチャフィールドに保存されたテキストをコピーし、スクリーン上のカレントエリアにペーストしたいものとします。この例題は高度な用語システムの作成方法を示しています。

```

Temp:=WR New offscreen area
WR PICTURE TO AREA (Temp:[Templates]TheText_) `フィールドの拡張
WR EXECUTE COMMAND(Temp;wr_wmd_select_all)
WR EXECUTE COMMAND(Temp;wr_cmd_copy)
WR DELETE OFFSCREEN AREA (Temp) `エリアの削除
WR EXECUTE COMMAND(Area;wr_cmd_paste)`ペーストメニューコマンドを実行
    
```

注：BLOBフィールドの中に4D Writeエリアを保存した場合は、**WR BLOB TO AREA**コマンドの説明を参照してください。

参照

WR Area to picture

WR GET REFERENCE

WR GET REFERENCE(エリア;テーブル;フィールド;名前;タイプ)

引数	タイプ	説明
エリア	倍長整数	4D Write エリア
テーブル	整数	テーブル番号を取得
フィールド	整数	フィールド番号を取得
名前	文字列	リファレンス名を取得
タイプ	整数	リファレンスタイプを取得

説明

WR GET REFERENCE は、4D Write エリア内で選択されたリファレンスに関する情報を取得します。

選択されたリファレンスに関する情報は、<テーブル>、<フィールド>、<名前>および<タイプ>に返されます。<テーブル>には、テーブル番号が入ります。<フィールド>には、そのリファレンスがフィールドであれば、フィールド番号が返されず。<名前>は、リファレンスが式やホットリンクである場合は、式のテキストやホットリンクの名前が返ります。<テーブル>、<フィールド>、<名前>に返される値は、<タイプ>によって変わります。選択されたオブジェクトがリファレンスでない場合、<タイプ>には0が返されます。

タイプ	説明
1	リファレンスはフィールドです。引数<テーブル>は、テーブル番号を示します。引数<フィールド>は、フィールド番号を示します。引数<名前>は空です。
2	リファレンスは式です。引数<テーブル>と<フィールド>には、0が返されます。引数<名前>には、変数または式の名前が返されます。
3	リファレンスはホットリンクです。引数<テーブル>と<フィールド>には、0が返されます。引数<名前>には、ホットリンクの名前が返されます。

例題

次の例は、リファレンスであるオブジェクトをユーザが選択しているかどうかを判定します。また選択されたオブジェクトについて、フィールド、ホットリンク、式のどれであるのかを知らせてくれます。

WR GET REFERENCE (Letter;\$Table;\$Field;\$Name;\$Type)

Case of

¥ (\$Type=0) `テキスト、または選択されていない

ALERT("テキスト、または選択されていません")

¥ (\$Type=1)

ALERT("フィールドが選択されています : "+Fieldname(\$テーブル;\$フィールド))

¥ (\$Type=2)

ALERT("変数または式が選択されています : "+\$名前)

¥ (\$Type=3)

ALERT("ホットリンク名が選択されています : "+\$名前)

End case

参照

なし

WR INSERT EXPRESSION

WR INSERT EXPRESSION(エリア;式)

引数	タイプ	説明
エリア	倍長整数	4D Write エリア
式	文字	挿入する式

説明

WR INSERT EXPRESSION は、<エリア> に式の参照を挿入し、現在選択されているテキストと置き換えます。

式は、値を返す正しい 4th Dimension の式でなければなりません。<式> は、4th Dimension の変数や関数、または値を返すステートメントです。<式> は、外部関数やユーザ定義の関数（プロセスメソッド）でも構いません。<式> がキャリッジリターンやタブを含むテキストを返す場合、4D Write はその<式>がある段落のルーラーに従ってテキストをフォーマットします。

例題

以下の2つの部分に別れた例は、4D Write エリアに 4th Dimension のプロセスメソッドへの参照を挿入した例です。

前半に示した例のプロセスメソッド"SHOW INVOICES"は、顧客に関連したインボイスを探し、インボイス番号と金額を連結します。

```

`プロジェクトメソッド 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
NEXT RECORD ([Invoices])
End for

```

後半に示した例は、プロセスメソッド SHOW INVOICES をエリアに挿入します。4D Write がエリアを表示または印刷する際に、各インボイスは各行に表示されます。

```
WR INSERT EXPRESSION (Area;"SHOW INVOICES")
```

参照

WR INSERT FIELD

WR INSERT FIELD

WR INSERT FIELD(エリア;テーブル;フィールド)

引数	タイプ	説明
エリア	倍長整数	4D Write エリア
テーブル	整数	テーブル番号
フィールド	整数	フィールド番号

説明

WR INSERT FIELD は、<エリア>にフィールドへの参照を挿入し、現在選択されているテキストと置き換えます。挿入されるフィールドは、<テーブル>と<フィールド>に対して、テーブル番号とフィールド番号で指定します。

WR INSERT EXPRESSIONと異なり、**WR INSERT FIELD**は、ダイナミック参照を挿入します。すなわち、フィールドへの変更は、ドキュメントの参照されている内容に直ちに反映されます。

参照

WR INSERT EXPRESSION

WR Insert picture area

WR Insert picture area(エリア;ピクチャ;場所) エラー

引数	タイプ	説明
エリア	倍長整数	4D Write エリア
ピクチャ	ピクチャ	挿入する 4D Write エリアピクチャ
場所	整数	0= 挿入ポイント 1= ドキュメントの最後
戻り値	倍長整数	エラーコード

説明

WR Insert picture area は、ピクチャにある 4D Write ドキュメントを <エリア> に挿入します。

<場所> は、テキストがどこに挿入されるかを示します。

<場所> が 0 ならば、テキストは現在の挿入ポイントに挿入されるか、現在選択されているテキストと置き換えられます。

<場所> が 1 ならば、<ピクチャ> にあるテキストはドキュメントの最後に挿入されます。

WR Insert picture area は、エラーコードを含む倍長整数を返します。関数が正常終了すると、値は 0 になります。エラーコードは、付録 C の「エラーコード」を参照してください。

例題

次の例では、送り主の署名をドキュメントの最後に挿入します。

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

参照

なし

WR GET DOCUMENT INFO

WR GET DOCUMENT INFO(エリア;題名;件名;著者;組織;ノート;作成日;作成時刻;修正日;修正時刻;ロック)

引数	タイプ	説明
エリア	倍長整数	4D Writeエリア
題名	文字列	ドキュメントのタイトル
件名	文字列	ドキュメントの主題
著者	文字列	ドキュメントの著者
組織	文字列	組織名
ノート	テキスト	ドキュメントのノート
作成日	日付	作成した日
作成時刻	時間	作成した時刻
修正日	日付	修正した日
修正時刻	時間	修正した時刻
ロック	整数	0=ロックされていない 1=ロックされている

説明

WR GET DOCUMENT INFOを使うことで、「ドキュメント情報」ダイアログに表示されているドキュメント情報を取得することができます。「ドキュメント情報」ダイアログは「ツール」メニューから「ドキュメント情報」を選択することで表示されます。

ドキュメント情報、著者名、組織名、ノートなどのいくつかのドキュメント情報は、**WR SET DOCUMENT INFO**を使って設定することができます。

<ロック>は、**WR LOCK DOCUMENT**で設定することができます。これは、ユーザによるドキュメントの変更を防止する論理的なロックです。ロックはペーストやカット、テキスト入力、修正または属性の変更といったユーザによる操作に影響を及ぼします。ドキュメントの読み取りやテキストのコピー、文字の検索、ドキュメントの印刷といったユーザの操作は行うことができます。

< 作成日 >、< 作成時刻 >、< 修正日 >、< 修正時刻 > は、ドキュメントが保存された際に、4D Write によって自動的に更新されます。

例題

WR SET DOCUMENT INFO コマンドの例題を参照してください。

参照

WR SET DOCUMENT INFO

WR LOCK DOCUMENT

WR LOCK DOCUMENT(エリア;ステータス)

引数	タイプ	説明
エリア	倍長整数	4D Write エリア
ステータス	整数	0=ロックされていない 1=ロックされている

説明

WR LOCK DOCUMENT は、<エリア>によって参照される4D Write エリアのユーザによる変更を禁止します。ドキュメントが一旦ロックされると、ユーザはテキストのペースト、カット、更新ができなくなります。テキストのスクロール、コピー、検索、印刷は引き続き可能です。

現在のドキュメントのロックステータスを知るには、**WR GET DOCUMENT INFO** を使用します。この情報は「ドキュメント情報」ダイアログでも表示されます。「ツール」メニューから「ドキュメント情報」を選択することでも表示できます。

ステータス

- ・ <ステータス> が0の場合は、ドキュメントはロックされていません。
- ・ <ステータス> が1の場合は、ドキュメントはロックされています。

例題

最終的なレコードを閉じ、その後ユーザが編集するのを禁止したいものとします。

```

`ドキュメントの編集は不可
WR LOCK DOCUMENT(Area;1)
`ダイアログボックスを開くために「ツール」メニューコマンドから
`「ドキュメント情報」を選択することはできないが、オプションは有効
WR LOCK COMMAND(Area;wr cmd doc information;0)

```

参照

WR LOCK COMMAND

WR OPEN DOCUMENT

WR OPEN DOCUMENT(エリア;ドキュメント{;タイプ})

引数	タイプ	説明
エリア	倍長整数	4D Write エリア
ドキュメント	文字列	開くドキュメントの名前
タイプ	文字列	開くドキュメントのタイプ (4文字) 開いたドキュメントのタイプ (4文字)

説明

WR OPEN DOCUMENT は、<ドキュメント> で指定されたドキュメントを開き、<エリア> で参照された 4D Write エリアに配置します。

<ドキュメント> は、ドキュメントファイルへのアクセスパスです。Windows 上では、ファイル拡張子を含める必要があります。

例題

- Windows 上では、フォルダの間に、“¥” が含まれている必要があります。

```
"D:¥folder1¥folder2¥file.4W7"
```

- Mac OS 上では、フォルダの間に、“:” が含まれている必要があります。

```
"MacintoshHD:folder:document"
```

<ドキュメント> が空の場合、**WR OPEN DOCUMENT** は標準の「ファイルを開く」ダイアログを表示します。「ファイルを開く」ダイアログで「開く」ボタンがクリックされるとシステム変数 OK に 1 がセットされ、<ドキュメント> 変数にユーザが選択したファイルのアクセスパスがセットされます。

この場合、<タイプ> には、ユーザがドロップダウンリストで選択したタイプか、何も選択しなかった場合はドキュメントタイプが返されます。

「キャンセル」ボタンがクリックされると、<ドキュメント> には空の文字列が返り、システム変数 OK には 0 がセットされます。

このコマンドでは、以下のファイルフォーマットがサポートされます：

タイプ	ファイル
4WR7	4D Write 6.5
4WR6	4D Write 6
4WT7	4D Write 6.5テンプレート
RTF	RTF ファイル
ASCW	Windows テキストファイル
ASCM	MacOS テキストファイル
HTML	HTML テキストファイル

例題

以下の例は、データベースフォルダにあるファイルを開きます。

WR OPEN DOCUMENT(Area;"HD:Folder:database folder:file")
WR OPEN DOCUMENT(Area;"D:¥folder¥BaseFolder¥file.4W7")

`Mac OS 上
`Windows 上

参照

WR SAVE DOCUMENT

WR SAVE DOCUMENT

WR SAVE DOCUMENT(エリア;ドキュメント;タイプ)

引数	タイプ	説明
エリア	倍長整数	4D Write エリア
ドキュメント	文字	作成するドキュメントの名前
タイプ	文字	ドキュメントのタイプ

説明

WR SAVE DOCUMENT は、<ドキュメント>に指定されたアクセスパスを使って、<エリア>で指定される4D Write エリアの中にドキュメントを保存します。

<ドキュメント>は、ドキュメントファイルへのアクセスパスです。Windows上では、ファイルの拡張子を含める必要があります。

例題

- ・ Windows上では、フォルダの間に円マーク(¥)が含まれている必要があります。

```
"D:¥folder1¥folder2¥file.4W7"
```

- ・ Mac OS上では、フォルダの間にコロン(:)が含まれている必要があります。

```
"MacintoshHD:folder:document"
```

ドキュメント文字列が空なら、**WR SAVE DOCUMENT**は標準の「ファイル保存」ダイアログを表示します。「ファイル保存」ダイアログで「保存」(MacOS)または「OK」(Windows)ボタンがクリックされると、システム変数OKに1が設定され、<ドキュメント>変数にユーザが選択したファイルのアクセスパスが設定されます。

この場合、<タイプ>には、ユーザがドロップダウンリストで選択したタイプ、または何も選択しなかった場合はドキュメントタイプが返されます。

「キャンセル」ボタンがクリックされると、<ドキュメント>には空の文字列が返り、システム変数OKには0がセットされます。

このコマンドでは、以下のファイルフォーマットがサポートされます：

タイプ	ファイル
4WR7	4D Write 6.5
4WR6	4D Write 6
4WT7	4D Write 6.5 テンプレート
RTF	RTF ファイル
ASCW	Windows テキストファイル

ASCM	MacOS テキストファイル
ASCU	Unicode テキストファイル
ASCN	MacOS 上では MacOS テキストファイル、Windows 上では Windows ファイル

注：これらのタイプは仮想タイプで、MacOS ファイルタイプや Windows の拡張子との関係はありません。

例題

1. 4D Write ファイルタイプの 'LetterClient' というファイル名でドキュメントを保存したい場合の例を示します。このドキュメントはデータベースストラクチャファイルと同じ階層に位置する "WriteDocuments" フォルダに保存されます。

```

`データベースストラクチャファイルのフルパス名を取得
$Doc:=Structure file
Doc:=$Doc
$Long:=0
`フルパス名からストラクチャ名を取り除くために
`最後のセパレーートの位置を取得
While((Position(":;",$Doc)#0)
    $Doc:=Substring($Doc;1+Position(":;",$Doc);Length($Doc))
    $Long:=Length($Doc)
End while
`ドキュメントのフルパス名を作成するために名前を連結する
`ドキュメント名に拡張子を追加し、
`クロスプラットフォームで使用できるようにする
Doc:=Substring(Doc;1;Length(Doc)-$Long)+"WriteDocuments:LetterClient.4W7"
WR SAVE DOCUMENT(Area;doc;"4WR7")

```

2. 保存するドキュメントの名前とタイプの両方を選択し、その値をもう一度取得したいものとします。

```

DocName:=""
DocType:=""
WR SAVE DOCUMENT (Area;DocName;DocType)
If (OK=1)
... `DocName と DocType の値を使用
End if

```

参照

WR OPEN DOCUMENT

WR SET DOCUMENT INFO

WR SET DOCUMENT INFO(エリア;題名;件名;著者;組織;ノート)

引数	タイプ	説明
エリア	倍長整数	4D Write エリア
題名	文字列	ドキュメントの題名
件名	文字列	ドキュメント件名
著者	文字列	ドキュメントの著者
組織	文字列	組織名
ノート	テキスト	ノート

説明

WR SET DOCUMENT INFO は、引数に渡された情報をドキュメントに保存します。ユーザはこれらの情報を「ドキュメント情報」ダイアログで参照することができます。「ツール」メニューから「ドキュメント情報」を選択することで表示できます。

ドキュメントのロックステータスを管理するには、**WR LOCK DOCUMENT** コマンドの説明を参照してください。

例題

「ドキュメント情報」でユーザが編集できる箇所を、「題名」、「件名」、「ノート」に限定したい場合、メニューコマンドのセレクションに割り込むメソッドを実行し、ユーザが「ツール」メニューからドキュメント情報を選択した時にカスタマイズされたフォームを表示します。

1. 4D Write エリアを含むフォームのフォームメソッドで、メニューコマンドに割り込むためのコードです。

Case of

```
:(Form event=On Load)
```

```
WR ON COMMAND(WArea;"z65OnCmd")
```

End case

2. z65OnCmd メソッドは以下の通りです。

C_LONGINT(\$1;\$2;\$3)

Case of

:(\$2=wr cmd doc information) `=801, 「ツール」 から 「ドキュメント情報」
`を選択した場合

DIALOG([TheTable];"InfoArea") `カスタム情報フォーム

Else

WR EXECUTE COMMAND(\$1;\$2) `メニューコマンド以外を
`選択した場合

End case

3. vTitle、vSubject、vComments の変数だけが編集できるように、カスタマイズされた情報フォームで "InfoArea" という名前を付けます。このフォームに埋め込まれたメソッドを示します。

Case of

:(Form event=On Load)

WR GET DOCUMENT INFO(WArea;vTitle;vSubject;vAuthor;vCy;
vComments;DCreat;HCreat;DModif;HModif;Lock)

`必要であれば、空の要素を使用

If (vCy="")

vCy:="A.C.I."

vAuthor:=Current user

vCreation:=**String**(DCreat)+" at "+**Time string**(HCreat)

vModification:=**String**(DModif)+" at "+**Time string**(HModif)

End if

:(Form event=On Unload) `フォームが閉じられた時

WR SET DOCUMENT INFO(WArea;vTitle;vSubject;vAuthor;vCy;
vComments)

End case

参照

WR GET DOCUMENT INFO

WR DELETE PICTURE IN PAGE

WR DELETE PICTURE IN PAGE (エリア;ピクチャ番号)

引数	タイプ	説明
エリア	倍長整数	4D Writeエリア
ピクチャ番号	倍長整数	ピクチャの番号

説明

WR DELETE PICTURE IN PAGE は、<エリア>で指定された4D Writeエリア内の<ピクチャ番号>で指定された番号のピクチャを削除します。**WR DELETE PICTURE IN PAGE**を正確に動作させるためには、ピクチャがテキスト内ではなくページ内に配置されている必要があります。テキスト内のピクチャを削除するためには、目的のピクチャを選択して**WR DELETE SELECTION**を呼び出してください。

WR count(Area;13)を使用して、エリア内のピクチャの数を取得することができます。ピクチャを削除すると、後に続くピクチャの番号からそれぞれ1が引かれます。**WR Get selected picture**を使って、ピクチャ番号を取得することもできます。

例題

以下の例は、指定されたエリアのページに配置されたすべてのピクチャを削除します。

```

$NbOccurrence:=WR Count (Area;13)
For ($i;1;$NbOccurrence)
    `常に最初のピクチャが削除される
    WR DELETE PICTURE IN PAGE (Area;1)
End for
    
```

参照

なし

WR GET PICTURE IN PAGE INFO

WR GET PICTURE IN PAGE INFO (エリア;ピクチャ番号;ページ;背面;先頭ページ;
水平位置;垂直位置;幅;高さ;元の幅;元の高さ)

引数	タイプ	説明
エリア	倍長整数	4D Write エリア
ピクチャ番号	倍長整数	ピクチャ番号
ページ	倍長整数	ピクチャの場所
背面	整数	0=ピクチャはテキストの前面 1=ピクチャはテキストの背面
先頭ページ	整数	0=ピクチャはすべてのページに表示 1=ピクチャは最初のページ以外の全ページに表示
水平位置	実数	ページ内の水平位置
垂直位置	実数	ページ内の垂直位置
幅	実数	ピクチャの現在の幅
高さ	実数	ピクチャの現在の高さ
元の幅	実数	ピクチャの元の幅
元の高さ	実数	ピクチャの元の高さ

説明

WR GET PICTURE IN PAGE INFO は、<エリア>で指定された4D Writeエリア内に現在表示されているピクチャについて、<ピクチャ番号>を指定してピクチャの情報を取得することができます。

警告：このコマンドは、テキストの一部であるピクチャには使用しないでください。

<ページ>

ピクチャがどのページに表示されているかを知ることができます。

- ・ -1より大きい場合、返された数値のページにピクチャが表示されます。
この値は、現在定義されているページ番号付けも考慮します。
- ・ -1の場合、ピクチャはすべてのページに表示されます。
- ・ -2の場合、ピクチャはすべての右ページに表示されます。
- ・ -3の場合、ピクチャはすべての左ページに表示されます。

<背面>

- ・ 0の場合、ピクチャはテキストの前面にあります。
- ・ 1の場合、ピクチャはテキストの背面にあります。

<先頭ページ>

- ・ 0の場合、ピクチャはすべてのページに表示されます。
- ・ 1の場合、ピクチャは先頭ページを除くすべてのページに表示されます。

<水平位置>および<垂直位置>は、ページの左隅の座標を返します。<元の幅>および<元の高さ>には、変更される前のピクチャの元のサイズを返します。

ピクチャのサイズが変更されていないければ、<元の幅>および<元の高さ>は、<幅>および<高さ>と同じ値を返します。これらの値は、ドキュメントの現在のデフォルトの単位で表されます。

注：現在の単位を計算用にピクセルに変更すると便利です。

例題

WR SET PICTURE IN PAGE INFO コマンドの例題を参照してください。

参照

WR DELETE PICTURE IN PAGE、WR GET PICTURE SIZE、WR SET PICTURE IN PAGE INFO

WR GET PICTURE SIZE

WR GET PICTURE SIZE (エリア;幅;高さ;元の幅;元の高さ)

引数	タイプ	説明
エリア	倍長整数	4D Write エリア
幅	実数	ピクチャの現在の幅
高さ	実数	ピクチャの現在の高さ
元の幅	実数	ピクチャの元の幅
元の高さ	実数	ピクチャの元の高さ

説明

WR GET PICTURE SIZE を利用すれば、選択されたピクチャについてサイズに関する情報を取得することができます。この場合、選択されたピクチャはテキスト内に配置されている必要があります。ページ内に埋め込まれているピクチャのサイズ情報を取得するためには、**WR GET PICTURE IN PAGE INFO** を使用してください。**WR GET PICTURE SIZE** を正常に動作させるには、ピクチャが1つだけ選択されている必要があります。

<幅> は、ピクチャの幅を意味します。これは、ドキュメントの現在のデフォルトの単位で表されます。

<高さ> は、ピクチャの高さを意味します。これは、ドキュメントの現在のデフォルトの単位で表されます。

<元の高さ> と <元の幅> は、それぞれピクチャのサイズが変更される前の元の高さと幅のことを指しています。<元の高さ> と <元の幅> が、<高さ> と <幅> とそれぞれ一致している場合は、ピクチャはサイズを変更されていません。<元の高さ> と <元の幅> は、現在のドキュメントの単位で表されます。

注：ピクチャを選択する際には、**WR SELECT** を利用することができます。

例題

WR INSERT PICTURE コマンド、**WR GET CURSOR POSITION** コマンドの例題を参照してください。

参照

WR GET PICTURE IN PAGE INFO、**WR SET PICTURE SIZE**

WR Get selected picture

WR Get selected picture(エリア;ステータス) 結果

引数	タイプ	説明
エリア	倍長整数	4D Write エリア
ステータス	倍長整数	ピクチャのステータス
戻り値	ピクチャ	選択したピクチャ

説明

WR Get selected picture は、<エリア> で指定された 4D Write エリア内で現在選択されているピクチャのコピーを返します。

<ステータス> は、以下の値を返すことが可能です：

- ・ -1 の場合は、ピクチャは選択されていません。
- ・ 0 の場合は、選択されたピクチャはテキスト内にあります。
- ・ 0 の場合は、選択されたピクチャはページ内にあります。

<ステータス> は、**WR GET PICTURE IN PAGE INFO** や **WR SET PICTURE IN PAGE INFO**、**WR DELETE PICTURE IN PAGE** を使用する際にピクチャの確認に有効です。

例題

WR SET PICTURE IN PAGE INFO コマンドの例題を参照してください。

参照

なし

WR INSERT PICTURE

WR INSERT PICTURE(エリア;ピクチャ;挿入先;水平位置;垂直位置;背面;先頭ページ)

引数	タイプ	説明
エリア	倍長整数	4D Write エリア
ピクチャ	ピクチャ	挿入するピクチャ
挿入先	倍長整数	挿入する位置
水平位置	実数	ページ内の水平位置
垂直位置	実数	ページ内の垂直位置
背面	整数	0 = ピクチャはテキストの前面 1 = ピクチャはテキストの背面
先頭ページ	整数	0 = ピクチャはすべてのページに表示 1 = ピクチャは最初のページ以外の全ページに表示

説明

WR INSERT PICTURE は、 <エリア> で指定された 4D Write エリア内のピクチャを <挿入先>、 <水平位置>、 <垂直位置> で指定された位置に挿入します。

