

# WORDPERFECT 10 MACRO COMMANDS

© As To Editing, Not Content, 2001 by Doug Loudenback, Oklahoma City, Oklahoma, All Rights Reserved To Editing, Not To Content

In its PDF format, this document is 122 pages long. In its HTML format, the whole file is broken into 3 HTML files, not using columns, and the total document will be **much** longer. Most commands are also appropriate for earlier versions of WordPerfect, particularly WordPerfect 8 and 9, but some are not, particularly for WordPerfect 6.1 and 7.0. Some few commands are new to WordPerfect 10. No attempt is made here to distinguish or explain variations in, or even the existence of, particular macro commands for pre-WordPerfect 10 macro programming.

The sole sources for this document are text documents initially made using the "command dump" feature in the Command Browser feature of WordPerfect 10's PerfectScript Utility. After dumping both the WordPerfect Product and PerfectScript commands, the text was re-ordered (to put the [brief explanatory notes between square braces] after instead of before the commands). If a note reads, "[No note was included in the dump]", a note was not present for the command in the dump and I've added that note. The dumps were combined into a single WordPerfect document which was reformatted and hyperlinked for easier use. No editing has been done to the substance of commands or the associated notes. I've made no attempt to determine the accuracy of various statements. Commands are grouped by 2 categories: WordPerfect Product Commands (including System Variables) and PerfectScript Commands. For ease of visual identification, **I've formatted a command's name to be red bold text**. In various instances that possible alternative parameters were only 2 or 3 (e.g., Yes! or No!), I've placed them on the same line of text rather than stacking them in tabbed columns, the latter being the method being used in the original dump.

The Syntax used below follows WordPerfect's recommendations. In some cases, I would do things differently, e.g., for statements such as: **Var := Whatever** (). As I see it, the colon is always unnecessary and, if no parameters are associated with (or are required by) a command, the paired parentheses () are, too, and just adds clutter. In such cases, **Var = Whatever** works just as well.

Understand that, for commands containing an equation or comparison statement, the 1<sup>st</sup> element of the statement shown here represents the type of value returned to a variable you would name, e.g., **Boolean := ?ActivateHypertext** () means: **Var = ?ActivateHypertext**, and the variable "Var" will have a boolean value (in this example, True or False), True if Hypertext is activated, False if it is not.

[Go To Bottom of Document](#)

## List of 2175 System & Product Commands for WordPerfect 10 and List of 473 PerfectScript Commands for WordPerfect 10

Note: Within the lists, click on any link to return to its index. Click on an Index Title to return here.

### [- ? System Variables -](#)

[A](#) [B](#) [C](#) [D](#) [E](#) [F](#) [G](#) [H](#) [I](#) [J](#) [K](#) [L](#) [M](#) [N](#) [O](#) [P](#) [Q](#) [R](#) [S](#) [T](#) [U](#) [V](#) [W](#) [Z](#)

### [- WordPerfect Product Commands -](#)

[A](#) [B](#) [C](#) [D](#) [E](#) [F](#) [G](#) [H](#) [I](#) [J](#) [K](#) [L](#) [M](#) [N](#) [O](#) [P](#) [Q](#) [R](#) [S](#) [T](#) [U](#) [V](#) [W](#) [Z](#)

### [- WordPerfect PerfectScript Commands -](#)

[Misc \\* / - + <](#) [A](#) [B](#) [C](#) [D](#) [E](#) [F](#) [G](#) [I](#) [L](#) [M](#) [N](#) [O](#) [P](#) [Q](#) [R](#) [S](#) [T](#) [U](#) [V](#) [W](#) [X](#)  
[> % = etc](#)

### [- WORDPERFECT 10 SYSTEM VARIABLES -](#)

#### [-A-](#)

**Boolean := ?ActivateHypertext** ()

[A value representing whether Activate Hypertext is selected]

**Numeric := ?AddressBookSelection** ()

[The number of Address Book entries most recently selected]

**Numeric := ?Align** ()

[A value that represents the current character justification]

**Boolean := ?AutoUpdate** ()

[A value representing whether Update on retrieve is on or off for spreadsheet/database links]

**Boolean := ?AutoWordSelect** ()

[A value indicating whether Automatically select words is on or off]

#### [-B-](#)

**Boolean := ?BaselineMode** ()

[A value representing whether Baseline placement for typesetting is on or off]

**Boolean := ?BetweenTableCodes** ()

[A value representing whether the insertion point is in a table]

**Numeric := ?Binding** ()

[A value representing the binding edge for the current page]

**Boolean := ?BlockActive** ()

[A value representing whether Block is on]

**String := ?BookmarkList** ()

[The name of the earliest-created bookmark that still exists in the current document]

**Any := ?Border ()**  
[A value representing the current border style]

**Any := ?BorderBottomLine ()**  
[A value representing the bottom line style of a border]

**Numeric := ?BorderColorBlue ()**  
[A value representing the amount of blue in a border color]

**Numeric := ?BorderColorGreen ()**  
[A value representing the amount of green in a border color]

**Numeric := ?BorderColorRed ()**  
[A value representing the amount of red in a border color]

**Numeric := ?BorderColorShading ()**  
[A value representing the shading percentage of a border color]

**Numeric := ?BorderCornerRadius ()**  
[A measurement expression representing the corner radius of a border]

**Any := ?BorderInsideSpacingBottom ()**  
[A measurement expression representing the bottom spacing inside a border]

**Any := ?BorderInsideSpacingTop ()**  
[A measurement expression representing the top spacing inside a border]

**Any := ?BorderLeftLine ()**  
[A value representing the left line style of a border]

**Any := ?BorderRightLine ()**  
[A value representing the right line style of a border]

**Any := ?BorderSeparatorLine ()**  
[A value representing the separator line style of a border]

**Any := ?BorderSetSpacing ()**  
[A value representing whether Auto Spacing is on or off for the current border]

**Any := ?BorderTopLine ()**  
[A value representing the top line style of a border]

**Any := ?BorderUseBorderColor ()**  
[A value representing whether the border color option is used for a border]

**Any := ?BoxAttachTo ()**  
[A value representing whether a graphics box is attached to a page, a paragraph, or a character]

**Any := ?BoxCaptionDefaultInitialStyle ()**  
[A value representing the initial style of the current box caption]

**Any := ?BoxCaptionDefaultNumberStyle ()**  
[A value representing the current box caption number style]

**Any := ?BoxCaptionFormatting ()**  
[A value representing the width of a graphics box caption]

**Any := ?BoxCaptionPositionAlignment ()**  
[A value representing the alignment of a graphics box caption]

**Any := ?BoxCaptionPositionBorder ()**  
[A value representing the position of a graphics box caption relative to the border]

**Any := ?BoxCaptionPositionOffset ()**  
[A value representing the offset position of a graphics box caption]

**Any := ?BoxCaptionPositionSide ()**  
[A value representing the side of a box where a caption is located]

**Any := ?BoxCaptionRotation ()**  
[A value representing the rotation of a box caption]

**Any := ?BoxChangeLineHeight ()**  
[A value showing whether character-positioned graphics boxes affect line height]

**Any := ?BoxContentFilename ()**  
[The name of a text file or image file contained in a graphics box]

**Any := ?BoxContentHorizontalPosition ()**  
[A value representing the horizontal position of graphics box contents]

**Any := ?BoxContentInternal ()**  
[A value representing whether an image is stored in the current document or on a disk]

**Any := ?BoxContentPreserveAspectRatio ()**  
[A value representing whether the image width/height ratio is preserved when the position of a box changes]

**Any := ?BoxContentType ()**  
[A value representing the type of content in a graphics box]

**Any := ?BoxContentVerticalPosition ()**  
[A value representing the vertical position of graphics box content]

**Any := ?BoxCount ()**  
[The number of boxes in the current document]

**Any := ?BoxCounter ()**  
[A value representing the counter type of a graphics box or graphics box style]

**Any := ?BoxEquationColorBlue ()**  
[A value representing whether an equation color contains blue]

**Any := ?BoxEquationColorGreen ()**  
[A value representing whether an equation color contains green]

**Any := ?BoxEquationColorRed ()**  
[A value representing whether an equation color contains red]

**Any := ?BoxEquationColorShading ()**  
[A value representing the shade of an equation color]

**Any := ?BoxEquationFont ()**  
[The name of the font used in the current equation box]

**Any := ?BoxEquationFontSize ()**  
[A value representing the font size used in the current equation box]

**Any := ?BoxHeight ()**  
[A value representing the height of a graphics box]

**Any := ?BoxHorizontalAlignment ()**  
[A value representing the horizontal position of a graphics box attached to a page or a paragraph]

**Any := ?BoxHorizontalAlignTo ()**  
[A value representing whether a graphics box attached to a page is aligned to the margins or to columns]

**Any := ?BoxHorizontalOffset ()**  
[A value representing the right or left position of a graphics box relative to an alignment point]

**Any := ?BoxImageBlackWhiteThreshold ()**  
[A value representing a threshold where colors and/or grays convert to black or white]

**Any := ?BoxImageBrightness ()**  
[A value representing the brightness of an image color]

**Any := ?BoxImageContrast ()**  
[A value representing image contrast]

**Any := ?BoxImageDitherMethod ()**  
[A value representing a dither method]

**Any := ?BoxImageDitherSource ()**  
[A value representing a dither source]

**Any := ?BoxImageFill ()**  
[A value representing an image fill type]

**Any := ?BoxImageFlipX ()**  
[A value representing whether the image is reversed in its horizontal axis]

**Any := ?BoxImageFlipY ()**  
[A value representing whether the image is reversed on its vertical axis]

**Any := ?BoxImageHalftoneAngle0 ()**  
[The image halftone angle]

**Any := ?BoxImageHalftoneAngle1 ()**  
[The image halftone angle 1]

**Any := ?BoxImageHalftoneAngle2 ()**  
[The image halftone angle 2]

**Any := ?BoxImageHalftoneAngle3 ()**  
[The image halftone angle 3]

**Any := ?BoxImageHalftoneScreenLPI ()**  
[The halftone lines per inch (LPI)]

**Any := ?BoxImageInvertColors ()**  
[A value representing whether image colors have been inverted to their respective complementary colors]

**Any := ?BoxImageMonochrome ()**  
[A value representing whether a color image converts to black and white]

**Any := ?BoxImagePageBackground ()**  
[A value representing whether any background colors or gradients saved with an image are visible]

**Any := ?BoxImageRotation ()**  
[A value representing the degree of rotation for a graphics box]

**Any := ?BoxImageScalingX ()**  
[A value representing a scaling width]

**Any := ?BoxImageScalingY ()**  
[A value representing a scaling height]

**Any := ?BoxImageTranslationX ()**  
[A value representing the horizontal position of an image within a box]

**Any := ?BoxImageTranslationY ()**  
[A value representing the vertical position of an image within a box]

**Any := ?BoxLeftColumn ()**  
[A value representing the left column to which a graphics box aligns]

**Any := ?BoxNumber ()**  
[The number of the current graphics box]

**Any := ?BoxOverlap ()**  
[A value representing whether a graphics box may overlap other boxes]

**Any := ?BoxRightColumn ()**  
[A value representing the right column to which a graphics box aligns]

**Boolean := ?BoxStaysOnPage ()**  
[A value representing whether a box stays on the page to which it was attached or moves with text]

**Boolean := ?BoxSizelsNotLimited ()**  
[A value representing if a box is limited by table boundaries]

**Any := ?BoxStyleName ()**  
[The name of a box style]

**Boolean := ?BoxSuppressed ()**  
[A value representing if a box is suppressed]

**Any := ?BoxTextAngle ()**  
[A value representing the degree of rotation for text in a box]

**Any := ?BoxTextDefaultStyle ()**  
[The default style of box text]

**Any := ?BoxVerticalAlignment ()**  
[A value representing the vertical position of a graphics box attached to a page or character]

**Any := ?BoxVerticalAlignTo ()**  
[A value representing whether a box is aligned to the margins]

**Any := ?BoxVerticalOffset ()**  
[A value representing the vertical position of a graphics box relative to an alignment point]

**Any := ?BoxWidth ()**  
[A value representing the width of a graphics box]

**Any := ?BoxWrapping ()**  
[A value representing the direction of text flow around a graphics box]

**Any := ?BoxWrappingType ()**  
[A value representing whether the text outside a graphics box flows around an image]

**Boolean := ?ButtonBarActive ()**  
[A value representing whether the toolbar is active in the current document]

**- C -**

**Boolean := ?CapBeginSentence ()**  
[A value representing whether the QuickCorrect Capitalize first letter option is on]

**String := ?Cell ()**  
[A value representing the current table cell]

**Numeric := ?CellAlignment ()**  
[A value representing the vertical alignment of the current cell]

**Boolean := ?CellBold ()**  
[A value indicating whether the cell font attribute is Bold]

**Boolean := ?CellDoubleUnderline ()**  
[A value indicating whether the cell font attribute is Double Underline]

**Boolean := ?CellExtraLarge ()**  
[A value indicating whether the cell font attribute is Extra Large]

**Any := ?CellFill ()**  
[A value representing the current cell's fill style]

**Boolean := ?CellFine ()**  
[A value indicating whether the cell font attribute is Fine]

**Boolean := ?CellFormula ()**

[A value indicating whether the cell contains a formula]

**Boolean := ?CellIgnoreWhenCalculating ()**

[A value indicating whether the option to ignore the current cell when calculating formulas is active]

**Boolean := ?CellItalics ()**

[A value indicating whether the cell font attribute is Italics]

**Numeric := ?CellJustification ()**

[A value representing the justification of the current cell]

**Boolean := ?CellLarge ()**

[A value indicating whether the cell font attribute is Large]

**Boolean := ?CellLocked ()**

[A value indicating whether the current cell is locked]

**Boolean := ?CellNormal ()**

[A value indicating whether the cell font attribute is Normal]

**Boolean := ?CellOutline ()**

[A value indicating whether the cell font attribute is Outline]

**Boolean := ?CellRedline ()**

[A value indicating whether the cell font attribute is Redline]

**Boolean := ?CellShadow ()**

[A value indicating whether the cell font attribute is Shadow]

**Boolean := ?CellSmall ()**

[A value indicating whether the cell font attribute is Small]

**Boolean := ?CellSmallCaps ()**

[A value indicating whether the cell font attribute is Small Caps]

**Numeric := ?CellSpecificAttribute ()**

[A value indicating whether the current cell contains cell-specific attributes]

**Numeric := ?CellSpecificJustification ()**

[A value representing whether cell justification is active in the current cell]