<ピクチャ> は、ピクチャフィールドまたはピクチャ変数を指定可能です。

オプションの <挿入先> は、ピクチャの挿入先を定義することができます。

ピクチャをテキストに挿入する場合は、 <挿入先> に 0 を渡すか引数を省略してください。この場合、他の引数は使用されず、ピクチャはカーソルポイントの位置に挿入されるか、選択部分に置き換えられます。

ピクチャをページ内に挿入する場合は、以下のオプションの内の 1 つを使用します：

- ・ <挿入先> が 0 より大きい場合、ピクチャは <挿入先> で指定した番号のページ内に表示されます。挿入先として指定する値は、「環境設定」ダイアログで定義されたページ番号付けを考慮する必要があります。
- ・ <挿入先> が -1 の場合、ピクチャはすべてのページ内においてテキストの下に表示されます。
- ・ <挿入先> が -2 の場合、ピクチャはすべての右ページ内においてテキストの下に表示されます。
- ・ <挿入先> が -3 の場合、ピクチャはすべての左ページ内においてテキストの下に表示されます。
- ・ <挿入先> が -4 の場合、ピクチャはカーソルポイントのあるページ内においてテキストの下に見ることができます。

オプションの<水平位置>および<垂直位置>は、ドキュメントの現在のデフォルトの単位で表されます。この2つの引数は、ページの左上隅と関係したピクチャの左上隅の座標を設定します。

オプション<背面>は、ピクチャをテキストの背面に置くか前面に置くかを定義することができます。

- ・ <背面>の値が1の場合は、ピクチャはテキストの背面にあります。この場合、テキストおよび段落のバックグラウンドの属性に注意する必要があります。“なし”を選択すれば、テキストの背面にあるピクチャを見ることができます。
- ・ <背面>の値が0の場合は、ピクチャはテキストの前面にあります。

オプション<先頭ページ>は、<挿入先>の値が-1、-2または-3の場合にのみ使用します。

- ・ <先頭ページ>が1の場合、ピクチャは最初のページに表示されません。
- ・ <先頭ページ>が0の場合、ピクチャは最初のページに表示されます。

参照

なし

WR SELECT PICTURE IN PAGE

WR SELECT PICTURE IN PAGE (エリア;ピクチャ番号)

引数	タイプ	説明
エリア	倍長整数	4D Write エリア
ピクチャ番号	倍長整数	ピクチャ番号

説明

WR SELECT PICTURE IN PAGEは、<ピクチャ番号>で渡された番号のピクチャを選択することができます。プロパティを操作するコマンドにおいては、このピクチャは（テキスト中ではなく）ページ内に配置されている必要があります。テキスト中に配置されているピクチャを選択したい場合は**WR SELECT**を使用します。

例題

WR SET PICTURE IN PAGE INFO コマンドの例題を参照してください。

参照

WR GET PICTURE IN PAGE INFO、WR INSERT PICTURE、WR SELECT

WR SET PICTURE IN PAGE INFO

WR SET PICTURE IN PAGE INFO (エリア;ピクチャ番号;ページ;背面;
先頭ページ;水平位置;垂直位置;幅;高さ)

引数	タイプ	説明
エリア	倍長整数	4D Write エリア
ピクチャ番号	倍長整数	ピクチャ番号
ページ	倍長整数	ピクチャの場所
背面	整数	0 = ピクチャはテキストの下 1 = ピクチャはテキストの上
先頭ページ	整数	1 = ピクチャは最初のページには無い その他の場合は0
水平位置	実数	ページ内の水平位置
垂直位置	実数	ページ内の垂直位置
幅	実数	現在のピクチャの幅
高さ	実数	現在のピクチャの高さ

説明

WR SET PICTURE IN PAGE INFO は、 <ピクチャ番号> で指定された番号のピクチャの属性を変更することができます。

警告：このコマンドは、テキストフロー内に挿入されたピクチャには使用しないでください。

<ページ>

ピクチャをどのページに表示するかを定義することができます。そのためには、 <ページ> にページ番号を渡してください。このページ番号は、「環境設定」ダイアログで設定されているページ番号付けを考慮する必要があります。

- ・ -1を指定した場合、ピクチャはすべてのページに表示されます。
- ・ -2を指定した場合、ピクチャはすべての右ページに表示されます。
- ・ -3を指定した場合、ピクチャはすべての左ページに表示されます。
- ・ -4を指定した場合、以前の値を変更しません。

<背面>

- ・ 0を指定した場合、ピクチャはテキストの前面にあります。
- ・ 1を指定した場合、ピクチャはテキストの背面にあります。背景色をあらかじめ選択していない場合、このテキストは透明の背景を持つこととなります。

<先頭ページ>

- ・ 0を指定した場合、ピクチャはすべてのページに表示されます。
- ・ 1を指定した場合、ピクチャは先頭ページを除くすべてのページに表示されます。

<水平位置>および<垂直位置>は、物理的ページの左上隅と関係した左上隅の水平座標と垂直座標を設定することができます。<水平位置>の値は、0からページの全幅までの間で設定することが可能です。

注：「ユーザ」モードにおいてピクチャをペーストする場合、プリンタのマージンを考慮に入れてください。

<幅>および<高さ>を用いることによって、寸法を設定することができます。これらの値は、ドキュメントの現在のデフォルトの単位で表されます。

注：次の引数に-1を指定すると、それぞれの初期値は変更されません：

<背面>、<先頭ページ>、<水平位置>、<垂直位置>、<幅>、<高さ>

例題

各ドキュメントのヘッダに同じピクチャを挿入したいものとします。

```
C_REAL($PosHoriz;$PosVert;$PictWidth;$PictHeight;$OrigWidth;$OrigHeight;
                                             $TxtMgTop;$HeadMgBottom)

WR SET DOC PROPERTY(Area;wr_view_mode;0)
$PosHoriz:=WR Get doc property (Area;wr_text_left_margin)
$PosVert:=WR Get doc property(Area;wr_header_top_margin)
ALL RECORDS([Interface])
    ピクチャを挿入
WR INSERT PICTURE(Area;[Interface]Logo;-1;$PosHoriz;$PosVert;1;0)
    `ピクチャはロゴフィールドに保存される
WR SELECT PICTURE IN PAGE(Area;1) `ピクチャを選択
    `ピクチャのプロパティを取得
MyPict:=WR Get selected picture(Area;$NumPict)
WR SET PICTURE IN PAGE INFO(Area;$NumPict;$Page;$Behind;$PageOne;
    $PosHoriz;$PosVert;$PictWidth;$PictHeight;$OrigWidth;$OrigHeight)
    `ピクチャサイズを50%に縮小
$PictHeight:=$PictHeight*1/2
$PictWidth:=$PictWidth*1/2
WR SET PICTURE IN PAGE
INFO(Area;$NumPict;$Page;$Behind;$PageOne;$PosHoriz;$PosVert;$PictWidth;
    $PictHeight)
    `ヘッダ部分がロゴを挿入できるサイズかどうかをチェックする
$TxtMgTop:=WR Get doc property(Area;wr_text_top_margin)
$HeadMgBottom:=WR Get doc property(Area;wr_header_bottom_margin)
```

WR SET DOC PROPERTY(Area;wr text top margin;\$PosVert+\$PictHeight+\$TxtMgTop+\$HeadMgBottom)
WR SET DOC PROPERTY(Area;wr header bottom margin;\$PosVert+\$PictHeight)

参照

WR GET PICTURE IN PAGE INFO

WR SET PICTURE SIZE

WR SET PICTURE SIZE (エリア;幅;高さ)

引数	タイプ	説明
エリア	倍長整数	4D Write エリア
幅	実数	ピクチャの新しい幅
高さ	実数	ピクチャの新しい高さ

説明

WR SET PICTURE SIZEを使用することにより、<エリア>で指定された4D Write エリア内の選択されたピクチャのサイズを変更することができます。

このコマンドは、バックグラウンドのピクチャには影響を及ぼしません。バックグラウンドのピクチャのサイズを変更するには**WR SET PICTURE IN PAGE INFO**を使用します。

<幅>および<高さ>は、ドキュメントの現在のデフォルトの単位で表されます。これらの値は、ページ内、複数の段組を使用する場合は段組内で指定します。

単位としてピクセルを使用する場合は、ドキュメント用の現在のデフォルトの単位を一時的に変更することや、**WR SET PICTURE SIZE**を実行した後に元の設定に戻すことができます。

例題

WR INSERT PICTURE コマンドの例題を参照してください。

参照

WR GET PICTURE SIZE

WR INSERT PAGE NUMBER

WR INSERT PAGE NUMBER (エリア;フォーマット)

引数	タイプ	説明
エリア	倍長整数	4D Writeエリア
フォーマット	整数	フォーマットタイプ

説明

WR INSERT PAGE NUMBERを使用すれば、現在のページ番号を表示しているリファレンスをカーソルの位置に挿入することができます。このリファレンスは、本文やフッタエリア、またはヘッダエリアに配置することができます。**WR SET FRAME**を使って、選択したエリアの任意の場所にカーソルを配置することができます。

<フォーマット>を使用して、ページ番号用の表示フォーマットを選択することが可能です。これらのフォーマットは「ページ番号挿入」ダイアログで利用するフォーマットと全く同じものです。

フォーマットタイプ	値
123	0
abc	1
ABC	2
ローマン文字 (小文字)	3
ローマン文字 (大文字)	4

例題

下記のメソッド (OddPages) は、カレントドキュメントのフッタに挿入される変数に埋め込みます。

```
`「左右ページと異なる」モードがアクティブになっているかどうかを確認
If(WR Get doc property(Area;wr opposite headers)#1)
    `そうでなければ、このモードをアクティブにする
    WR SET DOC PROPERTY(Area;wr opposite headers;1)
    ALERT("警告：ドキュメントは、現在「左右ページと異なる」モードです!")
End if
    `左のフッタにカーソルを設定する
    WR SET FRAME(Area;wr left footer)
    `大文字のローマ数字でページ番号を挿入
    WR INSERT TEXT(Area;"Page")
    WR INSERT PAGE NUMBER(Area;4)
    WR INSERT TEXT(Area;" on ")
    WR INSERT EXPRESSION(Area;"WR Count(Area;11)")
```

参照

WR SET FRAME

WR PRINT

WR PRINT (エリア;モード;コピー数)

引数	タイプ	説明
エリア	倍長整数	4D Write エリア
モード	整数	0 = 値 1 = リファレンス
コピー数	整数	プリントするコピー枚数

説明

WR PRINT は、<エリア> で指定されたエリア内のドキュメントを印刷します。このコマンドは、「ファイル」メニューから「プリント」を選択した場合と同じ処理を行います。

WR PRINT は、<エリア> で指定されたエリアを一度だけ印刷します。範囲内の各レコードにおいて、<エリア> で指定されたエリアを一度印刷したい場合は **WR PRINT MERGE** を使用します。

<モード>

- ・ 1 の場合、リファレンス要素は 4D Write エリア内で左右の 2 重括弧 (<<>>) の間に表示されます。
- ・ 0 の場合、リファレンス要素の値は 4D Write エリア内に印刷されます。

<コピー数>

プリントするコピー部数を管理する引数です。

例題

以下は、<エリア> を含むフォーム上のボタンで使用されるスクリプトの例です。このボタンをクリックすると、<エリア> で指定されたエリア内が印刷されます。

```
WR EXECUTE COMMAND (area;wr cmd compute references)
WR PRINT (Area;0;1)
```

参照

WR PRINT MERGE

WR PRINT MERGE

WR PRINT MERGE (エリア;テーブル;表示)

引数	タイプ	説明
エリア	倍長整数	4D Write エリア
テーブル	整数	ファイル番号
表示	整数	「差し込みプリント」ダイアログボックスの表示 / 非表示

説明

WR PRINT MERGE は、<テーブル>の範囲内の各レコードについて、エリアに含まれるドキュメントを1回ずつ印刷します。<テーブル>は、マージするテーブルの番号です。<テーブル>が0の場合、**WR PRINT MERGE**は標準の「差し込みプリント」ダイアログボックスを表示し、テーブルの指定、またはそのテーブルについてレコードの範囲を変更することができます。

<表示>が0の場合、「差し込みプリント」ダイアログボックスは表示されません。

<表示>が1の場合には、「差し込みプリント」ダイアログボックスが表示されます。

例題

以下は、[Clients]テーブルの各レコードについて手紙を印刷する例です。この手紙は、[手紙]テーブルに保存されています。

```
ALL RECORDS ([Clients])
  `すべての顧客を選択
QUERY ([Letters];[Letters]Ref="Expedite")
  ` "Expedite"テンプレートを参照
Temp:=WR Picture to offscreen area (Temp;[Letters]Doc_)
  ` オフスクリーンエリアにテンプレートを配置
WR PICTURE TO AREA(Temp;[Letters]Doc_)
  ` WR PRINT MERGE (Temp;3)オフスクリーンエリアにテンプレートを配置
WR PRINT MERGE (Temp;3)
  ` テンプレートとテーブル3の範囲をマージ
WR DELETE OFFSCREEN AREA (Temp)
  ` オフスクリーンエリアを削除
```

参照

WR PRINT

WR ADD TAB

WR ADD TAB (エリア;位置;行揃え;埋め込み文字)

引数	タイプ	説明
エリア	倍長整数	4D Write エリア
位置	実数	タブ位置
行揃え	整数	行揃え値
埋め込み文字	文字	選択された埋め込み文字

説明

WR ADD TAB を使用すれば、ドキュメントの左余白から <位置> に指定された位置に新規のタブを追加することができます。またこのコマンドを使用すれば、埋め込み文字および新規のタブストップの行揃えを設定することもできます。

このタブストップは、選択されているすべての段落に追加されます。この位置にすでにタブストップがある場合、作成したタブストップに置き換えられます。

<位置> は、(ドキュメントのデフォルト単位で表されている) 左余白からの距離です。

オプション <行揃え> は、タブストップのタイプを指定します。値または以下の定義された定数を指定することができます。

値	定数	テキスト行揃え
1	wr left tab	左揃え
2	wr centered tab	中央揃え
3	wr right tab	右揃え
4	wr decimal tab	小数点区切り
5	wr vertical separator tab	縦区切り

<行揃え> が省略されている場合は、左揃えのタブが作成されます。

注：上記の定数一覧は、付録Dの「WR Tabs」を参照してください。

オプション<埋め込み文字>は、ASCIIコードの33から127までの間の文字を指定することができます。この文字は、タブストップと同じフォントを使用して追加されます。

<埋め込み文字>が省略されている場合や、空の文字列が指定されている場合は、埋め込み文字は挿入されません。

例題

以下の例は、左余白からドットを埋め込み文字として、現在の単位（ファイルメニュー「環境設定」の「単位」で設定）で50離れた位置に左タブストップを作成します。

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

または

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

参照

なし

WR DELETE TAB

WR DELETE TAB (エリア;タブ番号)

引数	タイプ	説明
エリア	倍長整数	4D Write エリア
タブ番号	倍長整数	タブの番号

説明

WR DELETE TAB は、<エリア> で指定された 4D Write エリアから、<タブ番号> で番号を指定（左から右へ数えて）されたタブを削除します。他のタブが同じ位置に配置された場合、それらのタブも削除されます。

注：選択範囲にいくつかの段落が含まれている場合、タブの番号付けは最後に選択された段落に適用されます。

例題

ドキュメントからすべてのタブストップを削除したい場合の例を示します。

```

C_LONGINT(Area;$i;$TabNum;$uniforme)
  `エリアの開始位置にカーソルを挿入する
WR SET SELECTION(Area;0;0)
  `ドキュメント内の段落の数をカウントする
NbParag:=WR Count(Area;wr nb paragraphs)
  `各段落の処理
For ($i;1;NbParag)
  `段落の位置を取得する
  WR GET PARAGRAPHS(Area;Start;Pos)
  `段落内に入る
  WR SET SELECTION(Area;Start+1;Debut+1)
  `タブストップの数を取得する
$TabNum:=WR Get text property(Area;wr tab:$uniforme)
While ($TabNum#0)
  WR DELETE TAB(Area;1)
  $TabNum:=$TabNum-1
End while
  `最後に処理した段落の直後に出る
  WR GET SELECTION(Area;Pos;Pos)
End for
    
```

参照

WR ADD TAB, WR DELETE STYLESHEET TAB

WR GET TAB

WR GET TAB (エリア;タブ番号;位置;行揃え;埋め込み文字)

引数	タイプ	説明
エリア	倍長整数	4D Write エリア
タブ番号	倍長整数	タブの番号
位置	実数	タブ位置
行揃え	整数	タブの行揃え値
埋め込み文字	文字	埋め込み文字

説明

WR GET TAB は、<エリア> で指定されたエリアの現在のルーラー内の <タブ番号> によって指定されたタブの位置と行揃え、埋め込み文字を返します。現在のルーラーは、カーソルポイントが表示されているルーラーもしくはいくつかの段落が選択された最後のルーラーです。

<タブ番号>

段落内のタブの数を知るには、タブストップの数を返す **WR Get text properties** (Area;45;1) を使用します。タブ番号を順にくり返し、処理することで、現在のルーラーの引数すべてを取得することができます。

<位置>

<位置> は、ドキュメントの左マージンからタブストップまでの距離です。これは、現在そのドキュメントのデフォルト単位となっているもので表されます。

<行揃え>

<行揃え> は、タブの行揃えのタイプです。

値	テキスト行揃え
1	左揃え
2	中央揃え
3	右揃え
4	小数点区切り
5	縦区切り

<埋め込み文字> は、ASCII コードの 33 から 127 までの間の文字を指定することができます。<埋め込み文字> が空の文字列の場合、タブ設定に用いられる埋め込み文字はありません。

例題

WR SET TAB コマンドと **WR DELETE TAB** コマンドの例題を参照してください。

参照

WR GET STYLESHEET TAB、WR SET TAB

WR SET TAB

WR SET TAB (エリア;タブ番号;位置;行揃え;埋め込み文字)

引数	タイプ	説明
エリア	倍長整数	4D Write エリア
タブ番号	倍長整数	タブの番号
位置	数値	新しいタブ位置
行揃え	整数	タブの行揃えの新しい値
埋め込み文字	文字	埋め込み文字として選択された新しい文字

説明

WR SET TABを使うことによって、<タブ番号>で指定されたタブストップに関する引数を設定することができます（タブは左から右へ数えます）。**WR SET TAB**は、<位置>で指定された位置にタブストップを移動し、さらに行揃え、埋め込み文字を設定します。

選択されたタブストップは、現在選択されている範囲内にあるすべての段落について修正されます。新しく指定された位置にすでにタブストップがある場合、定義したタブストップに置き換えられます。

<位置>は、左余白からの距離です。<位置>は、ドキュメントの現在のデフォルト単位で表されます。

<行揃え>は、タブストップの行揃えを指定します。値または定数を指定することができます。

値	定数	テキスト行揃え
1	wr left tab	左揃え
2	wr centered tab	中央揃え
3	wr right tab	右揃え
4	wr decimal tab	小数点区切り
5	wr vertical separator tab	縦区切り

注：テキストプロパティおよびそのリファンレスの一覧は、付録Dの「WR Tabs」を参照してください。

<埋め込み文字>は、ASCIIコードの33から127までの間の文字を指定することができます。この文字は、変更されたタブストップとしてフォントで表示されます。

例題

範囲において、168ポイントに配置されたタブストップを削除し、252ポイントのタブストップを280ポイントに移動し、“\$”文字を埋め込みます。

```

C_LONGINT(Area;$i;$Nbtab;$Unit;$unifome;$Justif)
C_REAL($Pos)
C_STRING(2;$fill)
$Nbtab:=WR Get text property(Area;wr tab;$unifome)
    `現在の単位をストア
$Unit:=WR Get doc property(Area;wr unit)
If ($Unit#2)
    `他の単位設定の場合は、単位をポイントにする
    WR SET DOC PROPERTY(Area;wr unit;2)
End if
$i:=1
Repeat
    WR GET TAB(Area;$i;$pos;$Justif;$fill)
    Case of
        :($Pos=168)
            `168ポイントに配置されたタブストップを削除する
            WR DELETE TAB(Area;$i)
            $Nbtab:=$Nbtab-1
        :($Pos=252)
            `252ポイントに配置されたタブストップを280mmまで移動する
            WR SET TAB(Area;$i;350;$Justif;"$")
            $i:=$i+1
    End case
Until ($i=$Nbtab)
    `元々の単位設定に戻す
WR SET DOC PROPERTY (Area;wr unit;$Unit)
    
```

参照

WR GET TAB、WR SET STYLESHEET TAB

WR ADD STYLESHEET TAB

WR ADD STYLESHEET TAB (エリア;スタイルシート番号;位置;{行揃え;{埋め込み文字}})

引数	タイプ	説明
エリア	倍長整数	4D Write エリア
スタイルシート番号	倍長整数	スタイルシートの番号
位置	実数	タブ位置
行揃え	整数	タブの行揃え値
埋め込み文字	文字	選択された埋め込み文字

説明

WR ADD STYLESHEET TAB を使用すれば、<スタイルシート番号> が参照するタブストップのリストに新規のタブストップを追加することができます。また **WR ADD STYLESHEET TAB** は、タブ位置やそのタイプ、埋め込み文字を設定することもできます。

<位置> によって指定された位置にすでにタブストップがある場合、定義したタブストップに置き換えられます。

注： **WR UPDATE STYLESHEET** を呼び出して、スタイルシートを使用しているテキストを更新しない限り、変更したいスタイルシートを使用しているテキストは更新されません。

<位置> は、(ドキュメントのデフォルト単位で表されている) 左余白からの距離です。

オプション<行揃え> は、タブストップのタイプを指定します。値または以下の定義された定数を指定することができます。

値	定数	テキスト行揃え
1	wr left tab	左揃え
2	wr centered tab	中央揃え
3	wr right tab	右揃え
4	wr decimal tab	小数点区切り
5	wr vertical separator tab	縦区切り

<行揃え>が省略されている場合は、左揃えのタブが作成されます。

注：上記の定数一覧は、付録Dの「WR Tabs」を参照してください。

オプション<埋め込み文字>は、ASCIIコードの33から127までの間の文字を指定することができます。この文字は、タブストップと同じフォントを使用して追加されます。

<埋め込み文字>が省略されている場合や、空の文字例が指定されている場合は、埋め込み文字は挿入されません。

例題

WR UPDATE STYLESHEET コマンドの例題を参照してください。

参照

WR ADD TAB、WR DELETE STYLESHEET TAB、WR GET STYLESHEET TAB、WR SET STYLESHEET TAB

WR APPLY STYLESHEET

WR APPLY STYLESHEET (エリア;スタイルシート番号)

引数	タイプ	説明
エリア	倍長整数	4D Write エリア
スタイルシート番号	倍長整数	スタイルシート番号

説明

WR APPLY STYLESHEET は、<エリア>の現在選択されている部分に<スタイルシート番号>で指定されたスタイルシートを適用します。スタイルシートの書式は選択部分に適用され、選択部分はスタイルシートを使って表示されます（カーソルがテキスト内にある場合、スタイルツールバーのスタイルシートドロップダウンリストにスタイルが表示されます）。

<スタイルシート番号>が任意のスタイルシートに対応していない場合、エラーが返されます。

例題

WR Create stylesheet関数の例題を参照してください。

参照

WR Create stylesheet、WR UPDATE STYLESHEET

WR Create stylesheet

WR Create stylesheet (エリア;名前{;適用{;ショートカット}) 結果

引数	タイプ	説明
エリア	倍長整数	4D Write エリア
名前	文字列	スタイルシートの名前
適用	倍長整数	0=キャラクタ (文字列) 1=段落
ショートカット	文字	1キャラクタのみ
戻り値	倍長整数	スタイルシート参照番号

説明

WR Create stylesheet は、新規スタイルシートを作成し、そのスタイルシートに割り当てられた番号を返します。この新規スタイルシートの内容は、<名前>、<適用>、<ショートカット> で設定されます。**WR SET STYLESHEET TEXT PROP**、**WR SET STYLESHEET FONT**、**WR SET STYLESHEET TAB** 及びスタイルシート参照番号を使って、スタイルシートを修正することができます。

<名前>

スタイルシートの名前は、最大31バイトです。

オプション<適用>

- ・ 0の場合、スタイルシートは1つのキャラクタが対象となります。
- ・ 1の場合、スタイルシートは1つの段落が対象となります。
- ・ 省略すると、スタイルシートは1つのキャラクタが対象となります。