**Boolean := ?CellStrikeout ()**

[A value indicating whether the cell font attribute is Strikeout]

**Boolean := ?CellSubscript ()**

[A value indicating whether the cell font attribute is Subscript]

**Boolean := ?CellSuperscript ()**

[A value indicating whether the cell font attribute is Superscript]

**Boolean := ?CellUnderline ()**

[A value indicating whether the cell font attribute is Underline]

**Boolean := ?CellVeryLarge ()**

[A value indicating whether the cell font attribute is Very Large]

**Numeric Measurement := ?CellWidth ()**

[A value indicating the width of the current table cell, in WordPerfect units (1200ths of an inch)]

**Boolean := ?CenterCurrentPage ()**

[A value indicating whether the contents of the current page only are set to be centered vertically]

**Boolean := ?CenterPages ()**

[A value indicating vertical centering by means of the Current and Subsequent Pages setting]

**Numeric := ?ChapterNumber ()**

[The current chapter number]

**Numeric := ?ChapterNumberMethod ()**

[A value representing the current chapter numbering method]

**Numeric := ?CharSpaceWidth ()**

[A value representing the amount of space between letters]

**Boolean := ?CleanupSpaces ()**

[A value indicating whether Cut and Paste Cleanup is on or off]

**Numeric := ?Column ()**

[The number of the current text column or table column]

**Any := ?ColumnBorder ()**

[A value representing whether the current column setting includes a border]

**Any := ?ColumnDefine ()**

[A value representing the current column type]

**Any := ?ColumnWidth ()**

[A value indicating the width of the current column in WordPerfect units (1200ths of an inch).]

**Boolean := ?ConfirmCodesDelete ()**

[A value indicating whether Confirm Deletion of Codes, Stop Insertion Point at Hidden Codes is on or off]

**Boolean := ?ConfirmFormulaDelete ()**

[A value indicating whether Confirm Deletion of Table Formulas is on or off]

**String := ?CurrentKeyboard ()**

[The name of the active keyboard]

**String := ?CurrentLabel ()**

[The name of the current label selection]

**String := ?CurrentMenu ()**

[The name of the active menu bar]

**String := ?CurrentPaperName ()**

[The page definition currently selected in the Page Size dialog box]

**Numeric := ?CurrentParagraphLevel ()**

[The level of the active outline paragraph]

**String := ?CurrentPrinterPort ()**

[Returns the name of the active printer port]

**String := ?CurrentStyle ()**

[The name of the active style]

**Numeric := ?CurrentSubdoc ()**

[A value representing which substructure is active]

**String := ?CurrentTableName ()**

[The active table name, or the previous table name if the insertion point is not in a table]

**String := ?CurrentTemplate ()**

[The path and filename of the active template]

**String := ?CurrentWord ()**

[The word at the insertion point]

**- D -**

**Numeric := ?DateDay ()**

[A numeric value representing the current day of the month]

**Numeric := ?DateMonth ()**

[A numeric value representing the current month]

**String := ?DateWeekday ()**  
[The current day of the week]

**Numeric := ?DateYear ()**  
[A numeric value representing the current year]

**String := ?DefaultDocumentExtension ()**  
[The default extension for document filenames]

**String := ?DefaultTemplateExtension ()**  
[The default extension for template filenames]

**Numeric := ?DefaultView ()**  
[A value representing the default view preference]

**Numeric := ?DisplayMode ()**  
[A value representing the current display mode]

**Boolean := ?DocBlank ()**  
[A value representing whether the current document is blank]

**Boolean := ?DocChanged ()**  
[A value representing whether the document has been modified since it was last saved]

**Boolean := ?DocModified ()**  
[A value indicating whether the active document has been modified since the last time it was saved]

**Boolean := ?DocNeedsGenerating ()**  
[A value indicating whether the active document needs to be generated]

**Numeric := ?DocNumber ()**  
[The active document number or substructure number]

**Boolean := ?DocReadOnly ()**  
[A value representing whether the current document is read-only]

**String := ?DocumentFont ()**  
[The name of the document default font]

**Numeric := ?DocumentInformation ()**  
[The number of characters in the active document]

**String := ?DOSName ()**  
[The active DOS filename]

**String := ?DOSPath ()**  
[The active DOS file path, not including the filename]

**String := ?DotLeader ()**  
[The active dot leader character]

**Boolean := ?DoubleSmartQuote ()**  
[A value representing whether QuickCorrect changes double quotes to double smart quotes]

**Any := ?DropCapBorderStyle ()**  
[The active drop cap border style]

**Numeric := ?DropCapCharacters ()**  
[The number of characters used in a drop cap]

**Numeric := ?DropCapCharacterSize ()**  
[The active drop cap character size relative to a line of text]

**Boolean := ?DropCapContourWrap ()**  
[A value indicating whether Wrap Text around Drop Cap is on or off]

**Any := ?DropCapFillStyle ()**  
[The active drop cap fill style]

**Numeric := ?DropCapHorizontalAdjustment ()**  
[A percentage indicating how far into the left margin to move a drop cap]

**Boolean := ?DropCapIncludeDescender ()**  
[A value indicating whether Allow for Descender is on in Drop Cap Options]

**Boolean := ?DropCapIncludeDiacritic ()**  
[A value indicating whether Allow for Diacritic is selected in Drop Cap Options]

**Boolean := ?DropCapInParagraph ()**  
[A value indicating whether the active paragraph contains a drop cap definition]

**Numeric Measurement := ?DropCapKerning ()**  
[An expression in WordPerfect units indicating text placement in relation to a drop cap]

**Boolean := ?DropCapNoWrap ()**  
[A value indicating whether Wrap Text around Drop Cap is on or off]

**Boolean := ?DropCapState ()**  
[A value indicating whether the Drop Cap feature bar is on or off]

**Boolean := ?DropCapUseColor ()**  
[A value indicating whether a drop cap uses a color different from that of the surrounding text]

**Numeric := ?DropCapVerticalAdjustment ()**  
[The number of lines down to move a drop cap]

**- E -**

**Boolean := ?EditScreen ()**  
[A value representing whether the current screen is an editing screen]

**Numeric := ?Endnote ()**  
[The number of the endnote to the left of the insertion point]

**Numeric := ?EndOfSentenceSpacing ()**  
[Indicate what QuickCorrect will do with spaces at the end of sentences]

**- F -**

**Numeric := ?FeatureBar ()**  
[A value representing which feature bar is displayed]

**Any := ?Fill ()**  
[A value representing the active fill style]

**Any := ?FillGradientSteps ()**  
[A value representing whether steps in a gradient are calculated automatically]

**Any := ?FillGradientType ()**  
[A value representing whether the active gradient fill is linear, radial, or rectangular]

**Any := ?FillType ()**  
[A value representing whether the fill is a pattern or a gradient]

**Any := ?FillWPG2Pattern ()**  
[A value representing a predefined fill pattern]

**Numeric Measurement := ?FirstLineIndent ()**  
[An expression indicating the amount the first line of a new paragraph is indented]

**Numeric := ?FirstPrintLineNum ()**  
[The first line number to print]

**String := ?Font ()**  
[The name of the active font]

**Boolean := ?FontBold ()**  
[A value representing whether the active font attribute is Bold]

**Boolean := ?FontDoubleUnderline ()**  
[A value representing whether the active font attribute is Double Underline]

**Boolean := ?FontExtraLarge ()**  
[A value representing whether the active font attribute is Extra Large]

**Boolean := ?FontFine ()**  
[A value representing whether the active font attribute is Fine]

**Boolean := ?FontItalics ()**  
[A value representing whether the active font attribute is Italics]

**Boolean := ?FontLarge ()**  
[A value representing whether the active font attribute is Large]

**Boolean := ?FontOutline ()**  
[A value representing whether the active font attribute is Outline]

**Boolean := ?FontRedline ()**  
[A value representing whether the current font attribute is Redline]

**Boolean := ?FontShadow ()**  
[A value representing whether the active font attribute is Shadow]

**Numeric Measurement := ?FontSize ()**  
[A value representing the active font size]

**Boolean := ?FontSmall ()**  
[A value representing whether the active font attribute is Small]

**Boolean := ?FontSmallCaps ()**  
[A value representing whether the active font attribute is Small Caps]

**Boolean := ?FontStrikeout ()**  
[A value representing whether the active font attribute is Strikeout]

**Boolean := ?FontSubscript ()**  
[A value representing whether the active font attribute is Subscript]

**Boolean := ?FontSuperscript ()**  
[A value representing whether the active font attribute is Superscript]

**Boolean := ?FontUnderline ()**  
[A value representing whether the active font attribute is Underline]

**Boolean := ?FontVeryLarge ()**  
[A value representing whether the active font attribute is Very Large]

**Boolean := ?FooterAlnParagraph ()**  
[Indicates whether a Footer A code is in the active paragraph]

**Numeric := ?FooterAOccur ()**  
[Indicates whether Footer A prints on odd, even, or all pages]

**Boolean := ?FooterBInParagraph ()**  
[Indicates whether a Footer B code is in the active paragraph]

**Numeric := ?FooterBOccur ()**  
[Indicates whether Footer B prints on odd, even, or all pages]

**Any := ?FooterSeparator ()**  
[An expression indicating the amount of blank space above a footer]

**Numeric := ?Footnote ()**  
[The footnote number to the left of the insertion point]

**Boolean := ?FootnoteContinued ()**  
[A value that determines whether to print a continued message when a footnote is split between two pages]

**Numeric := ?FootnotePosition ()**  
[A value representing where footnotes are printed: at the bottom of the page, or below the last line of text]

**Boolean := ?FootnoteRestartsPage ()**  
[A value representing whether footnote numbering restarts on every page]

**Numeric Measurement := ?FootnoteSpace ()**  
[An expression indicating the amount of blank space between footnotes]

**Boolean := ?FormatForDefaultPrinter ()**  
[A value indicating whether Reformat documents for the WordPerfect 10 default printer on open is selected]

**- G -**

**Numeric := ?GetSelEndEx ()**  
[A value indicating the end position of the selection. Used with SetSelEx]

**Numeric := ?GetSelSizeEx ()**  
[A value indicating the size of the selection set with SetSelEx]

**Numeric := ?GetSelStartEx ()**  
[A value indicating the start position of the selection. Used with SetSelEx]

**String := ?GetSelTextEx ()**  
[The data contained within a selection set with SetSelEx]

**String := ?GetSelTextAllEx ()**  
[The data contained within the entire document as if set with SetSelEx]

**Boolean := ?GraphicSelected ()**  
[A value indicating whether a graphic is selected]

**Numeric Measurement := ?GraphicsLineBottomSpacing ()**  
[A value representing the amount of space below a graphics line]

**Numeric := ?GraphicsLineColorBlue ()**  
[A value representing the amount of blue in a graphics line color]

**Numeric := ?GraphicsLineColorGreen ()**  
[A value representing the amount of green in a graphics line color]

**Numeric := ?GraphicsLineColorRed ()**  
[A value representing the amount of red in a graphics line color]

**Numeric := ?GraphicsLineColorShading ()**  
[A value representing the shade of a graphics line color]

**Numeric := ?GraphicsLineCount ()**  
[The number of graphics lines in the current document]

**Numeric := ?GraphicsLineHorizontalAlign ()**  
[A value representing the horizontal alignment of a graphics line]

**Numeric Measurement := ?GraphicsLineHorizontalColumn ()**  
[The column number]

**Numeric Measurement := ?GraphicsLineHorizontalPosition ()**  
[A value representing the horizontal position of a graphics line]

**Numeric Measurement := ?GraphicsLineLength ()**  
[The length of a graphics line]

**Numeric := ?GraphicsLineNum ()**  
[The current graphics line number]

**Any := ?GraphicsLineStyle ()**  
[A value representing a graphics line style]

**Any := ?GraphicsLineThickness ()**  
[A value representing the thickness of a graphics line]

**Numeric Measurement := ?GraphicsLineTopSpacing ()**  
[A value representing the amount of space above a graphics line]

**Numeric := ?GraphicsLineType ()**  
[A value representing whether a graphics line is horizontal or vertical]

**Boolean := ?GraphicsLineUseColor ()**  
[A value representing whether to use the current line style color or to choose another color]

**Numeric := ?GraphicsLineVerticalAlign ()**  
[A value representing the vertical alignment of a graphics line]

**Numeric Measurement := ?GraphicsLineVerticalPosition ()**  
[A value representing the vertical position of a graphics line]

- H -

**Boolean := ?HeaderAInParagraph ()**  
[Indicates whether a Header A code is in the active paragraph]

**Numeric := ?HeaderAOccur ()**  
[Indicates whether Header A prints on odd, even, or all pages]

**Boolean := ?HeaderBInParagraph ()**  
[Indicates whether a Header B code is in the active paragraph]

**Numeric := ?HeaderBOccur ()**  
[Indicates whether Header B prints on odd, even, or all pages]

**Any := ?HeaderSeparator ()**  
[An expression indicating the amount of blank space above a header]

**Boolean := ?HiddenTextActive ()**  
[A value indicating whether hidden text will be displayed]

**String := ?HypertextLast ()**  
[The name of the file in which a hypertext link originated]

**Numeric := ?HypertextType ()**  
[A value representing whether a hypertext link jumps to a bookmark or plays a macro]

- I -

**Numeric := ?Indent ()**  
[A value representing whether the first line of a new paragraph is indented]

**String := ?IndexDefinition ()**  
[A value representing the current index heading style]

**String := ?Initials ()**  
[The network initials of the current user]

**Boolean := ?InOutline ()**  
[A value representing whether the insertion point is in an outline]

**Boolean := ?InTable ()**  
[A value indicating whether the insertion point is in a table]

**Numeric := ?InterimVersion ()**  
[The interim release number of the current version of WordPerfect 10]

**Boolean := ?IrregularCap ()**  
[A value representing whether QuickCorrect checks for irregular capitalization]

- J -

**Numeric := ?Justification ()**  
[A value representing the current justification]

- K -

**Boolean := ?Kerning ()**  
[A value representing whether Automatic Kerning is on or off]

**Boolean := ?KeyCapsLock ()**  
[A value representing whether Caps Lock is on or off]

- L -

**Numeric := ?LabelsToDisplay ()**  
[A value representing whether to display labels for laser or tractor-feed printers, or both types]