オプション<ショートカット>

スタイルシートにキーボードショートカットを割り当てることができます。ただし1つのキャラクタしかショートカットに利用することはできません。ショートカットを利用するには、Windows上ではCtrlキー、Macintosh上ではcommandキーを使って、この引数に渡されるキーを押下する必要があります。4D Writeに標準で備わっているキーボードショートカットとコンフリクト（衝突）を起こさないように数値を用いることをお勧めします。

<ショートカット>を省略するか、または空の文字列を指定すると、スタイルシートにはショートカットキーは割り当てられません。

例題

文字のスタイルシートをカスタマイズして各ドキュメントに追加し、選択範囲に適用したいものとします。スタイルシートは、Macintosh上では「Command+I」キー、Windows上では「Ctrl+I」キーのショートカットキーを適用します。フォント属性は、MSゴシック、12ポイントにします。

```
$NumSheet:=WR Create stylesheet (Area;"MyOwnStyle";0;"1")  
WR SET STYLESHEET FONT (Area;$NumSheet;MSゴシック")  
WR SET STYLESHEET TEXT PROP (Area;$NumSheet;wr font size;12;1)  
WR EXECUTE COMMAND(Area;wr cmd select all)  
WR APPLY STYLESHEET(Area;$NumSheet)
```

参照

WR APPLY STYLESHEET、WR DELETE STYLESHEET、WR UPDATE STYLESHEET

WR DELETE STYLESHEET

WR DELETE STYLESHEET (エリア;スタイルシート番号)

引数	タイプ	説明
エリア	倍長整数	4D Write エリア
スタイルシート番号	倍長整数	スタイルシート番号

説明

WR DELETE STYLESHEET は、 < エリア > で参照される 4D Write エリアから < スタイルシート番号 > で渡された番号のスタイルシートを削除します。

警告：システムのスタイルシートは削除できません。**WR GET STYLESHEET INFO** を使うことにより、そのスタイルシートが削除から保護されているかどうかを確認することができます。

例題

ドキュメント内で保護されていないスタイルシートを削除したいものとします。

```
C_LONGINT(Area)
C_INTEGER(NbStyleSheet;$SheetNum)
  `スタイルシートの番号を調べる
NbStyleSheet:=WR Count(Area;wr_nb_stylesheets)
$SheetNum:=1
For ($i;1;NbStyleSheet)
  WR GET STYLESHEET
INFO(Area;$SheetNum;$Name;$ApplyTo;$Protected;$Shortcut)
If ($Protected=0) `スタイルシートが保護されていない場合...
  WR DELETE STYLESHEET (Area;$SheetNum)
Else
  $SheetNum:=$SheetNum+1
End if
End for
```

参照

WR CREATE STYLESHEET

WR DELETE STYLESHEET TAB

WR DELETE STYLESHEET TAB (エリア;スタイルシート番号;タブ番号)

引数	タイプ	説明
エリア	倍長整数	4D Write エリア
スタイルシート番号	倍長整数	スタイルシート番号
タブ番号	倍長整数	削除するタブの番号

説明

WR DELETE STYLESHEET TAB は、<エリア>で参照される 4D Write エリア内にある <スタイルシート番号> を持つスタイルシートから <タブ番号> で渡された番号のタブストップを削除します。スタイルシートは先頭から末端に向かって順に番号が振られ、「スタイルシート」ダイアログに一覧表示されます。このコマンドは、<スタイルシート番号> のスタイルシートが現在選択されたテキストで使用されている場合でも、何も影響を与えません。

修正されたスタイルシートを使用するテキストを更新するには、**WR UPDATE STYLESHEET** を使用する必要があります。

例題

WR UPDATE STYLESHEET コマンドの例題を参照してください。

参照

WR ADD STYLESHEET TAB、WR DELETE TAB

WR Get stylesheet font

WR Get stylesheet font (エリア;スタイルシート番号)

引数	タイプ	説明
エリア	倍長整数	4D Write エリア
スタイルシート番号	倍長整数	スタイルシート番号
戻り値	文字列	フォント名

説明

WR Get stylesheet font は、<エリア> で参照される 4D Write エリア内にある <スタイルシート番号> を持つスタイルシートに割り当てられたフォントの名前を返します。スタイルシートは先頭から末端に番号が振られ、「スタイルシート」ダイアログに一覧表示されます。このスタイルシート用に定義されたフォントがない場合は、空の文字列が返されます。

例題

システムに入っていない特定のフォントが使われているスタイルシートからフォント属性をなくしたいものとします。

```
ARRAY STRING(80;FontsArray)
WR FONTS TO ARRAY(FontsArray)
$StyleSheetNum:=WR Count(Area;wr nb stylesheets)
For ($i;1;$StyleSheetNum)
  $Fonts:=WR Get stylesheet font(Area;$i)
  If (($Fonts#"" ) & (Find in array(Area;$Fonts)=0))
    WR SET STYLESHEET FONT(Area;$i;"" )
  End if
End for
```

参照

WR Get font、WR SET STYLESHEET FONT

WR GET STYLESHEET INFO

WR GET STYLESHEET INFO (エリア;スタイルシート番号;名前;適用;プロテクト;
ショートカット)

引数	タイプ	説明
エリア	倍長整数	4D Write エリア
スタイルシート番号	倍長整数	スタイルシート番号
名前	文字	スタイルシートの名前
適用	倍長整数	0=キャラクタ (文字列) 1=段落
プロテクト	倍長整数	0=プロテクトなし 1=プロテクトあり
ショートカット	文字列	1キャラクタのみ

説明

WR GET STYLESHEET INFO は、<エリア>で参照される4D Writeエリア内にある<スタイルシート番号>を持つスタイルシートに関する情報を取り出します。

<名前> は、スタイルシートのタイトルを返します。

<適用>

- ・ 0の場合、スタイルシートは1つの文字列が対象となります。
- ・ 1の場合、スタイルシートは1つの段落が対象となります。

<プロテクト>

- ・ 0の場合、スタイルシートは保護されません。つまり、これはシステムに用意されたスタイルシートではありません。
- ・ 1の場合、スタイルシートは保護されます。これはシステムが用意するスタイルシートなので削除できません。

<ショートカット>

スタイルシートに割り当てられたショートカットを返します。ただし1つのキャラクタしかショートカットに利用することはできません。ショートカットを利用するには、Windows上ではCtrlキー、Macintosh上ではcommandキーを使って、この引数に渡されるキーを押下する必要があります。

<ショートカット> が空の文字列の場合、<スタイルシート番号>に割り当てられるショートカットはありません。

例題

WR SET STYLESHEET INFO コマンド、**WR DELETE STYLESHEET** コマンド、**WR UPDATE STYLESHEET** コマンドの例を参照してください。

参照

WR SET STYLESHEET INFO

WR GET STYLESHEET TAB

WR GET STYLESHEET TAB(エリア;スタイルシート番号;タブ番号;位置;行揃え;埋め込み文字)

引数	タイプ	説明
エリア	倍長整数	4D Write エリア
スタイルシート番号	倍長整数	スタイルシート番号
タブ番号	倍長整数	タブ番号
位置	実数	タブの位置
行揃え	整数	タブの行揃え
埋め込み文字	文字	選択された埋め込み文字

説明

WR GET STYLESHEET INFO は、<タブ番号> で渡された番号のタブ設定を取り出します。このタブ設定は<エリア>で参照される4D Writeエリア内にある<スタイルシート番号>を持つスタイルシートに属します。

<タブ番号>

スタイルシート内のタブ番号を知るには、タブストップの番号を返す **WR Get stylesheet text prop** を用います。

<位置>

ドキュメントの左マージンからタブストップまでの距離です。これは、現在そのドキュメントのデフォルト単位となっているもので表されます。

<行揃え>

タブの行揃えタイプです。

値	行揃え
1	左揃え
2	中央揃え
3	右揃え
4	小数点区切り
5	縦区切り

<埋め込み文字>

ASCIIコード33から127までの文字を使用することができます。<埋め込み文字>が空の文字列の場合、タブ設定に用いられる埋め込み文字はありません。

例題

各スタイルシートのタブストップに対する埋め込み文字を変更し、ドキュメントに反映させたいものとします。

```
$StyleSheetNum:=WR Count(Area;wr nb feuilles de style)
For ($i;1;$StyleSheetNum)
  $TabNum:=WR Get stylesheet text prop(Area;$i;wr tab:$Apply)
  If ($TabNum#0)
    For ($j;1;$TabNum)
      WR GET STYLESHEET TAB(Area;$i;$j;$Pos;$Justif;$FillChar)
      If
        WR SET STYLESHEET TAB(Area;$i;$j;$Pos;$Justif;Char(126))
      End if
    End for
  WR UPDATE STYLESHEET(Area;$i)
End if
End for
```

参照

WR ADD STYLESHEET TAB、WR GET TAB、WR SET STYLESHEET TAB

WR Get stylesheet text prop

WR Get stylesheet text prop (エリア;スタイルシート番号;プロパティ;適用)

引数	タイプ	説明
エリア	倍長整数	4D Write エリア
スタイルシート番号	倍長整数	スタイルシート番号
プロパティ	整数	読み込まれるテキストプロパティの番号
適用	整数	0=プロパティは適用されない 1=プロパティは適用される
戻り値	実数	適用の有無

説明

WR Get stylesheet text prop は、 <プロパティ> で渡されたプロパティが <エリア> の 4D Write エリア内の選択した範囲に適用されているかどうかを調べることができます。

<プロパティ>

- ・ <プロパティ> =7 (wr font number 定数)の場合、返される値はシステム内部で持っている番号です。4D Write は使用されるフォントにフォント番号を順番に割り振ります。このフォント番号は、**WR SET STYLESHEET TEXT PROP** でのみ使用されます。フォント名に関する基本的な操作については、**WR Get stylesheet font** や **WR SET STYLESHEET FONT** を使用することをお勧めします。
- ・ <プロパティ> =15 (wr stylesheet number 定数)は、この関数では機能しません。
- ・ <プロパティ> =64 (wr tab 定数)の場合、**WR Get stylesheet text prop** はスタイルシートのタブ設定を返します。

カラーに関するプロパティの戻り値は、(4D や 4D Write の旧バージョンと同じ) 次のようなフォーマットで表されます : 0x00RRVVB。RGB の値を分割するには、**WR COLOR TO RGB** を使用します。

-1 がプロパティ 11(wr strikethrough color 定数)、12(wr underline color 定数)または13(wr shadow color 定数)に対して返されると、これらの要素はテキスト内では同じカラーになります。

-1 がプロパティ 10(wr text back color 定数)に対して返されると、テキストに対して選択された背景色はありません。

注 : <プロパティ> は定数を使ってセットされます。

テキストプロパティの定数一覧は付録Dの「WR Text properties」を参照してください。

<適用> が1の場合、スタイルシートはプロパティを適用し、スタイルシートの定義内容でそれを修正します。

例題

WR UPDATE STYLESHEET コマンド、**WR GET STYLESHEET TAB** コマンドの例題を参照してください。

参照

WR SET STYLESHEET TEXT PROP

WR SET STYLESHEET FONT

WR SET STYLESHEET FONT (エリア;スタイルシート番号;フォント)

引数	タイプ	説明
エリア	倍長整数	4D Write エリア
ピクチャ番号	倍長整数	ピクチャ番号
フォント	文字列	フォントの名前

説明

WR SET STYLESHEET FONT は、`<エリア>` で参照される 4D Write ドキュメント内の `<スタイルシート番号>` で渡されたスタイルシートの文字フォントを修正することができます。

`<フォント>`

`<フォント>` には、使用したいフォントの名前を渡します。選択した範囲にスタイルシートを適用したい場合は、`<フォント>` に空の文字列を渡します。

例題

WR SET STYLESHEET INFO コマンドの例題を参照してください。

参照

WR Get stylesheet font、WR SET FONT

WR SET STYLESHEET INFO

WR GET STYLESHEET INFO (エリア;スタイルシート番号;名前;適用;ショートカット)

引数	タイプ	説明
エリア	倍長整数	4D Write エリア
スタイルシート番号	倍長整数	スタイルシート番号
名前	文字列	スタイルシートの名前
適用	整数	0=キャラクタ (文字列) 1=段落
ショートカット	文字	1つのキャラクタのみ ""ショートカットがない場合

説明

WR SET STYLESHEET INFO は、<スタイルシート番号>で渡される参照番号のスタイルシートのプロパティを修正することができます。このスタイルシートは、<エリア>の参照番号を持つ4D Writeドキュメント内に含まれています。スタイルシート番号は、スタイルシートが「スタイルシート」ドロップダウンリストまたは「スタイルシート」ダイアログ内のリストに表示される際の順序に対応しています。

<スタイルシート番号>

スタイルシート番号をそのまま残しておきたい場合は、**WR GET STYLESHEET INFO** を呼び出し、このコマンドによって返される参照番号を使用する必要があります。

<名前>

<名前> が空の文字列の場合、オリジナルのスタイルシート名は変更されません。

警告：2つのスタイルシートがお互い同じ名前を持つことはできますが、これらは常に異なる参照番号を持っています。

<適用>

- ・ -1の場合、現在の値がそのまま保持されます。
- ・ 0の場合、スタイルシートは文字列が対象となります。
- ・ 1の場合、スタイルシートは段落が対象となります。

段落のスタイルシートは、先頭または最終の段落だけを選択した場合でも常に選択範囲内のすべての段落に適用されます。デフォルトでは、新規に作成されるスタイルシートは文字用のスタイルシートとなります。

<ショートカット>

オプション<ショートカット>は、スタイルシートにキーボードショートカットを割り当てることができます。ただし1つのキャラクタしかショートカットに利用することはできません。ショートカットを利用するには、Windows上ではCtrlキー、Macintosh上ではcommandキーを使って、この引数に渡されるキーを押下する必要があります。4D Writeに標準で備わっているキーボードショートカットとコンフリクトを起こさないように数値を用いることをお勧めします。

<ショートカット>が省略または空の文字列の場合、スタイルシートに割り当てられるショートカットはありません。

例題

"Title"スタイルシートの定義を次のように修正したいものとします：名前は"Title 14"に変更し、フォントはMS明朝、14ポイント、太字、青に設定します。

```
NbStyles:=WR Count (Area;12)
For ($i;1;NbStyles)
  WR GET STYLESHEET INFO(Area;$i;$Name;$ApplyTo;$Protected;$Shortcut)
  If ($Name="Title")
    WR SET STYLESHEET INFO(Area;$i;"Title 14";$ApplyTo;$Shortcut)
    WR SET STYLESHEET FONT(Area;$i;"MS 明朝")
    WR SET STYLESHEET TEXT PROP(Area;$i;wr font size;14;1)
    WR SET STYLESHEET TEXT PROP(Area;$i;wr bold;1;1)
    WR SET STYLESHEET TEXT PROP(Area;$i;wr text color;212;1)
  End if
End for
```

参照

WR GET STYLESHEET INFO

WR SET STYLESHEET TAB

WR SET STYLESHEET TAB (エリア;スタイルシート番号;タブ番号;位置;行揃え;埋め込み文字)

引数	タイプ	説明
エリア	倍長整数	4D Write エリア
スタイルシート番号	倍長整数	スタイルシート番号
タブ番号	倍長整数	タブ番号
位置	数値	タブの位置
行揃え	整数	タブの行揃えの新しい値
埋め込み文字	文字列	選択された埋め込み文字

説明

WR SET STYLESHEET INFOは、<スタイルシート番号>で渡された番号のスタイルシートに属している<タブ番号>で渡された番号を持つタブストップのパラメータを修正することができます(タブは左右にカウントされ、スタイルシートは「スタイルシート」ダイアログに示されるように上下にカウントされます)。**WR SET STYLESHEET TAB**は、<位置>にタブを移動し、タブの行揃えだけでなく埋め込み文字の設定も行います。

このコマンドは、スタイルシートを修正中であっても、選択されたテキストには何の影響も与えません。

スタイルシートを用いるテキストを更新したい場合は、スタイルシート定義内容を修正した後に**WR UPDATE STYLESHEET**を呼び出します。

このコマンドを使って即座にスタイルシートと現在選択されているエリアの両方に新しく設定したスタイルシートのタブ属性を適用したい場合は、**WR APPLY STYLESHEET**を用います。

あるタブストップがすでにスタイルシート内の新しい場所に存在している場合、それはこのコマンドで設定されるタブストップによって置き換えられます。

<位置>は、ドキュメントの左マージンからタブストップまでの距離です。これは、現在そのドキュメントのデフォルト単位となっているもので表されます。

<行揃え>は、タブストップで選択したい行揃えのタイプを指定します。この場合、下記の値または定数のどちらかを使用することができます。

値	定数	行揃え
1	wr left tab	左揃え
2	wr centered tab	中央揃え
3	wr right tab	右揃え
4	wr decimal tab	小数点区切り
5	wr vertical separator tab	縦区切り

<埋め込み文字>

<埋め込み文字>には、ASCIIコード33から127までの文字を使用することができます。この文字は修正されたタブストップと同じフォントで表示されます。

例題

WR GET STYLESHEET TAB コマンドの例題を参照してください。

参照

WR ADD STYLESHEET TAB、WR DELETE STYLESHEET TAB

WR SET STYLESHEET TEXT PROP

WR SET STYLESHEET TEXT PROP (エリア;スタイルシート番号;プロパティ;値;適用)

引数	タイプ	説明
エリア	倍長整数	4D Write エリア
スタイルシート番号	倍長整数	スタイルシート番号
プロパティ	整数	読み込まれるプロパティの番号
値	実数	選択されたプロパティの値
適用	整数	0=プロパティに値を適用しない 1=プロパティに値を適用する

説明

WR SET STYLESHEET TEXT PROP は、<スタイルシート番号> で渡されたスタイルシートのテキストプロパティを修正することができます。

このスタイルシートを現在使用しているすべてのテキストを更新したい場合は、スタイルシート定義内容を修正した後に **WR UPDATE STYLESHEET** を呼び出します。

このコマンドを使って即座にスタイルシートと現在選択されているエリアの両方に新しく設定したスタイルシートのテキストプロパティを適用したい場合は、**WR APPLY STYLESHEET** を用います。

<値> に与えられる意味は、使用されるプロパティ値に依存します。

<プロパティ> の値が定数プロパティである wr bold または 0 の場合、<値> の値は 1(True) または 0(False) のいずれかになります。

<プロパティ> の値が定数プロパティである wr font size または 8 の場合、<値> の値は 9,10,12... になります。

注：<プロパティ> と <値> は定数を使って設定されます。

テキストプロパティの定数一覧は付録Dの「WR Text properties」を参照してください。

テキストプロパティの値一覧は付録Dの「WR Text properties values」を参照してください。こちらは、値または定数のどちらかを渡すことができます。

プロパティへの変更を行いたい場合は、<適用> に 1 を渡します。これを行うと、<値> はプロパティの新しい値を定義します。

プロパティへの変更を行いたくない場合は、<適用> に 0 を渡します。これを行うと、<値> は何も影響ありません。

例題

WR SET STYLESHEET INFO コマンドの例題を参照してください。

参照

WR Get stylesheet text prop

WR UPDATE STYLESHEET

WR UPDATE STYLESHEET (エリア;スタイルシート番号)

引数	タイプ	説明
エリア	倍長整数	4D Write エリア
スタイルシート番号	倍長整数	スタイルシート番号

説明

WR UPDATE STYLESHEETは、<エリア>で参照される4D Writeエリア内の<スタイルシート番号>で参照されたスタイルシートを使って、すべてのテキストの表示フォーマットを更新します。このコマンドを実行すると、参照されたスタイルシート上で書式設定されているすべてのテキストはこのスタイルシートで現在設定されている内容に従って新しく書式化されます。

例題

"LayoutPar"スタイルシート内のタブストップを置き換え、スタイルシートが適用されているテキストエリアを更新したいものとします。

```
  `スタイルシート番号を見つける
$StyleSheetNb:=WR Count(Area;wr_nb_stylesheets)
For ($i;1;$StyleSheetNb)
  WR GET STYLESHEET INFO(Area;$i;$Name;$ApplyTo;$Prot ;$Shortcut)
  If ($Name="LayoutPar")
    SheetNumber:=$i
  End if
End for
  `スタイルシート内のタブストップ番号を取得
$NbTab:=WR Get stylesheet text prop(Area;SheetNumber;wr_tab:Apply)
  `タブストップを削除
For ($i;1;$NbTab)
  WR DELETE STYLESHEET TAB(Area;SheetNumber;1)
End for
  `新しいタブを挿入
WR ADD STYLESHEET TAB(Area;SheetNumber;10;wr_left_tab:Char(126))
...
  `スタイルシートが適用されている各段落を更新
WR UPDATE STYLESHEET(Area;SheetNumber)
```

参照

WR APPLY STYLESHEET、WR CREATE STYLESHEET

WR BACKSPACE

WR BACKSPACE (エリア)

引数	タイプ	説明
エリア	倍長整数	4D Write エリア

説明

WR BACKSPACE は、delete キーまたはbackspace キーの押下をシミュレートします。 <エリア> 内で文字が選択されていると、その文字は削除されます。文字が選択されていない場合は、**WR BACKSPACE** はdelete キーまたはbackspace キーの押下と同じ動作を行います。1回につき1文字が削除され、挿入ポインタ（カーソル）が1文字分、左に移動します。

参照

WR DELETE PICTURE IN PAGE、WR DELETE SELECTION

WR DELETE SELECTION

WR DELETE SELECTION (エリア)

引数	タイプ	説明
エリア	倍長整数	4D Write エリア

説明

WR DELETE SELECTION は、<エリア> で参照される 4D Write エリアから現在選択されているテキストエリアを削除することができます。

次のステートメントを実行すると、**WR DELETE SELECTION** を使った場合とまったく同じ結果になります：

```
WR EXECUTE (Area;wr_cmd clear)
```

注：定数 wr cmd clear の値は 6 です。

現在何も選択されていない場合、カーソルの前にある文字を削除する **WR BACKSPACE** とは異なり、このコマンドは何も行いません。

例題

ドキュメント内のすべてのソフトハイフンを削除したいものとします。

```
  `数を調べる  
HyphenNb:=WR Count(Area;wr_nb_soft hyphens)  
For($i;1;HyphenNb)  
  `毎回最初のソフトハイフンを選択  
  WR SELECT(Area;9;1)  
  `削除する  
  WR DELETE SELECTION(Area)  
End for
```

参照

WR BACKSPACE、WR DELETE PICTURE IN PAGE

WR Direct find

WR Direct find (blob;検索文字列;検索方法;検索範囲) 結果

引数	タイプ	説明
blob	Blob	Blob
検索文字列	文字	検索される文字列
検索方法	整数	0=部分一致 1=完全一致
検索範囲	整数	0=カーソル以降を検索 1=ドキュメント全体を検索
戻り値	倍長整数	検索ステータス

説明

WR Direct find は、任意の4D Writeエリアに含まれるBLOB内の文字列を直接検索することができます。このコマンドの使用にBLOBを4D Writeエリア内であらかじめオープンしておく必要はありません。つまり、このコマンドはとても高速に実行されることを意味しています。

検索文字列が見つかったと、**WR Direct find** はテキスト内の検索文字列が見つかった位置を返します。

検索が失敗した場合は、**WR Direct find** は-1を返します。

< blob > が4D Writeエリアのコンテンツでない場合は、**WR Direct find** は-2を返します。

< 検索方法 > と < 検索範囲 > によって、検索方法を選択できます：

< 検索方法 > が1の場合、完全に一致した単語のみを検索します。このオプションを使った場合、検索される文字列は（空白やカンマなどの）句読文字で囲われている必要があります。< 検索方法 > が1でない場合、検索文字列は全単語または部分単語のどちらでも構いません。

< 検索範囲 > は、検索がドキュメント全体に適用されるかどうかを定義することができます。

< 検索範囲 > が1の場合、検索はドキュメント全体に対して行われます。これに対して、< 検索範囲 > が0の場合は、現在のカーソル位置からドキュメントの終わりまでの範囲で検索が行われます。