**String := ?Language ()**  
[A value representing the active language]

**String := ?LeftChar ()**  
[The character immediately to the left of the insertion point]

**Numeric := ?LeftCode ()**  
[A value representing the code immediately to the left of the insertion point]

**Numeric := ?LeftHyphZone ()**  
[A value representing the lower limit of a hyphenation zone]

**Numeric Measurement := ?Line ()**  
[A value representing the vertical position of the insertion point relative to the top of the page]

**Numeric := ?LineHeight ()**  
[A value representing whether line height is fixed or set automatically]

**Numeric := ?LineNumber ()**  
[The starting line number for the active document]

**Boolean := ?LineNumbering ()**  
[A value representing whether Line Numbering is on or off]

**Numeric := ?LineNumberingInterval ()**  
[A number indicating the interval between numbered lines]

**Numeric := ?LineNumberMethod ()**  
[A value representing the line numbering method]

**Boolean := ?LineNumCountBlankLines ()**  
[A value representing whether Line Numbering counts blank lines]

**Boolean := ?LineNumNewspaperCols ()**  
[A value representing whether Line Numbering numbers newspaper columns]

**Numeric Measurement := ?LineNumPosition ()**  
[The line numbering position]

**Numeric := ?LineSegmentBlue ()**  
[A value representing the amount of blue in a table line segment color]

**Numeric := ?LineSegmentCount ()**  
[The number of line segments in a line style]

**Numeric := ?LineSegmentGreen ()**  
[A value representing the amount of green in a line segment color]

**Numeric := ?LineStyleRed ()**

[A value representing the amount of red in a line segment color]

**Raw Binary Data := ?LineStyleSetUserData ()**

[An expression indicating the dash length in the active custom line pattern]

**Any := ?LineStyleShading ()**

[A value representing the shade of a line segment color]

**Any := ?LineStyleSpacing ()**

[A value representing the space between one stacked line segment and the next]

**Any := ?LineStyleWidth ()**

[The thickness of a line segment]

**Any := ?LineStyleWavy ()**

[The line segment is wavy]

**Any := ?LineStyleWavyWidth ()**

[The thickness of a wavy line segment]

**Any := ?LineStyleWPG2Pattern ()**

[A value representing a predefined line segment pattern]

**Numeric := ?LineStyle ()**

[An expression indicating the amount of space between lines]

**String := ?ListDefinition ()**

[The active list name]

- M -

**Boolean := ?MacroExecute ()**

[A value representing whether a macro is playing]

**Boolean := ?MacroPlaying ()**

[A value representing whether a macro is playing]

**Boolean := ?MacroRecord ()**

[A value representing whether a macro is recording]

**Numeric := ?MajorVersion ()**

[The major version number of the WordPerfect 10 program you are using]

**Boolean := ?MergeDone ()**

[A value representing whether a merge, executed while the macro is running, has finished]

**Numeric := ?MinorVersion ()**

[The minor version number of the WordPerfect 10 program you are using]

- N -

**String := ?Name ()**

[The active filename]

**Boolean := ?Network ()**

[A value representing whether WordPerfect 10 is running on a network]

**Boolean := ?NotFound ()**

[A value representing whether the object of the last search was not found]

**Numeric := ?NumberOpenDocuments ()**

[A value indicating the number of open documents]

- O -

**String := ?ODMAName ()**

[The active ODMA filename]

**String := ?ODMAPath ()**

[The active ODMA Document ID (not including the filename)]

**String := ?Overstrike ()**

[The active overstrike characters]

- P -

**Numeric := ?Page ()**

[The active page number]

**Any := ?PageBorder ()**

[A value representing whether the active page has a border]

**String := ?PageNumberFormat ()**

[The text included in the active page numbering format]

**Boolean := ?PageNumberingOn ()**

[A value representing whether page numbering is on in the current document]

**Any := ?PageNumberingValue ()**

[The new page number specified in Page Settings in the Numbering Value dialog box]

**Numeric := ?PageNumberMethod ()**

[A value representing the active page numbering method]

**Numeric := ?PageNumberPosition ()**

[A value representing the active page number position]

**Numeric := ?PageSuppress ()**

[A value representing the items to suppress, as set in the Suppress dialog box]

**Numeric Measurement := ?PaperLength ()**

[An expression indicating the current paper length in WordPerfect units of measure]

**Numeric := ?PaperSizeType ()**

[A value representing the paper type selected in the Paper Size dialog box]

**Numeric Measurement := ?PaperWidth ()**

[An expression indicating the current paper width in WordPerfect units of measure]

**Any := ?ParagraphBorder ()**

[A value representing whether the current paragraph has a border]

**Any := ?ParagraphNumber ()**

[A value indicating the paragraph number if the paragraph outline style is active]

**Any := ?ParagraphNumberMethod ()**

[The paragraph numbering method]

**String := ?Path ()**

[The current file path (does not include the filename)]

**String := ?PathBackup ()**

[The backup folder specified in File Settings]

**String := ?PathCurrent ()**

[The current folder]

**String := ?PathDatabase ()**

[The folder specified for database files in the File Settings dialog box]



**String := ?PathDatabaseSupplemental ()**

[The folder specified for supplemental database files in the File Settings dialog box]

**String := ?PathDocument ()**

[The folder specified for documents in the File Settings dialog box]

**String := ?PathGraphics ()**

[The folder specified for graphics files in the File Settings dialog box]

**String := ?PathGraphicsSupplemental ()**

[The folder specified for supplemental graphics files in the File Settings dialog box]

**String := ?PathLabels ()**

[The folder and filename specified for labels files in the File Settings dialog box]

**String := ?PathMacros ()**

[The folder specified for macros files in the File Settings dialog box]

**String := ?PathMacrosSupplemental ()**

[The folder specified for supplemental macros in the File Settings dialog box]

**String := ?PathPrinters ()**

[The directory specified for label files in the File Settings dialog box]

**String := ?PathProgram ()**

[The WordPerfect 10 program folder (where WPWIN10.EXE is located)]

**String := ?PathShared ()**

[The folder where PerfectFit files are located]

**String := ?PathSpreadsheet ()**

[The folder specified for spreadsheet files in the File Settings dialog box]

**String := ?PathSpreadsheetSupplemental ()**

[The folder specified for supplemental spreadsheet files in the File Settings dialog box]

**String := ?PathTemplate ()**

[The folder specified for template files in the File Settings dialog box]

**String := ?PathTemplateSupplemental ()**

[The folder specified for supplemental template files in the File Settings dialog box]

**String := ?PauseKey ()**

[A value representing the keystroke specified to resume a paused macro]

**Numeric Measurement := ?Position ()**

[A value representing the horizontal position of the insertion point]

**Boolean := ?PowerBarActive ()**

[A value indicating whether the property bar is active in the current document]

**- Q -**

**Boolean := ?QuickCorrect ()**

[A value representing whether QuickCorrect is on or off]

**- R -**

**Numeric := ?Random ()**

[A random number between 0 and 32767]

**Numeric := ?RedlineCharacter ()**

[A value representing the character used to mark redline text]

**Numeric := ?RedlineMethod ()**

[A value representing a redline method for the current document]

**Boolean := ?RevealCodesActive ()**

[A value representing whether Reveal Codes is active]

**String := ?RightChar ()**

[The character at the insertion point]

**Numeric := ?RightCode ()**

[A value representing the code at the insertion point]

**Numeric := ?RightHyphZone ()**

[A value representing the upper limit of a hyphenation zone]

**Numeric := ?Row ()**

[The current row number in a table]

**Numeric := ?RowHeader ()**

[A value representing whether the current table row is a header row]

**Numeric Measurement := ?RowHeight ()**

[An expression in WordPerfect units of measure indicating the current table row height]

**Boolean := ?RulerBarActive ()**

[A value representing whether the Ruler is active in the current document]

**- S -**

**Boolean := ?SaveDocumentSummaryOnExit ()**

[True if Create Summary on Save/Exit is active, False if not]

**Numeric := ?SaveWorkspace ()**

[A value representing the current Save Workspace environment preference setting]

**Numeric := ?SecondaryPageNumber ()**

[The current secondary page number]

**Numeric := ?SecondaryPageNumberMethod ()**

[A value representing the secondary page numbering method for the current document]

**String := ?SelectedText ()**

[The currently selected text]

**Numeric := ?SelectedTextSize ()**

[A value representing the number of bytes in selected text]

**Boolean := ?SetQuickmarkOnSave ()**

[A value indicating whether or not Set Quickmark on File Save is selected in the Bookmark dialog box]

**Boolean := ?SGMLActive ()**

[A value representing whether the current document is an XML document]

**Numeric := ?SGMLAttributePrompt ()**

[A value representing how Attribute Prompt is set]

**Boolean := ?SGMLChaining ()**

[A value representing whether chaining is turned on or off]

**String := ?SGMLCurrentLGC ()**

[The name of the .LGC file currently being used]

**String := ?SGMLCurrentLSI ()**

[The name of the .LSI file currently being used]

**String := ?SGMLCurrentRootElement ()**

[The name of the element that is currently the root]

**Numeric := ?SGMLDataTypeAllowed ()**  
[A value representing the type of data allowed (if any) at the current context]

**Numeric := ?SGMLDisplayCodes ()**  
[A value representing how Display Codes is set]

**Boolean := ?SGMLDisplayInclusions ()**  
[A value representing whether Display Inclusions is set]

**String := ?SGMLLGCPath ()**  
[The current .LGC file path (does not include the filename)]

**Boolean := ?ShowBoxes ()**  
[A value determining whether Display Settings hide or show graphics]

**Boolean := ?SingleSmartQuote ()**  
[A value representing whether QuickCorrect changes single quotes to single smart quotes]

**Boolean := ?SingleSpaceInSentence ()**  
[A value representing whether QuickCorrect changes multiple spaces within sentences to single spaces]

**Any := ?SortGetAction ()**  
[A value representing whether Select Without Sorting is on or off]

**Any := ?SortGetCase ()**  
[A value representing whether the Uppercase First sort option is on or off]

**Any := ?SortGetUndo ()**  
[A value indicating whether the Allow Undo sort option is on or off]

**Numeric := ?SpaceExpansion ()**  
[An expression representing the current compression limit for word spacing justification]

**Numeric := ?SpacingBelowPar ()**  
[An expression indicating the amount of blank space between paragraphs]

**Numeric := ?StartingLineNumber ()**  
[The first number to use when numbering lines of text]

**Boolean := ?StatusBarActive ()**  
[A value representing whether the status bar is active in the current document]

**Boolean := ?Substructure ()**  
[A value representing whether a substructure is active]

- T -

**Numeric := ?TableAlternateFill ()**  
[The state of Alternating Fill for the current table]

**Any := ?TableDefaultLine ()**  
[A value representing the default line style of the current table]

**Boolean := ?TableGuides ()**  
[A value representing whether the Table Cell Guides display preference is on or off]

**Boolean := ?TableInTable ()**  
[A value representing whether the insertion point is in a table]

**Boolean := ?TableLastCell ()**  
[A value representing whether the current cell is the last cell in a table]

**Any := ?TableSecondFillStyle ()**  
[The secondary fill style for the current table]

**Any := ?TableSkewedColumnAngle ()**  
[The column skew angle of this table]

**Any := ?TableSkewedColumn ()**  
[A value representing if the first or last column is skewed]

**Boolean := ?TableSkewedColumnText ()**  
[A value representing if the column text is skewed]

**Any := ?TableSkewedRowAngle ()**  
[The row skew angle of this table]

**Boolean := ?TableSkewedRowText ()**  
[A value representing if the row text is skewed]

**Any := ?TableSkewedSquareRow ()**  
[A value representing if first row is square]

**Boolean := ?TableSkewedJoinedCorners ()**  
[A value representing if the skewed table has joined corners]

**Any := ?TableStyle ()**  
[The current table style]

**Numeric := ?TabSettings ()**  
[A value representing whether tabs are positioned from the left edge of the paper or relative to the left margin]

**String := ?TemplateFile ()**  
[The path and filename of the default template]

**String := ?TemplateFileSupplemental ()**  
[The path and filename of the current supplemental template]

**Boolean := ?TemplateFillCanceled ()**  
[A value representing whether the last Template Fill was cancelled]

**Numeric := ?TextColorBlue ()**  
[A value representing the amount of blue in the current text color]

**Numeric := ?TextColorGreen ()**  
[A value representing the amount of green in the current text color]

**Numeric := ?TextColorRed ()**  
[A value representing the amount of red in the current text color]

**Numeric := ?TextColumnNumber ()**  
[The number of the text column containing the insertion point]

**Numeric := ?TextColumns ()**  
[The number of text columns in the current document]

**Numeric := ?TextShade ()**  
[A value representing the shade of the current text color]

**Numeric := ?TimeHour ()**  
[A value representing the current hour on a 24-hour clock]

**Numeric := ?TimeHundredth ()**  
[A value representing the current hundredth of a second]

**Numeric := ?TimeMinute ()**  
[A value representing the current minute]

**Numeric := ?TimeSecond ()**  
[A value representing the current second]

**String := ?TOADefinition ()**  
[The current table of authorities section name]

**String := ?TOAShortFormMarks ()**  
[The name of the last table of authorities short form marked]

**String := ?TOAShortForms ()**  
[The name of the last short form encountered in a table of authorities]

**Numeric := ?TOCDefinition ()**  
[The number of levels defined for the current table of contents]

**Boolean := ?TypeoverActive ()**  
[A value representing whether Typeover is active]

- U -

**Numeric := ?UndeleteBuffers ()**  
[The number of buffers available to undelete]

**Numeric := ?UnderlineColorBlue ()**  
[A value representing the amount of blue in the current underline color]

**Numeric := ?UnderlineColorGreen ()**  
[A value representing the amount of green in the current underline color]

**Numeric := ?UnderlineColorRed ()**  
[A value representing the amount of red in the current underline color]

**Numeric := ?UnderlineColorShading ()**  
[A value representing the amount of shade in the current underline color]

**Boolean := ?UnderlineSpaces ()**  
[A value representing whether blank spaces are underlined when underline is on]

**Numeric := ?UnderlineStyle ()**  
[A value representing style of underline when underline is on]

**Boolean := ?UnderlineTabs ()**  
[A value representing whether tabs are underlined when underline is on]

**Numeric Measurement := ?UnprintableBottomMargin ()**  
[An expression indicating the size of the unprintable bottom margin in WordPerfect units]

**Numeric Measurement := ?UnprintableLeftMargin ()**  
[An expression indicating the size of the unprintable left margin in WordPerfect units]

**Numeric Measurement := ?UnprintableRightMargin ()**  
[An expression indicating the size of the unprintable right margin in WordPerfect units]