例題

この例題は、レコードの選択範囲内で検索するキーワードからの検索メソッドを示しています。データベースの内容は料理レシピです。4D Write エリアはBLOBフィールドに保存されています。特別な素材を使ったレシピをすべて検索したいものとします。ここに、大変高速なメソッドを示します。

```
ToFind:=Request("素材を入力してください：")
`検索されたレコードを配置できるだけのスペースを作成
CREATE EMPTY SET([MyRecipes];"FoundRecords")
ALL RECORDS([MyRecipes]) `全テーブルの選択範囲をブラウズ
While (Not(End selection([MyRecipes])))
    If (WR Direct find ([MyRecipes]BlobRecipe_;ToFind;1;1)>0)
        `素材が見つかったら、そのレコードを追加
        ADD TO SET([MyRecipes];"FoundRecords")
    End if
    NEXT RECORD([MyRecipes])
End while
USE SET("FoundRecords")
OUTPUT FORM([MyRecipes];"Output")
MODIFY SELECTION([MyRecipes];*)
```

参照

WR Find

WR Find

WR Find (エリア;検索文字列;検索方法;大文字検索;検索範囲) 結果

引数	タイプ	説明
エリア	倍長整数	4D Write エリア
検索文字列	文字列	検索される文字列
検索方法	整数	0=部分一致 1=完全一致
大文字検索	整数	0=大文字/小文字の区別なし 1=大文字/小文字の区別あり
検索範囲	整数	0=カーソル以降を検索 1=ドキュメント全体を検索
戻り値	倍長整数	検索ステータス

説明

WR Findは、任意の4D Write エリア内の文字を検索することができます。**WR GET WORD**を使うと、検索した単語の位置を取り出すことができます。また、**WR GET SELECTION**を使うと、検索範囲の位置を取り出すことができます。検索文字列が見つかったら、**WR Find**は1を返し、最初に見つかった文字列を選択します。

検索が失敗すると、**WR Find**は0を返し、現在選択されている部分は修正されません。
<エリア>が存在しない場合は、**WR Find**は-1を返します。

<検索方法>と<検索範囲>によって検索方法を選択できます：

<検索方法>が1の場合、完全に一致した単語のみを検索します。このオプションを使った場合、検索される文字列は(空白やカンマなどの)句読文字で囲まれている必要があります。<検索方法>が1でない場合、検索文字列は全単語または部分単語のどちらでも構いません。

<大文字検索>が1の場合、大文字と小文字を区別せずに元の文字列と一致する文字列を検索します。

<検索範囲>は、検索がドキュメント全体に適用されるかどうかを定義することができます。

<検索範囲>が1の場合、検索はドキュメント全体に対して行われます。これに対して、<検索範囲>が0の場合は現在のカーソル位置からドキュメントの終わりまでの範囲で検索が行われます。

例題

1. 検索文字列をユーザが入力するように要求し、検索を実行します。

```
ToFind:=Request("探したい単語を入力してください:")
If(OK=1)
  WR SET SELECTION(Area;0;0)
  If(WR Find(Area;ToFind;1;1;1)=0)
    ALERT("何も見つかりませんでした")
  End if
End if
```

2. この例題は、レコードの選択範囲内で検索するキーワードからの検索メソッドを示しています。検索は、ピクチャエリアで実行されます。

重要：4D Write エリアが BLOB フィールドとして保存された場合は、**WR Find direct** 関数の例題を参照してください。その例題は、より高速なものです。

データベースの内容は料理レシピです。4D Write エリアはピクチャフィールドに保存されています。特別な素材が使われているレシピをすべて検索したい場合のメソッドを示します。

```
ToFind:=Request("素材を入力してください:")
`検索されたレコードを配置できるだけのスペースを作成
CREATE EMPTY SET([MyRecipes];"FoundRecords")
ALL RECORDS([MyRecipes]) `全テーブルの選択範囲をブラウズ
OffscreenArea:=WR New offscreen area
While (Not(End selection([MyRecipes])))
  WR PICTURE TO AREA (OffscreenArea:[MyRecipes]PictRecipe_)
  If (WR Find (OffscreenArea;ToFind;1;1;1)=1)
    `素材が見つかったら、そのレコードを追加
    ADD TO SET([MyRecipes];"FoundRecords")
  End if
  NEXT RECORD([MyRecipes])
End while
WR DELETE OFFSCREEN AREA (OffscreenArea)
USE SET("FoundRecords")
OUTPUT FORM([MyRecipes];"Output")
MODIFY SELECTION([MyRecipes];*)
```

参照

WR Direct find

WR Get font

WR Get font (エリア;同フォント)

引数	タイプ	説明
エリア	倍長整数	4D Write エリア
同フォント	倍長整数	1=選択範囲内に同じフォントを適用 0=選択範囲内に異なるフォントを適用
戻り値	文字列	フォント名

説明

WR Get font は、<エリア>で参照される4D Writeエリア内で選択されている箇所の最終文字に適用されているフォントの名前を返します。

<同フォント>

- ・ 0の場合、選択されている範囲に異なるフォントが適用されます。
- ・ 1の場合、選択されている全範囲に同じフォントが適用されます。

例題

現在の選択範囲のフォントをドキュメント全体に適用したいものとします。

```
vFont:=WR Get font(Area;vUniforme)
If (vUniforme=0) `現在の選択範囲に複数のフォントがある場合
    CONFIRM("選択範囲内に複数のフォントが存在します。最後の文字に使われた
        フォントは"+vFont+"です。このフォントをドキュメント全体に適用しますか?")
    Else
        CONFIRM("選択範囲のフォントは"+vFont+"です。このフォントをドキュメント
            全体に適用しますか?")
    End if
If (OK=1)
    WR EXECUTE COMMAND(Area;wr cmd selec all)
        `ドキュメント全体を選択
    WR SET FONT(Area;vFont) `新しいフォントを適用
    WR SET SELECTION(Area;0;0)
        `ドキュメントの開始位置に挿入ポイントを移動
    WR SCROLL TO SELECTION(Area)
        `現在のテキストの選択範囲を表示
End if
```

参照

WR Get stylesheet font、WR Get text property、WR SET FONT

WR GET PARAGRAPHS

WR GET PARAGRAPHS (エリア;開始段落;終了段落)

引数	タイプ	説明
エリア	倍長整数	4D Write エリア
開始段落	倍長整数	返される段落の開始位置
終了段落	倍長整数	返される段落の終了位置

説明

WR GET PARAGRAPHS は、<エリア>で参照される 4D Write エリア内で選択されている箇所の開始段落の開始文字位置および終了段落のキャリッジリターンの位置を返します。

例題

次の例はドキュメントをスキャンし、各段落の開始文字位置と終了文字位置を取り出します。

```
` エリアの先頭にカーソルを置く
WR SET SELECTION (Area;0;0)
` ドキュメント内の段落数を数える
nbPara:=WR Count (Zone;wr nb paragraphs)
` 1つずつ段落を処理する
For ($i;1;nbPara)
    ` 開始文字位置と終了文字位置を取り出す
    WR GET PARAGRAPHS (Area;begin;Pos)
    ` 最後に処理された段落の後にエリアを再配置する
    WR SET SELECTION (Area;Pos;Pos)
End for
```

参照

なし

WR Get selected text

WR Get selected text (エリア) 結果

引数	タイプ	説明
エリア	倍長整数	4D Write エリア
戻り値	テキスト	エリア内の選択テキスト

説明

WR Get selected text は、<エリア>内の選択テキストを返します。

4th Dimensionのフィールドまたは変数が格納できる最大文字数は、32000バイトです。そのため、**WR Get selected text**も最大32000バイトの文字を返します。32000バイトより大きい文字が選択されると、この関数は空の文字列を返し、エラーが**WR Error number**によってレポートされます。

例題

次の例は、vText変数の中に<エリア>内の選択テキストを配置します。

```
vYexy := WR Ger selected text (Area)
```

参照

WR GET SELECTION

WR GET SELECTION

WR GET SELECTION (エリア;先頭;最終)

引数	タイプ	説明
エリア	倍長整数	4D Write エリア
先頭	倍長整数	先頭文字位置を受け取る
最終	倍長整数	最終文字位置を受け取る

説明

WR GET SELECTION は、`<エリア>` で参照される 4D Write エリア内の選択テキストの先頭文字位置と最終文字位置を `<先頭>` と `<最終>` の中に返します。

`<先頭>` は、常に選択された先頭文字位置よりも1つ前です。`<最終>` は、常に選択された最終文字位置と同じです。`<先頭>` と `<最終>` が等しい場合、選択されているテキストはありません。また、`<先頭>` の位置にある文字の後ろにカーソルポイントが置かれます。

例題

次の例は、ドキュメント全体のマージンを設定し、元の選択範囲を取り出します。

```
WR GET SELECTION (Area;DebSel;FinSel)
  `現在選択されている部分を読み込む
WR EXECUTE COMMAND (Area;wr cmd select all)      `すべてを選択
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)      `元の選択範囲を再設定する
```

参照

なし

WR Get styled text

WR Get styled text (エリア) 結果

引数	タイプ	説明
エリア	倍長整数	4D Write エリア
戻り値	BLOB	フォーマットされたテキスト

説明

WR Get styled text は、<エリア> で指定された 4D Write エリア内で選択されたテキストを、BLOB フィールドまたは BLOB 変数に返します。返された BLOB の構造は、文字および段落のフォーマットを含んでスタイルシートを使用せずに選択されたテキストを表示します。

WR Get styled text を用いて返されたテキストは、**WR INSERT STYLED TEXT** を使用して他の 4D Write ドキュメントに配置することができます。スタイルを付加されたテキストが挿入されても、4D Write ドキュメントのページレイアウトは挿入の影響を受けません。

WR Get styled text と **WR INSERT STYLED TEXT** を使用することにより、クリップボードの代わりに BLOB をバッファとして使用し、コピー&ペースト操作をシミュレートすることができます。

警告：**WR Get styled text** によって返された BLOB は、4D Write エリアのすべての要素を含んでいるものではないので、**WR BLOB TO AREA** では使用できません。

例題

WR INSERT STYLED TEXT コマンドの例題を参照してください。

参照

WR INSERT STYLED TEXT

WR Get text

WR Get text (エリア;先頭;最終) 結果

引数	タイプ	説明
エリア	倍長整数	4D Write エリア
先頭	倍長整数	テキストの先頭文字位置
最終	倍長整数	テキストの最終文字位置
戻り値	テキスト	先頭文字と最終文字の間のテキスト

説明

WR Get text は、<エリア>内の<先頭>の位置にある文字と<最終>の位置にある文字の間にあるテキストを返します。

4th Dimensionのフィールドまたは変数が格納できる最大文字数は、32000バイトです。そのため、**WR Get text**も最大32000バイトの文字を返します。<先頭>と<最終>の差異が32000バイトより大きく、そのドキュメントが少なくとも<最終>の文字数を持っていると、この関数は空の文字列を返し、エラーが**WR Error number**によってレポートされます。

<最終>が対象ドキュメントの文字数より大きく、<先頭>と<最終>の間のバイト数が32000バイト以下の場合、**WR Get text**はドキュメントの最後まで文字列を返します。

WR Get textは、<エリア>内の選択テキストに何の影響も与えません。

例題

次の例は、vText変数の中にエリア内の先頭100文字を格納します。

```
vText := WR Ger selected text (Area; 0; 100)
```

参照

なし

WR Get text property

WR Get text property (エリア;プロパティ;同プロパティ)

引数	タイプ	説明
エリア	倍長整数	4D Writeエリア
プロパティ	整数	プロパティ番号
同プロパティ	整数	0=選択範囲の一部または全体にプロパティがない場合 1=選択範囲全体にプロパティがある場合
戻り値	実数	使用の有無

説明

WR Get text property は、`<プロパティ>` で渡されたプロパティが `<エリア>` の 4D Write エリア内で現在選択されている範囲内で使用されているかどうかを調べることができます。

- ・ `<同プロパティ>` が 0 の場合、そのプロパティは選択されている範囲内の一部のみに適用されます。
- ・ `<同プロパティ>` が 1 の場合、そのプロパティは選択されている範囲全体に適用されます。

返される値は、選択範囲の最終文字のステータスに対応します。

`<プロパティ>`

- ・ `<プロパティ> = 7` (`wr font number` 定数) の場合、返される値はシステム内部の番号です。4D Write は使用されるフォントにフォント番号を順番に割り振ります。このフォント番号は、**WR SET TEXT PROPERTY** でのみ使用されます。フォント名に関する基本的な操作については、**WR Get stylesheet font** や **WR SET STYLESHEET FONT** を使用することをお勧めします。
- ・ `<プロパティ> = 15` (`wr stylesheet number` 定数) の場合、**WR Get text property** は選択範囲の先頭に適用されたスタイルシート番号を返します。スタイルシートは、「スタイルシート」ダイアログに示されているように上から下に番号が振られます。
- ・ `<プロパティ> = 45` (`wr tab` 定数) の場合、**WR Get stylesheet text prop** は選択範囲の最終段落にあるタブストップの番号を返します。

カラーに関するプロパティのリターン値は、(4D や 4D Write の旧バージョンと同じ) 次のようなフォーマットで表されます：

```
0x00RRVVBB
```

RGB の値を分割するには、**WR COLOR TO RGB** を使用します。

- ・ -1がプロパティ 11(wr strikethrough color 定数)、12(wr underline color 定数)または13(wr shadow color 定数)に対して返されると、これらの要素はテキスト内では同じ色になります。
- ・ -1がプロパティ 10(wr text back color 定数)に対して返されると、テキストに対して選択された背景色はありません。

無効な値を渡した場合、エラー 1075 が返ります。エラーコードに関しては、付録Cの「エラーコード」を参照してください。

注：<プロパティ>は定数を使って設定されます。定数のリストに関しては、付録Dの「4D Write 定数」を参照してください。

テキストプロパティの定数一覧は、付録Dの「WR Text properties」を参照してください。

例題

1. マージンサイズの合計値が一定の値を超えないようにしたいものとします。

```
Left:=WR Get text property(Area;wr left margin;$Uniforme)
If(Left<3)    `左マージンは3に設定
    WR SET TEXT PROPERTY(Area;wr left margin;3)
End if
Right:=WR Get text property(Area;wr right margin;$Uniforme)
If(Right>43) `右マージンは43に設定
    WR SET TEXT PROPERTY(Area;wr right margin;43)
End if
```

2. メニューとルーラーにはアクセスせずに、行間と行揃えを設定したいものとします。入力フォームに含まれている「情報」とラベル付けされたボタンと、2つの変数 ("LineSpacing"及び"Alignment") をメソッドに埋め込みます。

- ・ 「情報」ボタンのオブジェクトメソッドで、現在のカーソル位置の情報を取得します。

```
LineSpacing:=WR Get text property(Area;wr line spacing;$Uniforme)
If($Uniforme=0)
    ALERT("選択範囲内には複数の行間タイプが存在します。")
    $Assign:=True
Else
    $Assign:=False
End if
Alignment:=WR Get text property(Area;wr justification;$Uniforme)
If($Uniforme=0)
    ALERT("選択範囲内には複数の行揃えのタイプが存在します。")
End if
```


- ・ "LineSpacing"オブジェクトメソッドは、ユーザが行間を選択した場合の設定です。
WR SET TEXT PROPERTY(Area;LineSpacing)
- ・ "Alignment"オブジェクトメソッドは、ユーザが行揃えを選択した場合の設定です。
WR SET TEXT PROPERTY(Area;Alignment)
- ・ On load フォームイベントにおいて、メニューとルーラーを隠します。
If(Form event=On load)
WR SET DOC PROPERTY(Area;wr view menubar;0)
End if

参照

WR SET TEXT PROPERTY

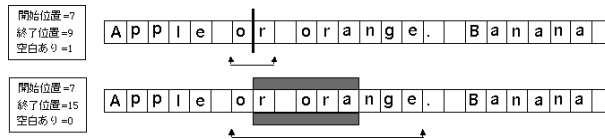
WR GET WORD

WR GET WORD (エリア;開始位置;終了位置;空白あり)

引数	タイプ	説明
エリア	倍長整数	4D Write エリア
開始位置	倍長整数	返される単語の開始位置
終了位置	倍長整数	返される単語の終了位置
空白あり	整数	0=最終文字が空白でない場合 1=最終文字が空白の場合

説明

WR GET WORDは、選択範囲内の最初の単語の開始文字位置と最後の単語の終了文字位置を返します。また、選択範囲の終了文字が空白かどうかを指定することができます。何もテキストが選択されていない場合は、<開始位置>と<終了位置>はカーソル上の単語の開始文字位置と終了文字位置を返します。



このコマンドは、現在選択されている部分に何の影響も与えません。

選択された範囲が単語の途中（または単語の最終文字の次に来る文字が空白である場合）から開始されている場合は、<開始位置>はその単語の開始文字位置を返します。

選択範囲が単語の途中で終わっている場合は、次の2つのケースが考えられます：

単語の次に来る文字が空白の場合は、<終了位置>は空白を含んだ位置を返し、<空白あり>は1を返します。

単語の次に来る文字が空白でない場合は、<終了位置>はその単語の終了文字位置を返し、<空白あり>は0を返します。

例題

次の例はドキュメントをスキャンし、各単語の開始文字位置と終了文字位置を取り出します。

```
` エリアの先頭にカーソルを置く
WR SET SELECTION (Area;0;0)
` ドキュメント内の単語数を数える
nbPara=WR Count (Area;wr nb words)
` 1語ずつ単語を処理する
For ($i;1;nbWord)
` 開始文字位置と終了文字位置を取り出す
WR GET PARAGRAPHS (Area;begin;Pos)
` 最後に処理された単語の後にエリアを再配置する
WR SET SELECTION (Area;Pos;Pos)
End for
```

参照

なし

WR INSERT STYLED TEXT

WR INSERT STYLED TEXT (エリア;blob)

引数	タイプ	説明
エリア	倍長整数	4D Write エリア
blob	Blob	変数またはフィールド

説明

WR INSERT STYLED TEXT は、<エリア> で参照される 4D Write エリアの中に < blob > の内容を挿入します。この挿入処理はカーソルの位置で起こるか、または blob の内容と現在選択されている部分を置き換えるかのどちらかです。< blob > は、BLOB 変数または BLOB フィールドのどちらかです。ただし < blob > が **WR Get styled text** を使って最初に作成されていることが必須条件となります。

< blob > 内で書式化されたテキストを表すために用いられる内部フォーマットは、プラットフォームに依存しません。そのため、MacOS 搭載のコンピュータを使って作成した blob コンテンツを、後で Windows ドキュメントの中に挿入することができます。また、その逆も可能です。

例題

この例は、最も頻繁に用いられるビジネスレターのテンプレートをテーブル[Letters]の中にストアし、ハードディスクの空きに保存したい場合を示します。'Templates' という名前の BLOB フィールドをテーブル中に作成します。テーブルの入力フィールドには、'Area' という名前の 4D Write エリアを挿入します。つまり、以下のメソッドをフォームに埋め込みます。

Case of

```
:(Form event=On Load)
  If (Record number([Letters])#-3)
    WR INSERT STYLED TEXT(Area:[Letters]Templates)
```

End if

```
:(Form event=On Data Change)
  WR EXECUTE COMMAND(Area;wr cmd select all)
  [Letters]Templates:=WR Get styled text(Area)
```

End case

参照

WR Get styled text、WR INSERT TEXT

WR INSERT TEXT

WR INSERT TEXT (エリア;先頭文字位置)

引数	タイプ	説明
エリア	倍長整数	4D Write エリア
挿入テキスト	テキスト	挿入されるテキスト

説明

WR INSERT TEXT は、選択された文字を置き換えたテキストを <エリア> に挿入します。文字が選択されていない場合、<先頭文字位置> はカーソルポイントに置かれます。このコマンドは、自動参照を行いたくない場合に **WR INSERT EXPRESSION** や **WR INSERT FIELD** の代わりに使用されます。

例題

次の例は、<エリア> の中に「vtext」変数内のテキストを挿入します。

WR INSERT TEXT (Area;vText)

参照

なし

WR Mouse to Selection

WR Mouse to Selection (エリア;横位置;縦位置;開始;終了) 結果

引数	タイプ	説明
エリア	倍長整数	4D Write エリア
横位置	整数	エリア内のマウスの横位置
縦位置	整数	エリア内のマウスの縦位置
開始	倍長整数	選択範囲の開始位置
終了	倍長整数	選択範囲の終了位置
戻り値	整数	カーソル位置に一致する選択範囲

説明

このコマンドは、4Dで得る **MOUSE POSITION** を4D Write上のカーソル位置に変換するルーチンです。例えば、4Dのフィールドから4D Writeエリアにオブジェクトをドラッグ&ドロップする場合、**MOUSE POSITION** でマウス位置を取得することが可能ですが、4D Writeのどの位置にドロップするかを知ることは4D側からは取得できません。そこで、このコマンドを使用し、4DWrite内の位置を取得し、その場合にピクチャやテキストをペーストします。

例題

4D Writeエリアには背景ピクチャを持つボタンがあると仮定します。ボタンのオブジェクトメソッドは背景ピクチャのドラッグ&ドロップを"B"地点にシミュレートします(カーソルは"A"地点に置かれているものとします)。

```
AP PICT DRAGGER ([Letters]Picture;PosH;PosV)
NPictureNot:=WR Mouse to selection(Field6;PosH;PosV;StartSel;EndSel)
WR SET SELECTION(Field6;StartSel;FiEndSelSel)
im:=[Letters]Picture*0,5
WR INSERT PICTURE(Field6;Im)
```

参照

GET MOUSE

WR Replace

WR Replace (エリア;検索文字列;置換文字列;完全一致;大文字検索;検索範囲;置換範囲)

引数	タイプ	説明
エリア	倍長整数	4D Write エリア
検索文字列	テキスト	検索される文字列
置換文字列	テキスト	置き換えられる文字列
完全一致	整数	完全に一致する検索
大文字検索	整数	大文字/小文字の区別を考慮する
検索範囲	整数	0= 選択範囲内の検索 1= ドキュメント全体を検索
置換範囲	整数	0= 次の置換のみ 1= 全置換
戻り値	倍長整数	置き換えられた数

説明

WR Replace は、「編集」メニューの「置換」コマンドと同じ働きをします。

<完全一致> が1の場合、完全に一致した単語のみ検索します。このオプションを使った場合、検索される文字列は（空白やカンマなどの）句読文字で囲まれている必要があります。

<大文字検索> が1の場合、検索は大文字と小文字を区別せずに元の文字列と一致する文字列を検索します。

<検索範囲> は、検索がドキュメント全体に適用されるかどうかを定義することができます。<検索範囲> が1の場合、検索はドキュメント全体に対して行われます。これに対して、<検索範囲> が0の場合は現在のカーソル位置からドキュメントの終わりまでの範囲で検索が行われます。

例題

ドキュメント内の不必要な連続した2つのスペース文字を取り除きたいものとします。

```

`連続した2つのスペース文字を含む変数
ToFind=" "
`それが見つかる間...
While(WR Find(Area;ToFind;0;0;1)=1)
`連続した2つのスペース文字を1つのスペース文字に置き換える
$N:=WR Replace(Area;ToFind;" ";0;0;1;0)
End while
    
```

参照

WR SELECT

WR SELECT

WR SELECT (エリア;タイプ;開始{;終了})

引数	タイプ	説明
エリア	倍長整数	4D Write エリア
タイプ	整数	選択タイプ
開始	倍長整数	開始文字位置
終了	倍長整数	終了文字位置、 <タイプ> によってはオプションの場合あり

説明

WR SELECT は、<タイプ>、<開始>、<終了> によって設定されたテキストを選択します。

タイプ	選択	コメント
0	文字	<開始> と <終了> の間にある文字を選択します。このケースでは、 WR SET SELECTION を使った場合と同じ結果となります。
1	リファレンス	<開始> に置かれたリファレンスを選択します。<終了> は省略しなければなりません。
2	段落	<開始> と <終了> の間にある段落を選択します。1つのピクチャも1つの段落と見なします。
3	ルーラー (段落属性)	(テキストの先頭から数えた) N番目のルーラーを使用する段落を選択します。<終了> は省略しなければなりません。
4	ピクチャ (文字属性)	<開始> に置かれたピクチャを選択します。<終了> は省略しなければなりません。
5	スタイル	(テキストの先頭から数えた) N番目のスタイルを使用する単語を選択します。<終了> は省略しなければなりません。
6	単語	カーソルポイントが置かれている単語を選択します。
7	ページブレイク	<開始> と <終了> の間にあるページブレイクを選択します。1つのページのみを選択したい場合は、<開始> と <終了> に同じ値を渡します。
8	カラムブレイク	<開始> と <終了> の間にあるカラムブレイクを選択します。1つのカラムのみを選択したい場合は、<開始> と <終了> に同じ値を渡します。
9	ハイフン	<開始> と <終了> の間にあるハイフンを選択します。1つのハイフンのみを選択したい場合は、<開始> と <終了> に同じ値を渡します。
10	ページ番号	<開始> と <終了> の間にあるページ番号を選択します。1つのページ番号のみを選択したい場合は、<開始> と <終了> に同じ値を渡します。
11	日付と時刻	<開始> と <終了> の間にある日付と時刻の変数を選択します。1つのページ番号のみを選択したい場合は、<開始> と <終了> に同じ値を渡します。