**Numeric Measurement := ?UnprintableTopMargin ()**  
[An expression indicating the size of the unprintable top margin in WordPerfect units]

**Boolean := ?UpdateQuickList ()**  
[A value representing whether the Update QuickList with Changes file preference is on or off]

**Boolean := ?UseDefaultDocExt ()**  
[A value representing whether Use Default Extension on Open and Save is on in File Settings]

**Numeric := ?UserColorBlue ()**  
[A value representing the amount of blue in the current user-defined color]

**Numeric := ?UserColorGreen ()**  
[A value representing the amount of green in the current user-defined color]

**Numeric := ?UserColorRed ()**  
[A value representing the amount of red in the current user-defined color]

**String := ?UserInitials ()**  
[The user initials specified in Environment Settings]

**String := ?UserName ()**  
[The user name specified in Environment Settings]

- V -

**Numeric := ?VolumeNumber ()**  
[The current volume number]

**Numeric := ?VolumeNumberMethod ()**  
[A value representing the current volume numbering method]

- W -

**Boolean := ?WatermarkAInParagraph ()**  
[Indicates whether a Watermark A code is in the current paragraph]

**Numeric := ?WatermarkAOccur ()**  
[A value representing whether Watermark A occurs on odd, even, or all pages]

**Boolean := ?WatermarkBInParagraph ()**  
[Indicates whether a Watermark B code is in the current paragraph]

**Numeric := ?WatermarkBOccur ()**  
[A value representing whether Watermark B occurs on odd, even, or all pages]

**Numeric := ?WatermarkGraphicShade ()**  
[A percentage indicating the shading value of the current watermark]

**String := ?WebBackgroundImage ()**  
[Obsolete Command]

**String := ?WebBaseURL ()**  
[The base URL for the current web document]

**String := ?WebBoxHTMLProperties ()**  
[A value representing the HTML properties of the current box]

**String := ?WebMetaTags ()**  
[Obsolete Command]

**Any := ?WebTitle ()**  
[Obsolete Command]

**Boolean := ?WidowOrphan ()**  
[A value indicating whether Widow/Orphan is on or off]

**Boolean := ?WritingToolsDisabledState ()**  
[A value representing whether Disable Writing Tools is on or off]

- Z -

**Numeric := ?Zoom ()**  
[A percentage indicating the size of the current view]

## - WORDPERFECT 10 PRODUCT COMMANDS -

### - A -

#### **AbbreviationCreate (**

**AbbreviationName:String;**  
**[Template:Enumeration {**  
    **CurrentDoc!;**  
    **PersonalLibrary!;**  
    **SharedLibrary!;**  
    **QuickWords!});**  
**[Text:String]**  
[Create a QuickWords entry]

#### **AbbreviationDelete (**

**AbbreviationName:String;**  
**[Template:Enumeration {**  
    **CurrentDoc!;**  
    **PersonalLibrary!;**  
    **SharedLibrary!;**  
    **QuickWords!});**  
[Delete a QuickWords entry]

#### **AbbreviationExpand (**

**[AbbreviationName:String];**  
**[Template:Enumeration {**  
    **CurrentDoc!;**  
    **PersonalLibrary!;**  
    **SharedLibrary!;**  
    **QuickWords!});**  
[Insert the current QuickWord]

#### **AbbreviationsShippingMacro (**

[Create and insert multiple QuickWords]

#### **AboutDlg (**

[Display product and license information]

#### **Boolean := AddressBookDlg (**

**[SelectionType:Enumeration {**  
    **SingleSelection!;**  
    **MultipleSelection!});**  
**[CurrentSelections:Enumeration {**  
    **Ignore!;**  
    **Reselect!});**  
**[AddressBook:String];**  
**[DialogTitle:String];**  
**[OKButtonText:String]**  
[Create, edit, and delete addresses]

#### **AddressBookFormatAdd (Name:String;Format:String)**

[Create a custom address format]

#### **AddressBookFormatDelete (Name:String)**

[Delete a custom address format]

#### **Numeric := AddressBookFormatGetCount (**

[A value representing the number of available address formats]

#### **String := AddressBookFormatGetCurrent (Item:Enumeration {**     **Name!;**     **Format!})**

[Return the current address format name or specification]

#### **String := AddressBookFormatGetFormat (Index:Numeric)**

[Return the specified address format]

#### **String := AddressBookFormatGetName (Index:Numeric)**

[Return the name of the specified address format]

#### **Numeric := AddressBookGetCount (**

[Return the number of available address books]

#### **Numeric := AddressBookGetFieldCount (AddressBook:String)**

[Return the number of available fields in the specified address book]

#### **String := AddressBookGetFieldName (**

**AddressBook:String;**  
**Index:Numeric)**  
[Return the name of a field in an address book]

#### **Numeric := AddressBookGetFieldNumber (**

**AddressBook:String;**  
**Index:Numeric)**  
[Return the code number of a field in an address book]

#### **String := AddressBookGetName (Index:Numeric)**

[Return the name of the specified address book]

#### **AddressBookInsertAddr (**

[Insert an address]

#### **AddressMergeShippingMacro (**

[Insert the address book into a merge data file]

#### **Advance (**

**Where:Enumeration {**  
    **AdvanceLeft!;**  
    **Left!;**  
    **AdvanceRight!;**  
    **Right!;**  
    **AdvanceFromLeftEdge!;**  
    **ToPosition!;**  
    **AdvanceUp!;**  
    **Up!;**  
    **AdvanceDown!;**  
    **Down!;**  
    **AdvanceFromTop!;**  
    **ToLine!};**  
**Amount:Numeric Measurement)**  
[Place the insertion point at a specified location]

#### **AdvanceDlg (**

[Move subsequent text up, down, right, or left]

#### **BoxAlignDistrDlgProc (**

[Align objects in your document]

#### **AllFontsShippingMacro (**

[Create a list of available font faces for the current printer]

#### **Append (**

[Copy the selection to the end of the Clipboard contents]

#### **AppendToFile ([Filename:String])**

[Add selected text, or the current document if no text is selected, to the end of another file]

#### **ApplicationBarShow ([State:Enumeration {Off!;On!}])**

[Show the application bar]

#### **ApplyHeading1 (**

[Apply heading 1 format]

#### **ApplyHeading2 (**

[Apply heading 2 format]

#### **ApplyHeading3 (**

[Apply heading 3 format]

#### **ApplyHeading4 (**

[Apply heading 4 format]

**ApplyHeading5 ()**  
[Apply heading 5 format]

**AppMaximize ()**  
[Expand the WordPerfect 10 window to its maximum size]

**AppMinimize ()**  
[Reduce the WordPerfect 10 window to an icon]

**AppMove ()**  
[Move the WordPerfect 10 window to another location on the screen]

**AppRestore ()**  
[Restore the WordPerfect 10 window to its previous size]

**AppSize ()**  
[Change the size of the WordPerfect 10 window]

**AppTaskList ()**  
[Activate the Taskbar and display the Start menu]

**AssociateDlg ()**  
[Assign menus and keyboards to selected WordPerfect 10 features]

**AttributeAppearanceOff** ([[Attrib:Enumeration {  
Outline!  
Italics!  
Shadow!  
Redline!  
DoubleUnderline!  
Bold!  
Strikeout!  
Underline!  
SmallCaps!  
Every!}}])  
[Turn off one or more font attributes]