例題

次の例は、あるページブレークの有無によって異なる関数を実行します。

```
`選択範囲の設定
WR SET SELECTION (Area;0;0)
`最初のページブレークを選択
WR SELECT (Area;7;1;1)
`新しい選択範囲を取り出す
WR GET SELECTION (Area;$vbegin;$vlend)
If (($vbegin=0) & ($vlend=0))
`ページブレークなし
Else
`ページブレークを使って何らかの処理を行う
End if
```

次の例は、<エリア>で参照される4D Writeエリア内のリファレンスを選択し、その参照を見つけやすいようにスタイルを適用します。

```
NbObjects:=WR Count (Area;4)
`リファレンス数をカウントする
For (i;1;NbObjects)
  WR SELECT (Area;1;i)
  `各リファレンスを選択する
  WR GET REFERENCE (Area;TableNo;FieldNo;vName;vType)
  WR SET TEXT PROPERTY (Area;wr bold;1)
  WR SET TEXT PROPERTY (Area;wr text color;wr blue)
  `選択範囲に青色とボールドを適用する
End for
```

参照

WR Count、WR Replace

WR SET FONT

WR SET FONT (エリア;フォント)

引数	タイプ	説明
エリア	倍長整数	4D Write エリア
フォント	文字列	フォントの名前

説明

WR SET FONT は、<エリア>によって参照される4D Write エリア内の現在選択されている範囲にフォントを指定することができます。

<フォント>には、使用したいフォントの名前を渡します。

例題

WR Get font 関数の例題を参照してください。

参照

WR FONTS TO ARRAY、WR Get font、WR SET STYLESHEET FONT

WR SET SELECTION

WR SET SELECTION (エリア;先頭;最終)

引数	タイプ	説明
エリア	倍長整数	4D Write エリア
先頭	倍長整数	先頭文字位置
最終	倍長整数	最終文字位置

説明

WR SET SELECTION は、<エリア>内の<先頭>の位置にある文字と<最終>の位置にある文字の間にあるテキストを選択します。選択されるテキストは、<先頭>に1プラスした文字から<最終>も位置にある文字までです。

<先頭>と<最終>が同じ場合、**WR SET SELECTION**は<先頭>の位置にある文字の後ろにカーソルポイントを置きます。<最終>の値がエリア内のテキストの長さよりも大きいと、**WR SET SELECTION**は、そのドキュメントの終わりにあるテキストを選択します。<終了>が<先頭>より小さい場合は、**WR SET SELECTION**は何も行いません。

例題

次の例は、vText変数の中にエリア内の先頭から10文字を選択します。

```
vText := WR Ger selected text (Area;0;10)
```

参照

なし

WR SET TEXT PROPERTY

WR SET TEXT PROPERTY (エリア;プロパティ;値)

引数	タイプ	説明
エリア	倍長整数	4D Write エリア
プロパティ	整数	セットするプロパティ番号
値	実数	選択されたプロパティの値

説明

WR SET TEXT PROPERTY は、<エリア>で参照される 4D Write エリア内で現在選択されている範囲のテキストプロパティを修正することができます。

WR SET TEXT PROPERTY に参照するスタイルシートを渡した場合、テキストはそのスタイルシートを使ってラベルが付けられますが、そのスタイルシートのプロパティはそのテキストには適用されないことに注意してください。それに対して、**WR APPLY STYLESHEET** はテキストにプロパティを設定すると同時に適用します。

<プロパティ>と<値>は、組み合わせて使用されます。

- ・ <プロパティ>の値が定数プロパティである wr bold または 0 の場合、<値>の値は 1(True) または 0(False) のいずれかになります。
- ・ <プロパティ>の値が定数プロパティである wr font size または 8 の場合、<値>の値は 9,10,12... になります。

無効なプロパティ番号を渡すと、エラー番号 1075 が生成されます。

選択されたプロパティに対して無効な値を渡すと、エラー番号 1076 が生成されます。

注：<プロパティ>と<値>は定数を使ってセットされます。

テキストプロパティの定数一覧は付録 D の「WR Text properties」を参照してください。

テキストプロパティの値一覧は付録 D の「WR Text properties values」を参照してください。こちらは、値または定数のどちらかを渡すことができます。

エラーコード一覧は、付録 C の「エラーコード」を参照してください。

例題

1. 選択範囲内に、以下のフォント設定を適用したいものとします：

Osaka フォント、12ポイント、紫、斜体なし、太字

```
Violet:=WR RGB to color(61952;2048;33792)
WR SET FONT(Area;"Osaka")
WR SET TEXT PROPERTY(Area;wr font size;12)
WR SET TEXT PROPERTY(Area;wr text color;Violet)
WR SET TEXT PROPERTY(Area;wr bold;1)
WR SET TEXT PROPERTY(Area;wr italic;0)
```

2. 前もって定義されていた値をマージンに設定したいものとします。

```
WR GET SELECTION(Area;StartSel;EndSel)
  `現在のテキストの選択範囲をストア
WR UPDATE MODE(Area;0)          `スクリーンの更新を無効にする
WR EXECUTE COMMAND(Area;wr cmd select all)  `すべてを選択
  `ドキュメントの単位を cm に設定
WR SET DOC PROPERTY(Area;wr unit;0)
  `ドキュメントのマージンを cm で設定
WR SET TEXT PROPERTY(Area;wr right margin;1,8)
WR SET TEXT PROPERTY(Area;wr left margin;1,3)
WR SET SELECTION(Area;StartSel;EndSel)  `セレクションに戻るように設定
WR UPDATE MODE(Area;1)  `スクリーンの更新を有効にする
```

参照

WR Get text property

WR COLOR TO RGB

WR COLOR TO RGB (色;赤;緑;青)

引数	タイプ	説明
色	倍長整数	色
赤	倍長整数	赤の値を受け取る
緑	倍長整数	緑の値を受け取る
青	倍長整数	青の値を受け取る

説明

WR COLOR TO RGB は、色の3要素である < 赤 >、< 緑 >、< 青 > の中に < 色 > によって定義された色をマッピングします。< 色 > は4D Writeで使用される内部番号で、**WR RGB to color** で取得することができます。

例題

次の例は、灰色を求めます。

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

参照

WR RGB to color

WR Count

WR Count (エリア;オブジェクト番号)		結果
引数	タイプ	説明
エリア	倍長整数	4D Write エリア
オブジェクト番号	整数	オブジェクト番号
戻り値	倍長整数	オブジェクト数

説明

WR Count は、指定エリア内にある指定されたオブジェクトの数をカウントすることができます。

カウントできるオブジェクトには、次のものがあります：

オブジェクト	オブジェクト番号
文字	0
単語	1
段落	2
テキスト内のピクチャ	3
リファレンス	4
ハイフン	5
ページブレイク	6
段組	7
時刻オブジェクト	8
ページ番号	9
行間	10
ページ	11
スタイルシート	12
ページ内のイメージ (バックグラウンド)	13

< オブジェクト番号 > が 3 の場合、バックグラウンドのピクチャは無視されます (バックグラウンドにあるピクチャをカウントしたい場合は、< オブジェクト番号 > は 13 にする必要があります)。

< オブジェクト番号 > が 12 の場合、**WR Count** は標準のスタイルシート (デフォルトスタイルシート) を含んだスタイルシートの数を返します。

< オブジェクト番号 > が 13 の場合、任意のイメージが (「ピクチャプロパティ」ダイアログで選択された場合と同じように) 複数のページで繰り返されている場合、そのイメージは 1 つとしてカウントされます。

例題

WR SELECT コマンド、**WR INSERT PAGE** コマンド、**WR DELETE PICTURE IN PAGE** コマンド、**WR GET WORDS** コマンド、**WR GET PARAGRAPHS** コマンド、**WR UPDATE STYLESHEET** コマンドの例題を参照してください。

参照

WR Replace、WR SELECT

WR Error number

WR Error number (エリア) 結果

引数	タイプ	説明
エリア	倍長整数	4D Write エリア
戻り値	整数	スタイルシート参照番号

説明

WR Error number は、4D Write によって <エリア> 内で実行された最終操作のステータスを意味する番号を返します。つまり、**WR Error number** が 0 を返した場合は、最終操作では何のエラーもなかったということになります。しかし、**WR Error number** が 0 を返さなかった場合は、<エリア> 内の最終操作を行っている途中で何らかのエラーが発生したことになります。

エラーメッセージを取得するには、**WR Error text** を使用します。また、デバッグウィンドウが開いている際にエラーが発生すると、そのデバッグウィンドウ内にエラー番号が表示されます。

例題

WR Error text 関数の例題を参照してください。

参照

なし

WR Error text

WR Error text (エラー) 結果

引数	タイプ	説明
エラー	整数	エラー番号
戻り値	文字列	エラーメッセージ

説明

WR Error text は、<エラー> で指定されたエラーメッセージが返されます。この関数を使って、**WR Error number** によって返されたエラーメッセージを受け取ることができます。

例題

次の例は、エラーをチェックして、ユーザが Designer (デザイナー) かどうかによって異なるエラーメッセージを表示します。

```
$Error:=WR Error number (Area)
If ($Error#0)
  If (Current user="Designer")
    ALERT (WR Error text ($Error))
  Else
    ALERT ("エラーが発生しました。管理者にご相談ください。")
  End if
End if
```

参照

なし

WR FONT TO ARRAY

WR FONT TO ARRAY (フォント)

引数	タイプ	説明
フォント	文字配列	使用可能なフォントの配列を受け取る

説明

WR FONT TO ARRAY は、<フォント> 配列内で使用可能なフォント一覧を返します。この一覧は「フォント」メニュー内のフォントに対応します。

<フォント> は、文字またはテキストタイプの配列で宣言されなければなりません。

例題

テンプレートに使用したいフォントが現在のシステムにインストールされているかどうかをチェックしたいものとします。[Fonts]テーブルは要求されたフォントのリストを保存します。「On Startup」データベースメソッドで書き込むことができます。

```
ARRAY STRING (256;aFonts;0)
WR FONTS TO ARRAY (aFonts)
ALL RECORDS([Fonts])
While(Not(End selection([Fonts])))
  If (Find in array(aFonts;[Fonts]Name)=-1)
    ALERT("フォント : "+[Fonts]Name+"が要求されました。
    インストールしてください。")
  End if
  NEXT RECORD([Fonts])
End while
```

参照

WR SET FONT

WR ON ERROR

WR ON ERROR (メソッド)

引数	タイプ	説明
メソッド	文字列	メソッドの名前

説明

WR ON ERROR は、4D Writeのエラーを管理するメソッドをインストールします。<メソッド> が空の文字列の場合、メソッドは呼び出されません。インストールを行ない、4D Writeエラーが発生すると、<メソッド> が呼び出されます。4D Writeが<メソッド> を呼び出すと、エラーを管理するために使用される次の3つのパラメータを渡します。

変数	タイプ説明
\$1	倍長整数エラーが発生した 4D Write エリア
\$2	整数エラー番号
\$3	テキストエラーメッセージ

データベースをコンパイルする場合は、**C_LONGINT** と **C_TEXT** を使って、変数 \$1、\$2、\$3 を宣言する必要があります。

例題

次の例は、エラー処理用のメソッドをインストールしています。

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

次のメソッドは、WRITE ERRORメソッドです。これはエラーが発生した場所を調べるために \$1 を検査しています。WRITE ERRORメソッドは、エラー番号とエラーメッセージを警告ボックスに表示します。

```

If ($1=Area)
  ALERT ("4D WriteArea : "+Area+"でエラーが発生しました。")
End if
ALERT ("エラー番号"+String($2)+Char(13)+$3)

```

参照

なし

WR ON EVENT

WR ON EVENT (エリア;イベント;メソッド)

引数	タイプ	説明
エリア	倍長整数	4D Write エリア
イベント	整数	イベントコード
メソッド	文字	実行するメソッド

説明

WR ON EVENTは、<イベント>のイベントが<エリア>内で発生する度に呼び出される<メソッド>をインストールします。イベントは、4D Writeで処理される前に直接メソッドに渡されます。

<エリア>が0の場合、<メソッド>はデータベースが閉じられるまですべての4D Write エリアに対するデフォルトのイベントメソッドとなります。任意のエリアに特定のイベントメソッドがインストールされている場合は、そのメソッドはデフォルトメソッドに代わって呼び出されます。

次の表は、<イベント>で使用可能な値の一覧です。

イベント	値
イベントメソッドがすべてのイベントで呼び出される	-1
キー押下(矢印キー、returnキー、tabキー等を含む)	0
ダブルクリック	1
マウスクリック	2
未使用(4D Calcとの互換性用)	3
未使用(4D Calcとの互換性用)	4
4D Writeエリアのアクティブまたは非アクティブ	5
未使用(4D Calcとの互換性用)	6
ドキュメントの印刷	7
ルーラーの変更	8
参照内容の変更	9
4D Writeエリアのクローズ	10

メソッドを呼び出すと、<メソッド>はイベント時に<エリア>の状態を表す6つのパラメータを受け取ります。コンパイル用のコマンドを使って、これらのパラメータをきちんとタイプ宣言する必要があります。次の表は、メソッドで受け取れるパラメータの説明です。

変数	タイプ	説明
\$1	倍長整数	4D Writeエリア
\$2	整数	shiftキー
\$3	整数	Alt(Windows上)、option(MacOS上)
\$4	整数	Ctrl(Windows上)、commandキー(MacOS上)
\$5	整数	イベントタイプ
\$6	整数	イベントタイプにより異なる

\$1はイベントが発生した場所のエリアIDを表す倍長整数を返します。\$2、\$3、\$4は特定のモディファイアキーがイベント時に押されていなかったかどうかを判定します。これらの値が0の場合は、モディファイアキーは押されなかったことを意味し、1の場合は押されたことを意味します。\$5は、イベントタイプを返します。\$5は、イベントタイプによって異なります。

メソッド変数とイベントパラメータ(\$6)

<イベント> が0の場合、\$6はそのイベントを呼び出しているキーのASCIIコードを返します。

<イベント> が1または2の場合は、\$6は任意の参照をシングルクリックまたはダブルクリックしたことを示します。\$6が0の場合は、何も参照は選択されてないことを意味し、1の場合は任意の参照が選択されたことを意味します。<メソッド> は、次のような操作の1つが行われた場合に呼び出されます：

任意リファレンスのシングルクリックまたはダブルクリック

Altキーを押しながらクリック (Windows上) またはoptionキーを押しながらクリック (Macintosh上)

「Shift+Alt」キーを押しながらクリック (Windows上) または「shift+option」キーを押しながらクリック (Macintosh上)

Macintosh上では、optionキーを押しながらクリックした時と「shift+option」キーを押しながらのクリックした時には、通常任意のポップアップメニューを表示します。一方、Windows上ではAltキーを押しながらクリックすると通常任意のドロップダウンメニューを表示します。

<イベント> が5の場合、\$6は4D Writeエリアがアクティブ (使用可能状態) かどうかを判定します。

\$6が0の場合は4D Writeエリアは非アクティブな状態を意味し、1の場合はアクティブ状態を意味します。

<イベント> が7で印刷ジョブが差し込みプリントの場合、\$6は使用されるテーブルのテーブル番号を示します。印刷ジョブが差し込みプリントでない場合は、\$6は0となります。

<イベント> が8の場合、\$6はドキュメント内でリセットされているマージンの場所を示します。\$6が0の場合、マージンはボディ内でリセットされ、1の場合はヘッダ上でリセットされます。

入力された文字をフィルタにするには、<メソッド>を0または1を返す関数として使用しなければなりません。これにより、4D Writeが無視するドキュメント内の文字を指定することができます。

特定のイベントをトラップするメソッドを作成するには、\$0に初期値として1を渡します。特定のイベントをトラップしたくない場合は、\$0に初期値として0を渡します。例えば、ドキュメント内に“@”文字を表示したくない場合は、すべてのキーストロークをトラップし、\$6が“@”文字のASCIIコードと同じ場合、\$0に初期値として1を渡し、それを無視します（下記の例題も参照してください）。

注：すべての文字をトラップすると、メソッドが各キーストローク用に呼び出されるため、処理が非常に遅くなってしまいます。

例題

次の例は、イベントタイプによって実行される操作を示すフォームメソッドです。

```
If (Form event=On load)
  WR ON EVENT (Area;0;"ProcName")
  `すべてのキーストロークを呼び出す
  WR ON EVENT (Area;5;"ProcName")
  `エリアの状態をチェック
  DISABLE MENU ITEM (2;1)
  `「フォント変更」メニューアイテムを使用不可にする
End if
```

ProcNameメソッド：

```
Case of
  ¥ ($5=0)
  `キーストロークを中止する
  If ($6=199) | ($6=200)
  `対応するASCIIコード
  BEEP
  $0:=1
  Else
  `4D Writeにイベントを残す
  $0:=0
  End if
  ¥ ($5=5)
  `エリアステータスの変更を中止する
  If ($6=0)
```



```
        `エリアが非アクティブの場合  
        DISABLE MENU ITEM (2;1)  
    Else  
        ENABLE MENU ITEM (2;1)  
    End if  
End case
```

参照

WR ON ERROR

WR RGB to color

WR RGB to color (赤;緑;青) 結果

引数	タイプ	説明
赤	倍長整数	赤の要素 Windows上：0から255 Macintosh上：0から65535
緑	倍長整数	緑の値を受け取る Windows上：0から255 Macintosh上：0から65535
青	倍長整数	青の値を受け取る Windows上：0から255 Macintosh上：0から65535
戻り値	倍長整数	カラー

説明

WR RGB to colorは、色を管理するために4D Writeで使用される色の内部番号を返します。この番号はカラーの3要素である<赤>、<緑>、<青>を意味しています。<赤>、<緑>、<青>は使用しているシステムのカラーピッカーで用いられている値と同じものです。

次の表は、カラーに使用される<赤>、<緑>、<青>の一般的な値です。

カラー	Macintosh			Windows		
	赤	緑	青	赤	緑	青
赤	56576	2048	1536	221	8	6
緑	0	32768	4352	0	128	17
青	0	0	54272	0	0	212
シアン	512	43776	59904	2	171	234
マゼンタ	64512	62208	1280	252	243	5
イエロー	61952	2048	33792	242	8	132

例題

次の例は、2色の中間色を返します：

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)

参照

WR COLOR TO RGB

WR O Area to picture

WR O Area to picture (エリア)

引数	タイプ	説明
エリア	倍長整数	4D Write エリア

バージョン6.5での注意事項：この関数は、旧バージョンとの互換性のために残しているだけです。**WR Area to picture**の使用をお勧めします。

説明

WR O Area to pictureは、4D Write エリア内のドキュメントに含まれる4th Dimensionピクチャを返します。返されたピクチャは4D Writeが自動的にピクチャフィールドに格納される値と同じものです。この関数を使って、任意のフィールドに4D Write エリアを保存する、あるいはオフスクリーンエリアから4D Writeドキュメントを取り出すことができます。

注：この関数によって返されるピクチャは、ピクチャタイプです。この関数で返される結果は、4Dのピクチャ変数またはフィールドのどちらかに入れなければなりません。

参照

なし

WR O AUTO SAVE

WR O AUTO SAVE (エリア)

引数	タイプ	説明
エリア	倍長整数	4D Write エリア

バージョン 6.5 での注意事項：このコマンドは、旧バージョンとの互換性のために残しているだけです。**WR Area to picture** の使用をお勧めします。

説明

WR O AUTO SAVE は、それが存在すると、<エリア>内のドキュメントを同じ名前を持つピクチャフィールドに保存します。このコマンドは自動的に 4D Write エリアを任意のピクチャフィールドに保存します。4D Write エリアがプロシージャで変更される時はいつでも **WR O AUTO SAVE** を使用します。また、その変更が対応するピクチャフィールドに保存される際にも **WR O AUTO SAVE** を使用します。

参照

なし

WR O CHANGE STYLE

WR O CHANGE STYLE (エリア;削除;追加)

引数	タイプ	説明
エリア	倍長整数	削除されるスタイル
削除	倍長整数	削除されるスタイル
追加	倍長整数	追加されるスタイル

バージョン6.5での注意事項：このコマンドは、旧バージョンとの互換性のために残しているだけです。**WR SET TEXT PROPERTY**の使用をお勧めします。

説明

WR O CHANGE STYLE は、選択したテキストのスタイルを設定します。

<削除> と <追加> は、異なるスタイル番号を一緒に追加して取得した番号です。<追加> は、選択されたテキストに保持したいスタイルのパラメータです。<削除> は、選択したテキストに保持したくないスタイルのパラメータです。次の表は、スタイル番号を示しています。

スタイル	値
標準	0
太字	1
斜体	2
下線	4
アウトライン	8
シャドウ	16
上付き	32
下付き	64

バージョン6.0ユーザへの注意事項：4D Writeのバージョン6.5には「アウトライン」スタイルが存在しません。そのため、8の値が渡されても何もしません。

参照

WR SET TEXT PROPERTY

WR O COMPUTE NOW

WR O COMPUTE NOW (エリア)

引数	タイプ	説明
エリア	倍長整数	4D Write エリア

バージョン 6.5 での注意事項：このコマンドは、旧バージョンとの互換性のために残しているだけです。**WR EXECUTE COMMAND** の使用をお勧めします。

説明

WR O COMPUTE NOW は、4D Write ドキュメントの <エリア> の変数要素（例えば、挿入された式、マーカー等）を再計算します。

参照

なし

WR O Count stylesheet

WR O Count stylesheet (エリア)

引数	タイプ	説明
エリア	倍長整数	4D Write エリア

バージョン6.5での注意事項：この関数は、旧バージョンとの互換性のために残しているだけです。**WR Count**の使用をお勧めします。

説明

WR O Count stylesheet は、<エリア> で使用可能なスタイルシートの番号を返します。

参照

なし

WR O CREATE STYLESHEET

WR O CREATE STYLESHEET (エリア;スタイルシート番号;名前;フォント;
サイズ;スタイル;カラー)

引数	タイプ	説明
エリア	倍長整数	4D Writeエリア
スタイルシート番号	整数	スタイルシートの番号順に作成
名前	文字列	スタイルシートのタイトル
フォント	整数	選択されたフォントを返す
サイズ	整数	選択されたフォントサイズ
スタイル	整数	使用されたスタイル
カラー	倍長整数	使用されたカラー

バージョン6.5での注意事項：このコマンドは、旧バージョンとの互換性のために残しているだけです。**WR GET Creates stylesheet**の使用をお勧めします。

説明

WR O CREATE STYLESHEETは、新規にスタイルシートを作成し、<スタイルシート番号>で指定した場所にそのリストを挿入します。この新規スタイルシートは、<名前>、<フォント>、<サイズ>、<スタイル>、<カラー>の値を割り当てて定義します。

<名前>は、最大32バイトの名前を持つスタイルシートのタイトルです。

<フォント>は、フォントの番号です。**WR O Font name**を使って、そのフォントに対応している整数(フォントID)を取得します。

<サイズ>は、フォントをポイント単位で表したサイズです。このサイズは、1から127ポイントの間でなければなりません。127ポイントよりも大きいサイズを指定すると、それは無視されます。

<スタイル>は、使用されるスタイルの値を合計した数値で示されます。スタイルシートの一覧とそのスタイルの値については、**WR O SET STYLESHEET**コマンドの説明を参照してください。

<カラー>は、テキストのカラーを表す倍長整数です。**WR RGB to color**を使って、任意のカラーに対応した倍長整数を取得することができます。一般的に使用されるカラーとそれに対応する値については、**WR O GET STYLESHEET**コマンドの説明を参照してください。

参照

なし

WR O DELETE STYLESHEET

WR O DELETE STYLESHEET (エリア;スタイル番号)

引数	タイプ	説明
エリア	倍長整数	4D Write エリア
スタイル番号	整数	スタイルシート内のスタイル番号

バージョン6.5での注意事項：このコマンドは、旧バージョンとの互換性のために残しているだけです。**WR DELETE STYLESHEET**の使用をお勧めします。

説明

WR O DELETE STYLESHEETは、<スタイル番号>で指定されたスタイルシートを削除します。

参照

なし

WR O DISPLAY MENUBAR

WR O DISPLAY MENUBAR (エリア;モード)

引数	タイプ	説明
エリア	倍長整数	4D Write エリア
モード	整数	1=非表示 0=表示

バージョン6.5での注意事項：このコマンドは、旧バージョンとの互換性のために残しているだけです。**WR SET DOCUMENT PROPERTY**の使用をお勧めします。

説明

WR O DISPLAY MENUBAR は、<エリア>内のメニューバーを表示または非表示します。<モード>が1の場合、メニューバーは隠されます。0の場合、メニューバーは表示されます。

参照

なし

WR O DISPLAY RULER

WR O DISPLAY RULER (エリア;モード)

引数	タイプ	説明
エリア	倍長整数	4D Write エリア
モード	整数	1=非表示 0=表示

バージョン6.5での注意事項：このコマンドは、旧バージョンとの互換性のために残しているだけです。**WR SET DOCUMENT PROPERTY**の使用をお勧めします。

説明

WR O DISPLAY RULERは、<エリア>内のルーラーを表示または非表示します。<モード>が1の場合、ルーラーは隠されます。0の場合、ルーラーは表示されます。

参照

なし

WR O DISPLAY SCROLLBARS

WR O DISPLAY SCROLLBARS (エリア;表示)

引数	タイプ	説明
エリア	倍長整数	4D Write エリア
表示	整数	0=非表示 1=表示

バージョン6.5での注意事項：このコマンドは、旧バージョンとの互換性のために残しているだけです。**WR SET DOCUMENT PROPERTY**の使用をお勧めします。

説明

WR O DISPLAY SCROLLBARS は、4D Writeドキュメントのスクロールバーを表示または非表示にすることができます。

<表示> が1の場合、スクロールバーは表示されます。0の場合、スクロールバーは隠されます。

参照

なし

WR O DO COMMAND

WR O DO COMMAND (エリア;コマンド;モディファイア)

引数	タイプ	説明
エリア	倍長整数	4D Write エリア
コマンド	整数	コマンド番号
モディファイア	整数	押されたモディファイアキー

バージョン6.5での注意事項：

- ・ このコマンドは、旧バージョンとの互換性のために残しているだけです。WR EXECUTE COMMANDの使用をお勧めします。
- ・ また、WR O DO COMMANDでは、次のコマンドは機能しません。
 - 405 (アウトラインスタイル)
 - 702 (ホットリンク使用)
 - 708 (ホットリンク公開)

説明

WR O DO COMMANDは、<コマンド>で指定されたメニューアイテムを実行します。このメニューアイテムは、まるで4D Writeのメニューから選択されたかのように実行されます。これらの値は、たとえ将来において4D Writeのメニューアイテムが変更または移動された場合でも変わりません。

コマンドの値は、付録Bも「メニューアイテム番号」に一覧されています。

次のモディファイアキーは、キーの数値を加算して使用されます。

キー	値
command	256
shift	512
caps lock	1024
option	2048
control	4096

参照

WR EXECUTE COMMAND

WR O EXPERT COMMAND

WR O EXPERT COMMAND (エリア;コマンド;ステータス)

引数	タイプ	説明
エリア	倍長整数	4D Write エリア
コマンド	整数	コマンド番号
ステータス	整数	0=使用可 1=使用不可

バージョン6.5での注意事項：このコマンドは、旧バージョンとの互換性のために残しているだけです。**WR LOCK COMMAND**の使用をお勧めします。

説明

WR O EXPERT COMMANDは、メニューアイテムを使用可/使用不可にできます。<ステータス>が0の場合、メニューアイテムは使用可になります。1の場合、メニューアイテムは使用不可になります。**WR O EXPERT COMMAND**は、4D Writeで使用不可能になったコマンドを使用可能にすることはできません。<コマンド>の値一覧は付録Bの「メニューアイテム番号」を参照してください。

任意メニュー内のすべてのアイテムを使用不可にするには、メニュー番号の後ろに“00”を渡します。例えば、「ファイル」メニュー内のすべてのアイテムを使用不可にするには、<コマンド>に“100”を渡します。「データベース」メニューの場合では、「データベース」メニュー内のすべてのアイテムを使用不可にすると、この「データベース」メニューは表示されません。

参照

WR LOCK COMMAND

WR O Find

WR O Find (エリア;検索文字列{;検索範囲{;検索方法}) 結果

引数	タイプ	説明
エリア	倍長整数	4D Write エリア
検索文字列	テキスト	検索される文字列
検索範囲	整数	0=一部の単語 1=全単語
検索方法	整数	0=部分一致検索 1=完全一致検索
戻り値	倍長整数	0=検索文字列が見つかった場合 1=検索文字列が見つからなかった場合

バージョン6.5での注意事項：この関数は、旧バージョンとの互換性のために残しているだけです。WR Findの使用をお勧めします。

説明

WR O Findは、<エリア>内の<検索文字列>を検索し、見つかるとその文字列を選択します。<検索文字列>が見つからなかった場合、**WR O Find**は0を返します。見つかったら、1を返します。**WR O Find**は常に選択されたテキスト内の最終文字位置から検索を始めます。**WR O Find**は<検索文字列>を選択できるので、その検索位置を**WR GET SELECTION**から取得することができます。

オプションの<検索範囲>は、<検索文字列>が単語のどの部分と一致するのかを決定します。<検索範囲>が0の場合、**WR O Find**は<検索文字列>が単語の一部分でしかない場合であっても、その<検索文字列>を選択します。これに対して、<検索方法>が1の場合、**WR O Find**は<検索文字列>が単語のすべての部分と一致しないと<検索文字列>を選択しません。<検索文字列>はスペースや句読点などのセパレータ文字で囲まれます。

オプション<検索方法>は、検索を厳密に行うかどうかを決定します。<検索方法>が0の場合、**WR O Find**はあまり厳密な検索を行わないで<検索文字列>を選択します。これに対して、<検索方法>が1の場合、**WR O Find**は完全に一致した場合にしか<検索文字列>を選択しません。

参照

なし

WR O Font name

WR O Font name (フォント番号) 結果

引数	タイプ	説明
フォント番号	整数	フォント番号
戻り値	テキスト	<フォント番号>のIDを持つフォント名

バージョン6.5での注意事項：この関数は、旧バージョンとの互換性のために残しているだけです。4Dコマンドの使用をお勧めします。

説明

WR O Font nameは、<フォント番号>のIDを持つフォント名を返します。このIDは、**WR O Font number**によって返される値と同じです。<フォント番号>が存在しない場合、**WR O Font name**は空の文字列を返します。

参照

なし

WR O Font number

WR O Font number (フォント名) 結果

引数	タイプ	説明
フォント番号	整数	フォント名
戻り値	整数	<フォント名>に付けられたフォントのID

バージョン6.5での注意事項：この関数は、旧バージョンとの互換性のために残しているだけです。**WR Area to picture**の使用をお勧めします。

説明

WR O Font numberは、<フォント名>に付けられたフォント用の整数IDを返します。このIDは、**WR O SET ATTRIBUTES**内で使用されます。<フォント名>が存在しない場合、**WR O Font number**は0を返します。

参照

なし

WR O GET ATTRIBUTES

WR O GET ATTRIBUTES (エリア;フォント;スタイル;カラー)

引数	タイプ	説明
エリア	倍長整数	4D Write エリア
フォント	整数	フォント番号を返す
サイズ	整数	フォントサイズを返す
スタイル	整数	フォントスタイルを返す
カラー	整数	フォントカラーを返す

バージョン6.5での注意事項：このコマンドは、旧バージョンとの互換性のために残しているだけです。**WR Get text property**の使用をお勧めします。

説明

WR O GET ATTRIBUTESは、変数の中に4D Writeエリア内の選択されたテキストプロパティの<フォント>、<サイズ>、<スタイル>、<カラー>を返します。<エリア>内の選択されたテキストが複数のフォント、サイズ、スタイル、カラーを含んでいる場合は、**WR O GET ATTRIBUTES**はそのプロパティに-1を返します。

<フォント>は、使用しているシステム内のフォントIDです。これは、**WR Font number**で返される値と同じです。フォント名を調べたい場合は、**WR Font name**を使用します。

<サイズ>は、選択されたテキストをポイントサイズで表したものです。

<スタイル>は、いくつかのスタイル番号を加算した値です。各スタイルの番号を下記に示します。

スタイル	値
標準	0
太字	1
斜体	2
下線	4
アウトライン	8
シャドウ	16
上付き	32
下付き	64

バージョン6.0ユーザへの注意事項：4D Writeのバージョン6.5には「アウトライン」スタイルが存在しません。そのため、8の値が渡されても何もしません。

スタイルのアンダーラインとシャドウは、Windows上にはありません。スタイル属性が返す値は、マルチプラットフォームデータベースで使用されるプラットフォームに依存します。

<カラー>は、テキストのカラーを表す倍長整数です。この番号は、別のテキストに同じカラーをセットするために**WR O SET ATTRIBUTES**内で使用されます。

参照

WR Get font、WR Get text property

WR O GET MARGINS

WR O GET MARGINS (エリア;左;インデント;右)

引数	タイプ	説明
エリア	倍長整数	4D Write エリア
左	整数	左マージンをポイント単位で返す
インデント	整数	インデントをポイント単位で返す
右	整数	右マージンをポイント単位で返す

バージョン6.5での注意事項：このコマンドは、旧バージョンとの互換性のために残しているだけです。**WR Get text property**の使用をお勧めします。

説明

WR O GET MARGINS は、<エリア>内で現在選択されている段落のマージン設定を<左>、<インデント>、<右>で返します。複数の段落が選択されている場合は、このコマンドは選択範囲の1番目の段落に対する値を返します。

参照

なし

WR O Get pack options

WR O Get pack options (セレクト) 結果

引数	タイプ	説明
セレクト	整数	テンプレート、メニュー、ルーラー用のオプション 1=テンプレートを保存する場所 2=テンプレートを読み込む場所 3=メニュー 4=ルーラー 5=罫線 6=ページ表示 7=メッセージ抑止の再フォーマット
戻り値	整数	オプションの状態

バージョン6.5での注意事項：この関数は、旧バージョンとの互換性のために残しているだけです。WR Document propertyの使用をお勧めします。

説明

WR O Get pack options は、**WR O SET PACK OPTIONS** を使ってセットされるオプションの状態を返します。

<セレクト> が1の場合、**WR O Get pack options** はテンプレートが保存される場所を返します。返された値が1の場合は、テンプレートはサーバ上に保存されます。0の場合は、クライアント上に保存されます。

<セレクト> が2の場合、**WR O Get pack options** はテンプレートが読み込まれる場所を返します。返された値が1の場合は、テンプレートはサーバから読み込まれます。0の場合は、クライアントから読み込まれます。

<セレクト> が3の場合、**WR O Get pack options** はメニューが表示または非表示にされたかどうかを返します。返された値が1の場合は、メニューは表示されます。0の場合は、メニューは隠されます。

<セレクト> が4の場合、**WR O Get pack options** はルーラーが表示または非表示にされたかどうかを返します。返された値が1の場合は、ルーラーは表示されます。0の場合は、ルーラーは隠されます。

<セレクト> が5の場合、**WR O Get pack options** は4D Writeエリアの罫線を表示 / 非表示を指定することができます。値が1の場合は、罫線は表示されます。0の場合は、罫線は隠されます。デフォルトでは、罫線は表示 (値=1) されます。

<セレクト>が6の場合、4D Writeエリアが「ページプレビュー」モードを使って表示されるかどうかを指定することができます。返された値が1の場合は、「ページプレビュー」モードが使用されます。0の場合は、「ページプレビュー」モードが使用されません。デフォルトでは、「ページプレビュー」モードは使用(値=1)されます。

<セレクト>が7の場合は、「用紙設定」が変更される際に任意のドキュメントを再フォーマットする前に4D Writeエリアが警告メッセージを表示するかどうかを指定することができます。返された値が1の場合は、警告メッセージが表示されます。0の場合は、「警告メッセージ」は表示されません。デフォルトでは、「警告メッセージ」は表示(値=1)されます。

参照

なし

WR O Get page

WR O Get page (エリア;位置) 結果

引数	タイプ	説明
エリア	倍長整数	4D Write エリア
位置	倍長整数	<エリア>内の文字位置
戻り値	整数	<位置>に渡される文字位置のページ

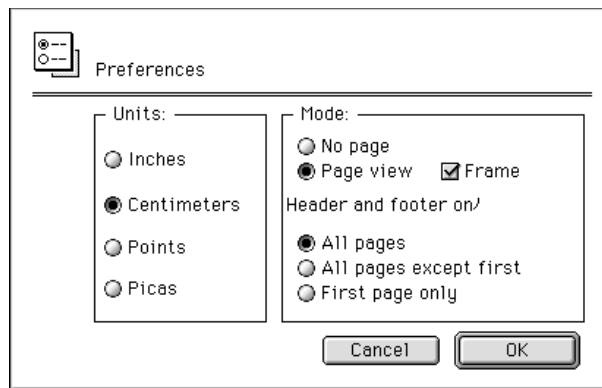
バージョン6.5での注意事項：この関数は、旧バージョンとの互換性のために残しているだけです。**WR GET CURSOR INFO**の使用をお勧めします。

説明

WR O Get pageは、<位置>に渡された文字位置のページを返します。例えば、100番目の文字が3ページ目にある場合は、下記コードの変数「v結果」の値は3になります。

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

この関数は、通常4D Writeの「編集」メニューから「環境設定」メニューアイテムを選択した場合にのみ有効となり、そのドキュメントを「ページプレビュー」モードに設定します。また、**WR SET PREFERENCES**を使って、ドキュメントモードを設定する、あるいは**WR GET PREFERENCES**を使って環境設定をチェックすることもできます。



参照

なし

WR O GET PICTURE

WR O GET PICTURE (エリア;高さ;幅;左上隅)

引数	タイプ	説明
エリア	倍長整数	4D Write エリア
高さ	整数	ピクチャの高さを返す
幅	整数	ピクチャの幅を返す
左上隅	整数	ピクチャの位置を返す

バージョン 6.5 での注意事項：このコマンドは、旧バージョンとの互換性のために残しているだけです。**WR GET PICTURE SIZE**の使用をお勧めします。

説明

WR O GET PICTURE は、<高さ>、<幅>、<左上隅> に選択されたピクチャの情報を返します。

<左上隅> は、そのピクチャの左上隅の水平座標の位置を示します。距離はルーラーの原点から計算され、ポイントで表されます。

参照

なし

WR O GET PREFERENCES

WR O GET PREFERENCES (エリア;モード;ページタイトル;単位)

引数	タイプ	説明
エリア	倍長整数	4D Write エリア
モード	整数	ページモードの値を返す
ページタイトル	整数	ページタイトルの値を返す
単位	整数	単位の値を返す

バージョン6.5での注意事項：このコマンドは、旧バージョンとの互換性のために残しているだけです。**WR Get document property**の使用をお勧めします。

説明

WR O GET PREFERENCESは、エリア内のドキュメントに現在設定されている内容を返します。これらの設定は、**WR O SET PREFERENCES**または4D Writeの「編集」メニューから「環境設定」を選択して行うことができます。

<モード>、<ページタイトル>、<単位>には、次のような値があります：

引数	値	説明
モード	0	「ページなし」モードのドキュメント
	1	「ページプレビュー」モードのドキュメント
	2	「罫線付きのページプレビュー」モードのドキュメント
ページタイトル	0	すべてのページのヘッダとフッタ
	1	先頭ページのみヘッダとフッタ
	2	先頭ページ除くその他すべてのページのヘッダとフッタ
単位	0	インチ
	1	センチメートル
	2	ポイント
	3	パイカ

参照

WR Get doc property

WR O GET RULER

WR O GET RULER (エリア;行間;行揃え)

引数	タイプ	説明
エリア	倍長整数	4D Write エリア
行間	整数	行間の値を返す
行揃え	整数	行揃えの値を返す

バージョン6.5での注意事項：このコマンドは、旧バージョンとの互換性のために残しているだけです。**WR Get text property**の使用をお勧めします。


説明

WR O GET RULER は、<行間>と<行揃え>の中に選択範囲用に選択した行間と行揃えの値を入れます。

<行揃え>は、次の4つの値の1つを持っています。

アイコン				
行揃え	0	1	2	3

値の1つをクリックすると、行間にスペースを1つ、あるいは1.5または2つ設定することができます。

アイコン			
行間	1	1.5	2

行間矢印をクリックして、2つ以上のスペースを行間に指定することができます。



<行間>は、次ページの表に示される通り、0から17までの18種類の整数値の1つです。例えば、値0は1.0の行間を表します。1.5から9.5までの行間の値を指定できます。

値	行間
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

値	行間
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

参照

なし

WR O Get ScrollBars

WR O Get ScrollBars (エリア) 結果

引数	タイプ	説明
エリア	倍長整数	4D Write エリア
戻り値	倍長整数	スクロールバーの状態

バージョン 6.5 での注意事項：この関数は、旧バージョンとの互換性のために残しているだけです。**WR Get document property** の使用をお勧めします。

説明

WR O Get ScrollBars は、4D Write ドキュメントのスクロールバーの状態を示す倍長整数を返します。

この関数が 1 を返したら、スクロールバーは表示されます。

この関数が 0 を返したら、スクロールバーは隠されます。

参照

なし

WR O GET STYLESHEET

WR O GET STYLESHEET (エリア;スタイルシート番号;名前;フォント;サイズ;スタイル;カラー)

引数	タイプ	説明
エリア	倍長整数	4D Write エリア
スタイルシート番号	整数	スタイルシート番号
名前	文字列	スタイルシートのタイトルを返す
フォント	整数	選択されたフォントを返す
サイズ	整数	選択されたフォントサイズを返す
スタイル	整数	使用されたスタイルを返す
カラー	倍長整数	使用されたカラーを返す

バージョン6.5での注意事項：このコマンドは、旧バージョンとの互換性のために残しているだけです。**WR GET STYLESHEET INFO**の使用をお勧めします。

説明

WR O GET STYLESHEET は、<スタイルシート番号>で指定したスタイルシートに対応する<名前>、<フォント>、<サイズ>、<スタイル>、<カラー>の値を返します。

<名前>は、最大32バイトの名前を持つスタイルシートのタイトルです。

<フォント>は、指定したスタイルシート内で使用されるフォント番号を返します。**WR O Font name**を使って、フォントIDでそのフォントの名前を取得することもできます。

<サイズ>は、指定したスタイルシート内で使用されるフォントサイズを返します。

<スタイル>は、使用されるスタイルの値を合計した数値が返されます。次の表は、スタイル一覧とそれに対応する値を示したものです。

スタイル	値
標準	0
太字	1
斜体	2
下線	4
アウトライン	8
シャドウ	16
上付き	32
下付き	64

<カラー>は、テキストのカラーを表す倍長整数です。**WR RGB to color**を使って、任意のカラーに対応した倍長整数を取得することができます。次の表は、一般的に使用されるカラーとそれに対応する値を示したものです。

カラー	Macintosh			Windows		
	赤	緑	青	赤	緑	青
赤	56576	2048	1536	221	8	6
緑	0	32768	4352	0	128	17
青	0	0	54272	0	0	212
シアン	512	43776	59904	2	171	234
マゼンタ	64512	62208	1280	252	243	5
イエロー	61952	2048	33792	242	8	132

参照

なし

WR O GET TAB

WR O GET TAB (エリア;タブ)

引数	タイプ	説明
エリア	倍長整数	4D Write エリア
タブ	2次元の整数配列	タブの配列を受け取る

バージョン6.5での注意事項：このコマンドは、旧バージョンとの互換性のために残しているだけです。**WR GET TAB**の使用をお勧めします。

説明

WR O GET TABは、<エリア>内で選択された段落のタブストップを<タブ>に返します。複数の段落が選択されている場合、**WR O GET TAB**はその選択が開始されている段落の値を返します。

タブは、3x10の2次元整数配列でなければなりません。**WR O GET TAB**は、<タブ>の大きさを変更できません。**WR O GET TAB**を実行すると、<タブ>内の3つの配列には以下のものが含まれます：

ポイント単位によるタブストップの位置

行揃えタイプ

埋め込み文字

タブストップの番号は、tabs{1}の0要素に返されます。

tabs{2}は、各タブの行揃えを示す番号を含んでいます。次の表は、tabs{2}で使用可能な値です。

行揃え	値
左揃え	0
右揃え	1
小数点区切り	2
中央揃え	3

tabs{3}は、各タブの埋め込み文字を示す番号を含んでいます。次の表は、tabs{3}で使用可能な値です。

行揃え	値
左揃え	0
右揃え	1
小数点区切り	2
中央揃え	3

注：WR O GET TAB内で使用される配列は、3x10の2次元整数配列でなければなりません。また、このコマンドでその配列の大きさを変更することはできません。配列要素{1}[0]をテストして、段落内のタブの数を調べることができます。

参照

なし

WR O INSERT HYPHEN

WR O INSERT HYPHEN (エリア)

引数	タイプ	説明
エリア	倍長整数	4D Write エリア

バージョン 6.5 での注意事項：このコマンドは、旧バージョンとの互換性のために残しているだけです。**WR EXECUTE COMMAND**の使用をお勧めします。

説明

WR O INSERT HYPHEN は、カーソルポイントにハイフンを挿入することができます。このカーソルポイントは、そのハイフンの前または後ろに置かれます。挿入されたハイフンは、単語が行の最終にいる場合にのみ表示されます。

ハイフンや句読文字がすでにカーソルポイントに表示されている場合は、**WR O INSERT HYPHEN** は何も行わずにエラーを返します。

参照

なし

WR O INSERT PICTURE

WR O INSERT PICTURE (エリア;ピクチャ)

引数	タイプ	説明
エリア	倍長整数	4D Write エリア
ピクチャ	ピクチャ	挿入するピクチャ

バージョン 6.5 での注意事項：このコマンドは、旧バージョンとの互換性のために残しているだけです。**WR INSERT PICTURE** の使用をお勧めします。

説明

WR O INSERT PICTURE は、選択した場所に任意のピクチャを挿入します。

参照

なし

WR O Is Hyphen

WR O Is Hyphen (エリア) 結果

引数	タイプ	説明
エリア	倍長整数	4D Write エリア
戻り値	倍長整数	1=カーソルポイントにハイフンがある場合 2=カーソルポイントにハイフンがない場合

バージョン 6.5 での注意事項：このコマンドは、旧バージョンとの互換性のために残しているだけです。**WR SELECT**の使用をお勧めします。

説明

WR O Is Hyphen は、カーソルポイントにハイフンが存在するかどうかを表す倍長整数を返します。カーソルポイントは、そのハイフンの前または後ろに置かれます。

この関数が 1 を返したら、カーソルポイントにハイフンがあります。

この関数が 0 を返したら、カーソルポイントにハイフンがありません。

参照

なし

WR O LINE SPACING

WR O LINE SPACING (エリア;行間)

引数	タイプ	説明
エリア	倍長整数	4D Write エリア
行間	整数	行間を示す値[0...17]

バージョン 6.5 での注意事項：このコマンドは、旧バージョンとの互換性のために残しているだけです。**WR SET TEXT PROPERTY**の使用をお勧めします。

説明

WR O LINE SPACING は、<行間>で示されている値を<エリア>内で選択されている段落の行間に設定します。<行間>は、0 から 17 までの整数値です。

参照

なし

WR O MENU STATUS

WR O MENU STATUS (エリア;コマンド;チェック;ステータス;名前)

引数	タイプ	説明
エリア	倍長整数	4D Write エリア
コマンド	整数	コマンド番号
チェック	整数	チェックされた状態を受け取る 0=チェックなし 7または195=チェックあり
ステータス	整数	アクティブの状態を受け取る 0=非アクティブ 1=アクティブ
名前	テキストまたは文字列	メニューアイテムの名前を受け取る

バージョン6.5での注意事項：このコマンドは、旧バージョンとの互換性のために残しているだけです。**WR GET COMMAND INFO**の使用をお勧めします。

説明

WR O MENU STATUSは、<コマンド>で示されたメニューアイテムがチェックされている、またはアクティブになっているかどうかを調べます。コマンド番号については付録Bの「メニューアイテム番号」を参照してください。**WR O MENU STATUS**を呼び出すと、<チェック>と<アクティブ>には<エリア>内の<コマンド>の状態を示す値が入ります。<アクティブ>が1の場合、メニューアイテムが使用可能になり、0の場合は使用不可になります。<チェック>が0の場合は、メニューアイテムはチェックされません。<チェック>が0以外の場合は、<チェック>内にASCIIコードで表された文字でそのメニューアイテムがチェックされます。

オプション<名前>は、テキストまたは文字列変数である必要があり、メニューアイテムのテキストを返します。

参照

WR GET COMMAND INFO

WR O MOVE PICTURE

WR O MOVE PICTURE (エリア;左上隅)

引数	タイプ	説明
エリア	倍長整数	4D Write エリア
左上隅	整数	ピクチャの新しい位置

バージョン 6.5 での注意事項：このコマンドは、旧バージョンとの互換性のために残しているだけです。整列ツールの使用をお勧めします。

説明

WR O MOVE PICTURE は、選択されたピクチャを水平方向に移動します。セレクションはそのピクチャ以外のものでも構成されてはいけません。

<左上隅> は、そのピクチャの左上隅の水平座標の位置を示します。距離は、ルーラーの原点から計算され、ポイントで表されます。

参照

なし

WR O ON MENU

WR O ON MENU (エリア;メソッド)

引数	タイプ	説明
エリア	倍長整数	4D Write エリア
メソッド	文字列	メソッドの名前

バージョン6.5での注意事項：このコマンドは、旧バージョンとの互換性のために残しているだけです。**WR ON COMMAND**の使用をお勧めします。

説明

WR O ON MENUは、4D Writeメニューを管理するメソッドとして<メソッド>をインストールします。<メソッド>が空の文字列の場合は、呼び出されるメソッドはありません。メニューメソッドをインストールしたら、すべてのメニュー動作をこの<メソッド>によって管理しなければなりません。

<エリア>が0の場合、**WR O ON MENU**はデータベースが閉じられるまで、すべての4D Writeエリアに適用されます。メソッドは、実行時に次のローカル変数を受け取ります：

変数	説明
\$1	メニューが選択された4D Writeエリアを示す倍長整数
\$2	選択されたメニューアイテム用のコマンド番号を示す倍長整数（付録B参照）
\$3	メニューアイテムの選択と同時に押されたモディファイアキーを示す倍長整数

コンパイラ用のコマンドを使って、明示的に変数\$1、\$2、\$3のタイプを宣言します。下記は使用できるモディファイアキーです。これらは、キーの値を合計して表されます。

キー	値
command	256
shift	512
caps lock	1024
option	2048
control	4096

注：**WR O ON MENU**コマンドによって使用されたコマンド番号は、このマニュアルに記述されていません。詳細は、4D Writeバージョン6.0のマニュアルを参照してください。または、**WR ON COMMAND**コマンドを使用してください。

参照

WR ON COMMAND

WR O OPTION

WR O OPTION (エリア;メッセージ;確認;印刷モード;フォーマット;取り消し)

引数	タイプ	説明
エリア	倍長整数	4D Write エリア
メッセージ	文字列	空のリファレンス用のメッセージ
確認	整数	0= 確認なし 1= 確認あり -1= 変更なし
印刷モード	整数	0= 固定サイズ 1= プリント領域可変 -1= 変更なし
フォーマット	整数	0=4D Write と PICT の先頭ページ 1=4D Write -1= 変更なし
取り消し	整数	0= 使用不可 1= 使用可 -1= 変更なし

バージョン6.5での注意事項：このコマンドは、旧バージョンとの互換性のために残しているだけです。**WR Area to picture**の使用をお勧めします。

説明

WR O OPTION は、<エリア>に機能的なパラメータを設定します。エリア名の代わりに<エリア>に0を指定すると、**WR O OPTION** は作成されたすべての新規エリアに適用されますが、影響を受けるのは<メッセージ>だけです。既存のエリアはこのコマンドによって影響を受けることはありません。

<メッセージ>は、4th Dimensionのフィールドや式のリファレンス（参照）が何も返さない場合に表示されるテキストのことです。<メッセージ>に割り当てるテキストは、ユーザが4th Dimensionを終了するか、または**WR O OPTION**を再度実行するまで使用されます。どちらの場合も、<メッセージ>はデフォルトの“{ }”に戻ります。またこのパラメータに何もテキストを割り当てなかった場合、<メッセージ>は“{ }”として表示されます。

4D Write エリアを作成してからそのエリアに何らかの変更を行うと、“このエリアは変更されています。保存しますか？”というメッセージが表示されます。**WR O OPTION**を使って、このメッセージの表示を使用不可にすることができます。

<確認> は、4D Writeエリアが保存されないで閉じられようとした際に確認のダイアログボックスを表示するかどうかを制御します。<確認>が0の場合、確認のダイアログボックスは表示されません。1の場合は、4D Writeは確認のダイアログボックスを表示します。-1の場合は、現在の設定内容がそのまま有効となります。

<印刷モード> は、<エリア>を含む任意のフォームが印刷される際にそのエリアがどのように処理されるのかを制御します。<印刷モード>が0の場合、<エリア>内のテキストはそのエリアサイズに切り取られてしまいます。これは、固定サイズの組み込みフォーム印刷の場合とまったく同じです。1の場合は、すべてのテキストを印刷するために<エリア>は次ページにまたがっても拡張されます。これは、プリント領域可変の組み込みフォーム印刷の場合とまったく同じです。-1の場合は、現在の設定内容がそのまま有効となります。

<取り消し> は、「編集」メニュー内の「取り消し」メニューアイテムを使用可または使用不可にします。このパラメータはオフスクリーンエリア内で使用します。オフスクリーンエリアはメモリに保持され、4D Writeの処理速度を高速にします。<取り消し>が0の場合、「取り消し」メニューアイテムは使用可能になります。1の場合は、使用不可になります。-1の場合は、現在の設定内容がそのまま有効となります。

参照

WR SET AREA PROPERTY

WR O Page number

WR O Page number {(エリア)} 結果

引数	タイプ	説明
エリア	倍長整数	4D Write エリア
戻り値	整数	現在印刷されているページ番号

バージョン6.5での注意事項：このコマンドは、旧バージョンとの互換性のために残しているだけです。**WR INSERT PAGE NUMBER**の使用をお勧めします。

説明

WR O Page number は、現在印刷されているページ番号を返します。この関数は4D Writeドキュメントのヘッダエリアまたはフッタエリア内、あるいは4D Writeドキュメントのヘッダまたはフッタエリアの内部から呼び出されるメソッド内でしか使用できません。**WR O Page number**を直接4D Writeエリア内で使用する場合、エリア名は省略できます。

参照

なし

WR O PICTURE TO AREA

WR O PICTURE TO AREA (エリア;ピクチャ)

引数	タイプ	説明
エリア	倍長整数	4D Write エリア
ピクチャ	ピクチャ	オープンするピクチャ

バージョン6.5での注意事項：このコマンドは、旧バージョンとの互換性のために残しているだけです。**WR PICTURE TO AREA**の使用をお勧めします。

説明

WR O PICTURE TO AREAは、<エリア>に<ピクチャ>内の4D Writeドキュメントを割り当てます。このコマンドを使って、手動で任意のフィールドからピクチャを読み込むことや、4D Writeドキュメントのオフスクリーンエリア内にピクチャを配置することができます。

参照

なし

WR O Picture to offscreen area

WR O Picture to offscreen arear (ピクチャ) 結果

引数	タイプ	説明
ピクチャ	ピクチャ	オープンするピクチャ
戻り値	整数	エリア参照番号

バージョン6.5での注意事項：このコマンドは、旧バージョンとの互換性のために残しているだけです。**WR Picture to area**の使用をお勧めします。

説明

WR O Picture to offscreen areaは、ユーザの目には見えない4D Writeエリア（オフスクリーンエリア）の中に<ピクチャ>内に含まれているドキュメントを配置し、この新規エリアにアクセスするために用いられる値を返します。

WR O Picture to offscreen areaによって返される値は、4D Writeエリアを必要とする4D Writeコマンドの中で使用されます。この関数は、**WR New offscreen area**と**WR O PICTURE TO AREA**を組み合わせたものと同じです。

注：オフスクリーンエリア上で何らかの操作を実行した場合は、必ず**WR DELETE OFFSCREEN AREA**を使用しなければなりません。

参照

なし

WR O REMOVE HYPHEN

WR O REMOVE HYPHEN (エリア)

引数	タイプ	説明
エリア	倍長整数	4D Write エリア

バージョン 6.5 での注意事項：このコマンドは、旧バージョンとの互換性のために残しているだけです。**WR SELECT**の使用をお勧めします。

説明

WR O REMOVE HYPHEN は、カーソルポイントにあるハイフンを取り除きます。カーソルポイントは、ハイフンの前または後ろに置かれます。

ハイフンがカーソルポイントに存在しない場合は、**WR O REMOVE HYPHEN** は何も行わずに、エラーを返します。

参照

なし

WR O Replace

WR O Replace (エリア;検索文字列;置換文字列;置換範囲;置換方法;モード) 結果

引数	タイプ	説明
エリア	倍長整数	4D Write エリア
検索文字列	テキスト	検索する文字列
置換文字列	テキスト	置き換える文字列
置換範囲	整数	0=部分一致 1=完全一致
置換方法	整数	0=大文字/小文字区別なし 1=大文字/小文字区別あり
モード	整数	0=次の置換 1=全置換
戻り値	倍長整数	置き換えた数

バージョン6.5での注意事項：このコマンドは、旧バージョンとの互換性のために残しているだけです。**WR Replace**の使用をお勧めします。

説明

WR O Replaceは、<エリア>内に存在する<検索文字列>を検索し、その文字を<置換文字列>で置き換えます。またこの関数は、置き換えた数を返します。**WR O Replace**は選択されたテキストの最終文字位置から置き換え処理を開始します。

<置換範囲>は、<検索文字列>が置換単語と完全一致するかどうかを調べます。<置換範囲>が0の場合、**WR O Replace**は<検索文字列>が置換文字の一部である場合に<検索文字列>を置き換えます。1の場合は、<検索文字列>が置換文字そのものの場合にしか置き換えません。単語全体とは、スペースや句読点などの区切り文字で囲まれた単語です。

<置換方法>は、置換処理を厳密に行うかどうかを決定します。<置換方法>が0の場合、**WR O Replace**は大文字/小文字等を区別を厳密に行わずに<検索文字列>を置き換えます。1の場合は、完全に一致した場合にしか置き換えません。

<モード>は、何回の置換処理を行うかを決定します。<モード>が0の場合、<検索文字列>は最初に見つかった箇所しか置き換えません。1の場合は、選択されたテキストの位置がどこであろうとも、<検索文字列>が見つかったすべての箇所を<置換文字列>で置き換えます。

参照

なし

WR O RESIZE PICTURE

WR O RESIZE PICTURE (エリア;新しい高さ;新しい幅)

引数	タイプ	説明
エリア	倍長整数	4D Write エリア
新しい高さ	整数	ピクチャの新しい高さ
新しい幅	整数	ピクチャの新しい幅

バージョン6.5での注意事項：このコマンドは、旧バージョンとの互換性のために残しているだけです。**WR SET PICTURE SIZE**の使用をお勧めします。

説明

WR O RESIZE PICTURE は、4D Write エリア内で選択されたピクチャのサイズを変更します。選択範囲は、選択されたピクチャだけにする必要があります。ピクチャが選択されていない場合は、エラーコード 1034 が生成されます。

<新しい高さ> は、変更するピクチャの高さをポイント単位で表します。同様に、<新しい幅> は、変更するピクチャの幅をポイント単位で表します。

一般的に Macintosh のモニタは、72dpi です。使用しているモニタの解像度が 82dpi 以上の場合は、使用する dpi の数値を決定するために case 文を用います。

<新しい高さ> と <新しい幅> が 0 より小さい場合、ピクチャのサイズは変更されません。<新しい高さ> と <新しい幅> が 0 の場合は、ピクチャは削除されます。

参照

なし

WR O Save to picture

WR O Save to picture (エリア) 結果

引数	タイプ	説明
エリア	倍長整数	4D Write エリア
戻り値	ピクチャ	4th Dimentionのピクチャ

バージョン6.5での注意事項：このコマンドは、旧バージョンとの互換性のために残しているだけです。**WR Area to picture**の使用をお勧めします。

説明

WR O Save to pictureは、<エリア>内のドキュメントに含まれている4th Dimentionのピクチャを返します。返されたピクチャは、4D Writeでピクチャフィールドに自動的に格納された値と同じものです。この関数は、<エリア>が手動でフィールドまたはディスク上に保存される時に有効です。また、この関数を使って、オフスクリーンエリアから4D Writeドキュメントを取り出すこともできます。

WR O Area to pictureと**WR O Save to picture**の違いはほんのわずかです。それは、**WR O Area to picture**がエリアに対するドキュメント保存フラグにTrueをセットするのに対して、**WR O Save to picture**はそれをセットしません。その結果、**WR O Area to picture**を呼び出して<エリア>内にあるウインドウやフォームを閉じることは、ユーザにそのドキュメントを保存するように4D Writeが指示しなくなります。

注：この関数によって返されるピクチャは、4th Dimentionのピクチャフィールドまたはピクチャ変数のどちらかに入れる必要があります。

参照

なし

WR O SET ATTRIBUTES

WR O SET ATTRIBUTES (エリア;フォント;サイズ;スタイル;カラー)

引数	タイプ	説明
エリア	倍長整数	4D Write エリア
フォント	整数	選択されたフォントを返す
サイズ	整数	選択されたフォントサイズを返す
スタイル	整数	使用されたスタイルを返す
カラー	倍長整数	使用されたカラーを返す

バージョン6.5での注意事項：このコマンドは、旧バージョンとの互換性のために残しているだけです。**WR SET TEXT PROPERTY**の使用をお勧めします。

説明

WR O SET ATTRIBUTESTは、<エリア>内の選択されたテキストプロパティに<フォント>、<サイズ>、<スタイル>、<カラー>の値を設定します。プロパティの1つをそのまま変更せずに残しておきたい場合は、そのプロパティの引数に-1を渡します。

<フォント>は、使用しているシステム内にあるフォントのIDです。この値は、**WR O Font number**から取得することができます。

<サイズ>は、ポイント単位で表されたフォントのサイズです。

<スタイル>は、使用されるスタイルの値を合計した数値が表されたスタイルです。次の表は、スタイル一覧とそれに対応する値を示したものです。

スタイル	値
標準	0
太字	1
斜体	2
下線	4
アウトライン	8
シャドウ	16
上付き	32
下付き	64

スタイルがテキストの選択範囲に対して設定されると、各スタイルは別々に適用され、その選択範囲は次の状況によって影響を受けます。

太字や斜体、アンダーライン等のスタイルが、選択範囲全体に影響を与える場合、**WR O SET ATTRIBUTEST**は、その選択範囲全体で選択されていないスタイル番号とスタイルを適用します。

スタイルが選択範囲の一部のみに影響を与えるか、まったく影響を与えない場合、**WR O SET ATTRIBUTEST**は、その選択範囲全体にそのスタイルを適用します。任意のスタイルが選択範囲全体に影響を与えるかどうかを調べるには、当該メニューアイテム用のチェックマーク文字のASCIIコード値を検査します。次の中の1つがTrueの場合、特定のスタイルが選択されたテキスト内で使用されているかどうかを調べることができます：

スタイルメニューアイテムがチェックマーク文字を持っていないため、選択範囲のどこにも影響を与えません。

スタイルメニューアイテムがチェックマーク文字 (ASCII 215) でチェックされているため、その選択範囲の一部に影響を与えます。

スタイルメニューアイテムがチェックマーク文字 (ASCII 195) でチェックされているため、その選択範囲全体に影響を与えます。

注：アウトラインとシャドウは、Windows には存在しません。テキストプロパティ用に返される値は、マルチプラットフォームデータベース上で使用されるプラットフォームに依存します。

<カラー> は、テキストのカラーを表す倍長整数です。次の表は、一般的に使用されるカラーとそれに対応する値を示したものです。

カラー	Macintosh			Windows		
	赤	緑	青	赤	緑	青
赤	56576	2048	1536	221	8	6
緑	0	32768	4352	0	128	17
青	0	0	54272	0	0	212
シアン	512	43776	59904	2	171	234
マゼンタ	64512	62208	1280	252	243	5
イエロー	61952	2048	33792	242	8	132

参照

WR SET FONT、WR SET TEXT PROPERTY

WR O SET MARGINS

WR O SET MARGINS (エリア;左;インデント;右)

引数	タイプ	説明
エリア	倍長整数	4D Write エリア
左	整数	左マージン (ポイント単位)
インデント	整数	インデント (ポイント単位)
右	整数	右マージン (ポイント単位)

バージョン6.5での注意事項：このコマンドは、旧バージョンとの互換性のために残しているだけです。**WR SET TEXT PROPERTY**の使用をお勧めします。

説明

WR O SET MARGINSは、<エリア>内の選択された段落のマージンに<左>、<インデント>、<右>の値を設定します。<左>、<インデント>、<右>の値は、ドキュメントの左端からポイント単位で表されます。

マージンの1つをそのまま残しておくには、そのパラメータに-1を渡します。

参照

なし

WR O SET PACK OPTIONS

WR O SET PACK OPTIONS (セレクトク;値)

引数	タイプ	説明
セレクトク	整数	テンプレート、メニュー、ルーラーのオプション 1=テンプレートの保存場所 2=テンプレートの読み込み場所 3=メニュー 4=ルーラー 5=罫線 6=ページレビュー 7=メッセージ消去の再フォーマット
値	整数	テンプレート用 1=サーバ 0=クライアント メニューとルーラー用 1=表示 0=非表示

バージョン6.5での注意事項：このコマンドは、旧バージョンとの互換性のために残しているだけです。**WR SET DOCUMENT PROPERTY**の使用をお勧めします。

説明

WR O SET PACK OPTIONSは、テンプレート、メニュー、ルーラー用のオプションを設定するために使用されます。**WR O SET PACK OPTIONS**で設定されたすべてのオプションは一時的なものです。そのオプションが常に有効であるようにするには、Startupメソッド内でオプションを設定します。

<セレクトク>が1の場合、テンプレートをどこに保存するかを指定できます。<値>が1の場合は、テンプレートはサーバ上に保存されます。<値>が0の場合は、テンプレートはクライアント上に保存されます。デフォルトでは、テンプレートはサーバ上(値=1)に保存されます。

<セレクトク>が2の場合、テンプレートが読み込まれる場所を指定できます。<値>が1の場合は、テンプレートはサーバから読み込まれます。0の場合は、クライアントから読み込まれます。デフォルトでは、テンプレートはサーバ上(値=1)から読み込まれます。

<セレクトク>が3の場合、メニューを表示または非表示にすべきかを設定できます。<値>が1の場合は、メニューが表示されます。0の場合は、メニューが隠されます。デフォルトでは、メニューは表示(値=1)されます。

<セレクトク>が4の場合、ルーラーを表示または非表示にすべきかを設定できます。

<値> が1の場合は、ルーラーが表示されます。0の場合は、ルーラーが隠されます。デフォルトでは、ルーラーは表示 (値=1) されます。

<セレクトア> が5の場合、4D Write エリアの罫線を表示または非表示にすべきかを設定できます。<値> が1の場合は、罫線が表示されます。0の場合は、罫線が隠されます。デフォルトでは、罫線は表示 (値=1) されます。

<セレクトア> が6の場合、4D Write エリアが「ページプレビュー」モードを使って表示されるかどうかを指定することができます。<値> が1の場合は、「ページプレビュー」モードが使用されます。0の場合は、「ページプレビュー」モードが使用されません。デフォルトでは、「ページプレビュー」モードは使用 (値=1) されます。

<セレクトア> が7の場合は、「用紙設定」が変更される際に任意のドキュメントを再フォーマットする前に4D Write エリアが警告メッセージを表示するかどうかを指定することができます。<値> が1の場合は、警告メッセージが表示されます。0の場合は、警告メッセージが表示されません。デフォルトでは、警告メッセージは表示 (値=1) されます。

WR O SET PACK OPTIONS は、プラグインウインドウ内のメニューやルーラーを隠す場合に特に効果的です。**Open external window** を使ってプラグインウインドウの作成を行う前にこの関数を使用することにより、そのプラグインウインドウが表示される前にメニューやルーラーを隠すことができます。それに対して、**WR O DISPLAY RULER** や **WR O DISPLAY MENUBAR** を使用すると、4th Dimension はプラグインウインドウが表示された後でそのメニューやルーラーを隠します。これは、ウインドウの再描画をおかしくする原因となります。

参照

なし

WR O SET PREFERENCES

WR O SET PREFERENCES (エリア;モード;先頭ページ;単位)

引数	タイプ	説明
エリア	倍長整数	4D Write エリア
モード	整数	0=ページなし 1=ページプレビュー 2=罫線付きのページプレビュー
先頭ページ	整数	0=標準 1=先頭ページ以外のヘッダとフッタ 2=先頭ページだけのヘッダとフッタ -1=変更なし
単位	整数	0=インチ 1=センチメートル 2=ポイント 3=パイカ -1=変更なし

バージョン6.5での注意事項：このコマンドは、旧バージョンとの互換性のために残しているだけです。**WR SET DOCUMENT INFO**の使用をお勧めします。

説明

WR O SET PREFERENCESは、<エリア>に含まれるドキュメントを表示するために使用することができるオプションを設定します。

パラメータ内のあるプロパティをそのまま残しておくには、そのプロパティのパラメータに-1を渡します。

参照

なし

WR O SET STYLESHEET

WR O SET STYLESHEET (エリア;スタイルシート番号;名前;フォント;サイズ;
スタイル;カラー)

引数	タイプ	説明
エリア	倍長整数	4D Write エリア
スタイルシート番号	整数	スタイルシート番号
名前	文字列	スタイルシートのタイトルを返す
フォント	整数	選択されたフォントを返す
サイズ	整数	選択されたフォントサイズを返す
スタイル	整数	使用されたスタイルを返す
カラー	倍長整数	使用されたカラーを返す

バージョン6.5での注意事項：このコマンドは、旧バージョンとの互換性のために残しているだけです。WR SET STYLESHEET INFOの使用をお勧めします。

説明

WR O SET STYLESHEETは、<スタイルシート番号>で指定したスタイルシートに対応する<名前>、<フォント>、<サイズ>、<スタイル>、<カラー>の値を設定します。

<名前>は、最大32バイトの名前を持つスタイルシートのタイトルです。

<フォント>は、指定したスタイルシート内で使用されるフォント番号です。WR O Font nameを使って、フォントIDでそのフォントの名前を取得することもできます。

<サイズ>は、指定したスタイルシート内で使用されるフォントサイズです。

<スタイル>は、使用されるスタイルの値を合計した数値です。次の表は、スタイル一覧とそれに対応する値を示したものです。

スタイル	値
標準	0
太字	1
斜体	2
下線	4
アウトライン	8
シャドウ	16
上付き	32
下付き	64

注：フォントのプロパティをそのまま残しておくには、そのプロパティの引数に-1を渡します。

<カラー>は、テキストのカラーを表す倍長整数です。**WR RGB to color**を使って、任意のカラーに対応した倍長整数を取得することができます。一般的に使用されるカラーとそれに対応する値の一覧は、**WR O GET STYLESHEET**コマンドの説明を参照してください。

参照

WR SET STYLESHEET INFO

WR O SET TABS

WR O SET TABS (エリア;元の位置;新しい位置;埋め込み文字;整列)

引数	タイプ	説明
エリア	倍長整数	4D Write エリア
元の位置	整数	元のタブ位置 (ポイント単位)
新しい位置	整数	新しいタブ位置 (ポイント単位)
埋め込み文字	整数	埋め込み文字の値[0...3]
整列	整数	整列の値[0...3]

バージョン6.5での注意事項：このコマンドは、旧バージョンとの互換性のために残しているだけです。**WR SET TABS**の使用をお勧めします。

説明

WR O SET TABSは、<エリア>内で選択された段落のタブストップを追加、修正、削除します。このコマンドは、<元の位置>で示されるタブストップを<新しい位置>で示されたタブストップに移動します。

<元の位置>が-1あるいは<元の位置>にタブストップがない場合は、**WR O SET TABS**は<新しい位置>で指定された位置に新しいタブを作成します。<新しい値>が-1の場合、**WR O SET TABS**は<元の位置>で指定された位置のタブを削除します。タブを作成または移動した場合は、そのタブは<埋め込み文字>と<整列>で修正されます。

<埋め込み文字>は、タブに使用する埋め込み文字を指定します。次の表は、使用可能な<埋め込み文字>の値を示しています。

埋め込み文字	値
なし	0
ドット	1
ダッシュ	2
アンダーライン	3

<整列>は、タブが持っている整列のタイプを指定します。次の表は、使用可能な<整列>の値を示しています。

整列	値
左に整列	0
中央に整列	1
右に整列	2
均等位置	3

参照

なし

WR O STATISTICS

WR O STATISTICS (エリア;文字数;段落数;オブジェクト数;ホットリンク数;修正;ページ数)

引数	タイプ	説明
エリア	倍長整数	4D Write エリア
文字数	倍長整数	文字数を受け取る
段落数	整数	段落数を受け取る
オブジェクト数	整数	オブジェクト数を受け取る
ホットリンク数	整数	ホットリンク数を受け取る
修正	整数	修正ステータスを受け取る
ページ数	整数	ページ数を受け取る

バージョン6.5での注意事項：このコマンドは、旧バージョンとの互換性のために残しているだけです。**WR Count**の使用をお勧めします。

説明

WR O STATISTICS は、4th Dimentionのオブジェクト数と一緒に<エリア>内の文字数、段落数、ホットリンク数を返します。またこのコマンドは、<エリア>内のページ数も返します。

WR O STATISTICS が呼び出されると、各引数は次の表で示されている値を持っています。

引数	返される値
文字数	引数<エリア>内の総文字数
段落数	引数<エリア>内の総段落数
オブジェクト数	4th Dimensionオブジェクト(フィールド、式、ホットリンク)の総数
ホットリンク数	引数<エリア>内のホットリンク数
修正	0=修正されていないドキュメント 1=修正されたドキュメント
ページ数	引数<エリア>内の総ページ数

参照

なし

WR O STRUCTURE ACCESS

WR O STRUCTURE ACCESS (エリア;モード)

引数	タイプ	説明
エリア	倍長整数	4D Write エリア
モード	整数	0=アクセスを許可する 1=アクセスを制限する

バージョン6.5での注意事項：このコマンドは、旧バージョンとの互換性のために残しているだけです。**WR LOCK COMMAND**の使用をお勧めします。

説明

WR O STRUCTURE ACCESS は、4th Dimension フィールドへのアクセスを制御します。<モード>が0の場合は、4D Writeは「フィールド挿入」メニューアイテムを使用できるようになります。しかし、<モード>が1の場合は、「データベース」メニュー内の「フィールド挿入」メニューアイテムは使用不可になる上、「テーブル-フィールド」ポップアップメニューへのアクセスに用いるキーボードショートカットを取り消します。このコマンドは、デザイナーに対してデータベース内の特定フィールドへのアクセスを制限することができます。

「テーブル-フィールド」ポップアップメニューへのアクセスを可能にするには、次のようにします：

Windows上では、右マウスボタンをクリックする。

Macintosh上では、controlキーを押したままマウスボタンを押す。

参照

WR LOCK COMMAND

WR O TEXT ALIGNMENT

WR O TEXT ALIGNMENT (エリア;整列)

引数	タイプ	説明
エリア	倍長整数	4D Write エリア
整列	整数	0=左に整列 1=中央に整列 2=右に整列 3=均等位置

バージョン6.5での注意事項：このコマンドは、旧バージョンとの互換性のために残しているだけです。**WR SET TEXT PROPERTY**の使用をお勧めします。

説明

WR O TEXT ALIGNMENT は、<エリア>内で選択された段落の整列に<整列>で示された値を設定します。次の表は、整列用に使用できる値を示しています。

整列	値
左に整列	0
中央に整列	1
右に整列	2
均等位置	3

参照

なし

付録 A : ショートカット

特殊キー

スクロール処理の他に、キーボードの中には4D Write ドキュメントの場所を素早く見つけることができるキーを持っているものがあります。

キー	説明
Home	ドキュメントの先頭を表示します。
End	ドキュメントの最終を表示します。
Page Up	1ページ分先にスクロールします (現在選択されている部分を変更するわけではありません)。
Page Down	1ページ分後ろにスクロールします (現在選択されている部分を変更するわけではありません)。
Ctrl(またはcommand)+ 左矢印キー	カーソルポイントを現在選択されている単語の先頭に移動します。 現在選択されている単語の先頭に既にカーソルポイントがある場合は、1つ前の単語の先頭に移動します。
Ctrl(またはcommand)+ 右矢印キー	カーソルポイントを現在選択されている単語の最終に移動します。 現在選択されている単語の最終に既にカーソルポイントがある場合は、1つ後の単語の先頭に移動します。
Ctrl(またはcommand)+ Delete	カーソル右にある単語を削除します。
Ctrl(またはcommand)+ Backspace	カーソル右にある単語を削除します。
shift (カーソルポイント や表示領域を移動する ために上記の任意の キーと組み合わせて 使用)	現在選択されている部分を拡張します。

クリックコンビネーション

スクロール処理の他に、4D Write では次のクリックコンビネーションを使用できます。

コンビネーション	説明
シングルクリック	選択されたテキストを選択解除するためにカーソルポイントを移動します。
ダブルクリック	ダブルクリックされた単語を選択します。
トリプルクリック	段落を選択します。
左マージンのクリック	クリックした次の線を選択します。
左マージンのダブルクリック	クリックした次の段落を選択します。
shift+クリック	現在選択されている部分をクリックした位置まで拡張します。

付録 B : メニューアイテム番号

次の表は、4D Writeの各メニューアイテムのコマンド値を一覧表示したものです。これらの番号は、4D Writeのバージョンがアップされた際、メニューアイテムが修正または移動された場合でも、そのまま保持されます。詳細は、「**WR EXECUTE COMMAND**」の説明を参照してください。また次のコードは、**WR ON COMMAND**および**WR GET COMMAND INFO**で使用することもできます。これらのコマンドを使用する際、ユーザはメニューアイテムまたは定数のどちらでも受け渡すことができます。

メニュー	ツールバー表示	コマンド	リファレンス	
ファイル	あり	新規	101	
	あり	開く	102	
	あり	保存	103	
	なし	新規保存...	104	
	なし	テンプレートとして保存	110	
	なし	環境設定...	105	
	なし	用紙設定...	106	
	あり	プリントプレビュー	107	
	あり	プリント...	108	
	なし	差し込みプリント...	109	
	なし	フルウィンドウを開く / フォームに戻る		
	編集	あり	取り消しできません	1
		あり	やり直しできません	2
あり		カット	3	
あり		コピー	4	
あり		ペースト	5	
なし		クリア	6	
なし		すべてを選択	7	
あり		検索...	208	
なし		次を検索	209	
なし		置換	210	
なし		次を置換	211	
なし		文字変換	220	
なし		/小文字	221	
なし		/大文字	222	
なし		/先頭大文字	223	
なし		/先頭以外大文字	224	
なし		選択範囲を表示	309	
なし	ページへ移動...	807		

表示	なし	通常	302	
	なし	ページ	303	
	なし	ツールバー	330	
	なし	/標準	331	
	なし	/書式	332	
	なし	/スタイル	333	
	なし	/罫線	334	
	なし	ルーラ	311	
	なし	ヘッダ	312	
	なし	フッタ	313	
	なし	参照	314	
	なし	ピクチャ	315	
	あり	非表示文字	316	
	なし	枠	317	
	なし	横スクロールバー	318	
	なし	縦スクロールバー	319	
	なし	メニューバー	310	
	なし	ステータスバー	320	
	挿入	なし	日付と時刻...	401
		あり	現在時刻挿入	411
あり		現在日付挿入	412	
なし		ページ番号...	402	
なし		特種文字...	409	
なし		ソフトハイフン	404	
なし		ノンブレイクスペース	405	
なし		改段組	410	
なし		ページブレイク	406	
なし		4D式...	407	
スタイル		なし	標準	501
	あり	太字	502	
	あり	斜体	503	
	なし	シャドウ	504	
	なし	取り消し線	505	
	なし	下線		
	なし	/下線なし	521	
	なし	/単下線	522	
	なし	/単語下線	523	
	なし	/二重下線	524	
	なし	/点下線	525	
	あり	下線ボタン	530	

	なし	上付き	506
	なし	下付き	507
	なし	英大文字	508
	なし	小型英大文字	509
カラー	テキスト		
	/602 黒		
	/603 赤		
	/604 オレンジ		
	/605 黄色		
	/606 緑		
	/607 青		
	/608 紫		
	/609 白		
	/610 薄い灰色		
	/611 灰色		
	/612 濃い灰色		
	/613 その他...		
	背景		
	/628 なし		
	/61 白		
	/617 赤		
	/618 オレンジ		
	/619 黄色		
	/620 緑		
	/621 青		
	/622 紫		
	/623 薄い灰色		
	/624 灰色		
	/625 濃い灰色		
	/626 黒		
	/627 その他...		
	取り消し線		
	/632 自動		
	/633 黒		
	/634 赤		
	/635 オレンジ		
	/636 黄色		
	/637 緑		
	/638 青		
	/639 紫		

/640 白
/641 薄い灰色
/642 灰色
/643 濃い灰色
/644 その他...

下線

/646 自動
/647 黒
/648 赤
/649 オレンジ
/650 黄色
/651 緑
/652 青
/653 紫
/654 白
/655 薄い灰色
/656 灰色
/657 濃い灰色
/658 その他...

シャドウ

/662 薄い灰色
/656 灰色
/657 濃い灰色
/658 その他...

段落背景

/684 なし
/672 白
/673 赤
/674 オレンジ
/675 黄色
/676 緑
/677 青
/678 紫
/679 薄い灰色
/680 灰色
/681 濃い灰色
/682 黒
/683 その他...

罫線

/686 黒

	/687	赤	
	/688	オレンジ	
	/689	黄色	
	/690	緑	
	/691	青	
	/692	紫	
	/693	白	
	/694	薄い灰色	
	/695	灰色	
	/696	濃い灰色	
	/697	その他...	
段落	なし	ル - ラーコピー	701
	なし	ル - ラ - ペースト	702
	なし	箇条書き	
	なし	/なし	1021
	なし	/黒四角	1022
	なし	/白四角	1023
	なし	/黒丸	1024
	なし	/白丸	1025
	なし	/ダイヤモンド	1026
	なし	/クローバー	1027
	なし	その他...	1028
	あり	左に整列	711
	あり	中央に整列	712
	あり	右に整列	713
	あり	均等位置	714
	あり	1行間	721
	あり	1.5行間	722
	あり	2行間	723
	なし	その他...	724
書式	なし	文字...	751
	なし	段落...	752
	なし	タブ...	753
	なし	罫線...	754
	あり	左罫線	1005
	あり	上罫線	1006
	あり	右罫線	1007
	あり	下罫線	1008
	あり	全罫線	1009
	あり	内罫線	1010

	あり	罫線なし	1011
	なし	スタイルシート...	755
	なし	段組...	756
ツール	なし	テーブルウイザード...	408
	なし	スペル...	805
	なし	言語...	806
	なし	ドキュメント情報...	801
	なし	ドキュメント統計...	802
	なし	リファレンスをいま計算	803
	なし	リファレンスを凍結	804

付録 C : エラーコード

下記は4D Writeによって返されるエラーコードの一覧です：

エラーコード	エラーメッセージ
1002	印刷中にエラーが起きました。
1003	左マージンパラメータが無効です（右マージンを閉じることができません）。
1004	インデントパラメータが無効です（右マージンを閉じることができません）。
1005	右マージンパラメータが無効です（左マージンまたはインデントを閉じることができません）。
1006	タブパラメータが無効です。
1007	配列パラメータが無効です：配列が有効なタイプまたはサイズでないか、または配列自体がありません。
1012	ファイルが保存されてません。
1013	選択範囲が無効です（開始が0以下か、あるいは開始が終了より大きい値です）。
1015	ファイルが読み込まれていません。
1016	メニューまたはアイテム参照が無効です。
1017	このフィールドは、4D Writeのフィールドにはなれないみたいです。
1022	外部コマンドに渡されるエリアパラメータが無効です。
1023	4D ファイル参照番号が無効です。
1024	4Dのテキスト変数およびフィールドは、最大32000バイトです。
1028	WR Select関数に渡される位置が無効です。
1032	このファイルは存在しません。
1034	ピクチャが選択されていません。
1035	サイズパラメータが無効です。
1036	位置パラメータが無効です。
1038	このスタイルは存在しません。
1041	このコマンドを実行するにはメモリが足りません。
1044	イベントタイプが無効です。
1047	フィールド参照が無効です。
1048	オプション番号が無効です。
1051	このパスは存在しません。
1054	第1パラメータが無効です。
1055	第2パラメータが無効です。
1056	第3パラメータが無効です。
1058	第5パラメータが無効です。
1060	サブフィールドを挿入できません。
1066	256以上のタブストップは作成できません。

- 1067 タブの位置が無効です。
- 1068 タブの整列が無効です。
- 1069 Blobを挿入できません。
- 1072 削除するハイフンはありません。
- 1073 式が無効です。
- 1074 Blobが無効です。
- 1075 テキストプロパティが範囲外です。
- 1076 テキストプロパティの値が範囲外です。
- 1077 フォントがシステム内にありません。
- 1078 不明のスタイルシートです。
- 1079 ドキュメントプロパティが範囲外です。
- 1080 ドキュメントプロパティの値が範囲外です。
- 1081 印刷中に選択範囲が変更されました。
- 1082 距離の値が無効です。
- 1083 ページ番号内のピクチャが無効です。
- 1084 タブ番号が無効です。
- 1085 ページ番号の書式が範囲外です。
- 1086 ページ番号が無効です。
- 1087 カラム番号が無効です。
- 1088 ライン番号が無効です。
- 1089 オプション番号が無効です。
- 1090 統計番号が無効です。
- 1091 枠線リファレンスが無効です。
- 1092 コマンド番号が無効です。
- 1093 印刷できません：ドキュメントはすでに印刷中です。
- 1094 スタイルシートを保存しました。
- 1095 ファイルを開けません。

付録 D : 4D Write 定数

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 violet1	3369599
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	
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

A

WR ADD STYLESHEET TAB(エリア;スタイルシート番号;位置;{行揃え; 埋め込み文字})	93
WR ADD TAB(エリア;位置;行揃え;埋め込み文字).....	85
WR APPLY STYLESHEET(エリア;スタイルシート番号)	95
WR Area to blob(エリア{;保存ドキュメント}) 結果	45
WR Area to picture(エリア;保存ドキュメント;プレビュー) 結果.....	46

B

WR BACKSPACE(エリア).....	115
WR BLOB TO AREA(エリア;blob).....	48
WR Build preview(エリア;ページ) ピクチャ.....	31

C

WR COLOR TO RGB(色;赤;緑;青).....	143
WR Count(エリア;オブジェクト番号) 結果.....	144
WR Create stylesheet(エリア;名前{;適用{;ショートカット}) 結果.....	96

D

WR DELETE OFFSCREEN AREA(エリア).....	49
WR DELETE PICTURE IN PAGE(エリア;ピクチャ番号).....	69
WR DELETE SELECTION(エリア)	116
WR DELETE STYLESHEET(エリア;スタイルシート番号)	98
WR DELETE STYLESHEET TAB(エリア;スタイルシート番号;タブ番号).....	99
WR DELETE TAB(エリア;タブ番号)	87
WR Direct find(blob;検索文字列;検索方法;検索範囲) 結果	117

E

WR Error number(エリア) 結果	146
WR Error text(エラー) 結果	147
WR EXECUTE COMMAND(エリア;コマンド番号)	18

F

WR Find(エリア;検索文字列;検索方法;大文字検索;検索範囲) 結果	119
WR FONT TO ARRAY(フォント)	148

G

WR GET AREA PROPERTY(エリア;オプション;値;文字値)	33
WR GET COMMAND INFO(エリア;コマンド番号;数値戻り値;文字戻り値; 名前;ステータス)	19
WR GET CURSOR POSITION(エリア;ページ;段組;行;位置)	35
WR Get doc property(エリア;プロパティ)	21
WR GET DOCUMENT INFO(エリア;題名;件名;著者;組織;ノート;作成日; 作成時刻;修正日;修正時刻;ロック)	59
WR Get font(エリア;同フォント)	121
WR Get frame(エリア) 結果	37
WR GET PARAGRAPHS(エリア;開始段落;終了段落)	122
WR GET PICTURE IN PAGE INFO(エリア;ピクチャ番号;ページ;背面; 先頭ページ;水平位置;垂直位置;幅;高さ;元の幅;元の高さ)	70
WR GET PICTURE SIZE(エリア;幅;高さ;元の幅;元の高さ)	72
WR GET REFERENCE(エリア;テーブル;フィールド;名前;タイプ)	53
WR Get selected picture(エリア;ステータス) 結果	73
WR Get selected text(エリア) 結果	123
WR GET SELECTION(エリア;先頭;最終)	124
WR Get styled text(エリア) 結果	125
WR Get stylesheet font(エリア;スタイルシート番号)	100
WR GET STYLE SHEET INFO(エリア;スタイルシート番号;名前;適用; ショートカット)	108
WR GET STYLE SHEET INFO(エリア;スタイルシート番号;名前;適用; プロテクト;ショートカット)	101
WR GET STYLE SHEET TAB(エリア;スタイルシート番号;タブ番号;位置; 行揃え;埋め込み文字)	103
WR Get stylesheet text prop(エリア;スタイルシート番号;プロパティ; 適用)	105

WR GET TAB(エリア;タブ番号;位置;行揃え;埋め込み文字)	88
WR Get text(エリア;先頭;最終) 結果	126
WR Get text property(エリア;プロパティ;同プロパティ)	127
WR GET WORD(エリア;開始位置;終了位置;空白あり).....	130

I

WR INSERT EXPRESSION(エリア;式).....	55
WR INSERT FIELD(エリア;テーブル;フィールド).....	56
WR INSERT PAGE NUMBER(エリア;フォーマット)	81
WR Insert picture area(エリア;ピクチャ;場所) エラー	57
WR INSERT PICTURE(エリア;ピクチャ;挿入先;水平位置;垂直位置;背面; 先頭ページ).....	74
WR INSERT STYLED TEXT(エリア;blob).....	132
WR INSERT TEXT(エリア;先頭文字位置).....	133

L

WR LOCK COMMAND(エリア;コマンド番号;ロック).....	22
WR LOCK DOCUMENT(エリア;ステータス)	61

M

WR Mouse to Selection(エリア;横位置;縦位置;開始;終了) 結果	134
---	-----

N

WR New offscreen area 結果.....	50
-------------------------------	----

O

WR O Area to picture(エリア).....	155
WR O AUTO SAVE(エリア)	156
WR O CHANGE STYLE(エリア;削除;追加).....	157
WR O COMPUTE NOW(エリア)	158
WR O Count stylesheet(エリア)	159
WR O CREATE STYLESHEET(エリア;スタイルシート番号;名前;フォント; サイズ;スタイル;カラー).....	160
WR O DELETE STYLESHEET(エリア;スタイル番号).....	161
WR O DISPLAY MENUBAR(エリア;モード)	162
WR O DISPLAY RULER(エリア;モード).....	163
WR O DISPLAY SCROLLBARS(エリア;表示)	164

WR O DO COMMAND(エリア;コマンド;モディファイア)	165
WR O EXPERT COMMAND(エリア;コマンド;ステータス)	166
WR O Find(エリア;検索文字列{;検索範囲{;検索方法}) 結果.....	167
WR O Font name(フォント番号) 結果	168
WR O Font number(フォント名) 結果.....	169
WR O GET ATTRIBUTES(エリア;フォント;スタイル;カラー).....	170
WR O GET MARGINS(エリア;左;インデント;右).....	172
WR O Get pack options(セレクタ) 結果.....	173
WR O Get page(エリア;位置) 結果	175
WR O GET PICTURE(エリア;高さ;幅;左上隅)	176
WR O GET PREFERENCES(エリア;モード;ページタイトル;単位)	177
WR O GET RULER(エリア;行間;行揃え)	178
WR O Get ScrollBars(エリア) 結果	180
WR O GET STYLESHEET(エリア;スタイルシート番号;名前;フォント; サイズ;スタイル;カラー).....	181
WR O GET TAB(エリア;タブ)	183
WR O INSERT HYPHEN(エリア).....	185
WR O INSERT PICTURE(エリア;ピクチャ).....	186
WR O Is Hyphen(エリア) 結果	187
WR O LINE SPACING(エリア;行間).....	188
WR O MENU STATUS(エリア;コマンド;チェック;ステータス;名前)	189
WR O MOVE PICTURE(エリア;左上隅).....	190
WR O ON MENU(エリア;メソッド).....	191
WR O OPTION(エリア;メッセージ;確認;印刷モード;フォーマット; 取り消し)	192
WR O Page number{(エリア)} 結果	194
WR O PICTURE TO AREA(エリア;ピクチャ)	195
WR O Picture to offscreen arear(ピクチャ) 結果	196
WR O REMOVE HYPHEN(エリア)	197
WR O Replace(エリア;検索文字列;置換文字列;置換範囲;置換方法; モード) 結果	198
WR O RESIZE PICTURE(エリア;新しい高さ;新しい幅).....	199
WR O Save to picture(エリア) 結果	200
WR O SET ATTRIBUTES(エリア;フォント;サイズ;スタイル;カラー)	201
WR O SET MARGINS(エリア;左;インデント;右)	203
WR O SET PACK OPTIONS(セレクタ;値).....	204
WR O SET PREFERENCES(エリア;モード;先頭ページ;単位)	206

WR O SET STYLESHEET(エリア;スタイルシート番号;名前;フォント; サイズ;スタイル;カラー).....	207
WR O SET TABS(エリア;元の位置;新しい位置;埋め込み文字;整列)	209
WR O STATISTICS(エリア;文字数;段落数;オブジェクト数; ホットリンク数;修正;ページ数).....	210
WR O STRUCTURE ACCESS(エリア;モード)	211
WR O TEXT ALIGNMENT(エリア;整列).....	212
WR ON COMMAND(エリア;4Dメソッド).....	24
WR ON ERROR(メソッド).....	149
WR ON EVENT(エリア;イベント;メソッド).....	150
WR OPEN DOCUMENT(エリア;ドキュメント{;タイプ})	62

P

WR PICTURE TO AREA(エリア;ピクチャ).....	51
WR PRINT (エリア;モード;コピー数).....	83
WR PRINT MERGE(エリア;テーブル;表示).....	84

R

WR REDRAW(エリア).....	25
WR Replace(エリア;検索文字列;置換文字列;完全一致;大文字検索; 検索範囲;置換範囲).....	135
WR RGB to color(赤;緑;青) 結果	154

S

WR SAVE DOCUMENT(エリア;ドキュメント;タイプ).....	64
WR SCROLL TO SELECTION(エリア)	17
WR SELECT(エリア;タイプ;開始{;終了})	137
WR SELECT PICTURE IN PAGE(エリア;ピクチャ番号).....	76
WR SET AREA PROPERTY(エリア;オプション;値;文字値).....	38
WR SET CURSOR POSITION(エリア;ページ;段組;行;位置)	40
WR SET DOC PROPERTY(エリア;プロパティ;値)	26
WR SET DOCUMENT INFO(エリア;題名;件名;著者;組織;ノート).....	66
WR SET FONT(エリア;フォント).....	139
WR SET FRAME(エリア;フレーム).....	42
WR SET PICTURE IN PAGE INFO(エリア;ピクチャ番号;ページ;背面; 先頭ページ;水平位置;垂直位置;幅;高さ).....	77
WR SET PICTURE SIZE(エリア;幅;高さ)	80

WR SET SELECTION(エリア;先頭;最終)	140
WR SET STYLESHEET FONT(エリア;スタイルシート番号;フォント).....	107
WR SET STYLESHEET TAB(エリア;スタイルシート番号;タブ番号;位置; 行揃え;埋め込み文字)	110
WR SET STYLESHEET TEXT PROP(エリア;スタイルシート番号; プロパティ;値;適用).....	112
WR SET TAB(エリア;タブ番号;位置;行揃え;埋め込み文字).....	90
WR SET TEXT PROPERTY(エリア;プロパティ;値)	141

T

WR TEXT ACCESS(エリア;モード)	43
-------------------------------	----

U

WR UPDATE MODE(エリア;モード).....	29
WR UPDATE STYLESHEET(エリア;スタイルシート番号)	114