**AttributeAppearanceOn** ([[Attrib:Enumeration {  
Outline!  
Italics!  
Shadow!  
Redline!  
DoubleUnderline!  
Bold!  
Strikeout!  
Underline!  
SmallCaps!  
Every!}}])  
[Turn on one or more font attributes]

**AttributeAppearanceToggle** ([[Attrib:Enumeration {  
Outline!  
Italics!  
Shadow!  
Redline!  
DoubleUnderline!  
Bold!  
Strikeout!  
Underline!  
SmallCaps!  
Every!}}])  
[Change the appearance, position, and relative size of text]

**AttributeNormal ()**  
[Turn off appearance and relative size attributes from selected text]

**AttributePosition** (Position:Enumeration {  
Superscript!  
Subscript!  
NormalPosition!})  
[Adjust the vertical position of selected or subsequent text relative to the current text line]

**AttributePositionToggle** (Position:Enumeration {  
Superscript!  
Subscript!  
NormalPosition!})  
[Toggle the vertical position of text]

**AttributeRelativeSize** (Size:Enumeration {  
ExtraLarge!  
VeryLarge!  
Large!  
Small!  
Fine!  
NormalSize!})  
[Specify a relative font size for selected or subsequent text]

**AttributeRelativeSizeToggle** (Size:Enumeration {  
ExtraLarge!  
VeryLarge!  
Large!  
Small!  
Fine!  
NormalSize!})  
[Toggle the specified relative font size]

**AttributeSizeRatios** (  
Attrib:Enumeration {  
ExtraLarge!  
VeryLarge!  
Large!  
Small!  
Fine!  
SuperSubscript!  
Percentage:Numeric)  
[Specify relative font size ratios]

**AutoCodePlacement** (State:Enumeration {Off!; On!})  
[Turn Auto Code Placement on or off]

**AutoscrollOn ()**  
[Turn on autoscroll]

**AutoWordSelect** (State:Enumeration {No!;Yes!})  
[Turn Automatically Select Words on or off]

- B -

**BackTab ()**  
[Move the current line of text left one tab from the insertion point]

**Boolean := Backup** ([State:Enumeration {Off!;On!})  
[Turn Timed Document Backup on or off]

**BackupMinutes** (Minutes:Numeric)  
[Specify the number of minutes between timed backups]

**BackupOriginalDoc** (State:Enumeration {Off!; On!})  
[Turn Original Document Backup on or off]

**BarcodeDlg ()**  
[Create POSTNET bar codes]

**BarcodePOSTNET** (  
Zip:String;  
[State:Enumeration {  
MergeFieldAllowed!}])  
[Specify a ZIP Code or a merge field containing a ZIP Code]

**BaselinePlacement** (State:Enumeration {Off!;On!})  
[Turn Baseline placement for typesetting on or off]

**Binding** (  
Edge:Enumeration {  
LeftEdge!

**RightEdge!**;  
**TopEdge!**;  
**BottomEdge!**};  
**Offset:Numeric Measurement**;  
**[Alternating:Enumeration {Off!; On!}]**)  
[Specify a binding offset and location]

**BindingOptionsDlg ()**  
[Specify the binding width and duplexing options]

**Boolean := BlockProtect ([State:Enumeration {Off!;On!}])**  
[Turn Block Protect on or off]

**BlockProtectToggle ()**  
[Turn on/off Block Protect]

**BoldKey ()**  
[Turn on Bold]

**BookmarkBlock ([Name:String])**  
[Find and select a specified bookmark]

**BookmarkCreate (**  
  **Name:String;**  
  **[Selected:Enumeration {No!; Yes!}]**)  
[Create a bookmark]

**BookmarkDelete ([Name:String])**  
[Delete a specified bookmark]

**BookmarkDlg ()**  
[Create, move, delete, rename, and go to bookmarks]

**BookmarkFind ([Name:String])**  
[Find a specified bookmark]

**BookmarkMove ([Name:String])**  
[Move a bookmark to the insertion point]

**BookmarkRename (OldName:String; NewName:String)**  
[Rename a bookmark]

**BorderBottomLine (Style:Any {**  
  **SingleLine!**;  
  **DoubleLine!**;  
  **DashedLine!**;  
  **DottedLine!**;  
  **ThickLine!**;  
  **ExtraThickLine!**;  
  **ThinThickLine!**;  
  **ThickThinLine!**;  
  **ButtonTopLeftLine!**;  
  **ButtonBottomRightLine!**;  
  **HeavySingleLine!**;  
  **HeavyDoubleLine!**;  
  **TableDefaultLine!**;  
  **NoLine!})**  
[Specify the style of a bottom border line]

**BorderColor (**  
  **[ColorName:String];**  
  **[Red:Numeric];**  
  **[Green:Numeric];**  
  **[Blue:Numeric];**  
  **[Shade:Numeric])**  
[Specify the color of a paragraph, page, or column border]

**BorderCornerRadius (Radius:Numeric Measurement)**  
[Specify the curve of border corners]

**BorderDropShadow (**  
  **[Location:Enumeration {**  
    **NoShadow!**

**UpperLeft!**;  
  **LowerLeft!**;  
  **LowerRight!**;  
  **UpperRight!}]**);  
**[Size:Numeric Measurement];**  
**[ColorName:String];**  
**[Red:Numeric];**  
**[Green:Numeric];**  
**[Blue:Numeric];**  
**[Shade:Numeric];**  
**[VerticalSize:Numeric Measurement]**)  
[Specify the location, size, and color of a border shadow]

**BorderInsideSpacing (**  
  **[Bottom:Numeric Measurement];**  
  **[Left:Numeric Measurement];**  
  **[Right:Numeric Measurement];**  
  **[Top:Numeric Measurement]**)  
[Specify inside border spacing]

**BorderLeftLine (Style:Any {**  
  **SingleLine!**;  
  **DoubleLine!**;  
  **DashedLine!**;  
  **DottedLine!**;  
  **ThickLine!**;  
  **ExtraThickLine!**;  
  **ThinThickLine!**;  
  **ThickThinLine!**;  
  **ButtonTopLeftLine!**;  
  **ButtonBottomRightLine!**;  
  **HeavySingleLine!**;  
  **HeavyDoubleLine!**;  
  **TableDefaultLine!**;  
  **NoLine!})**  
[Specify the left line style of a border]

**BorderOutsideSpacing (**  
  **Left:Numeric Measurement;**  
  **Right:Numeric Measurement;**  
  **Top:Numeric Measurement;**  
  **Bottom:Numeric Measurement)**  
[Specify outside border spacing]

**BorderRightLine (Style:Any {**  
  **SingleLine!**;  
  **DoubleLine!**;  
  **DashedLine!**;  
  **DottedLine!**;  
  **ThickLine!**;  
  **ExtraThickLine!**;  
  **ThinThickLine!**;  
  **ThickThinLine!**;  
  **ButtonTopLeftLine!**;  
  **ButtonBottomRightLine!**;  
  **HeavySingleLine!**;  
  **HeavyDoubleLine!**;  
  **TableDefaultLine!**;  
  **NoLine!})**  
[Specify a right line border style]

**BorderSeparatorLine (Style:Any {**  
  **SingleLine!**;  
  **DoubleLine!**;  
  **DashedLine!**;  
  **DottedLine!**;  
  **ThickLine!**;  
  **ExtraThickLine!**;  
  **ThinThickLine!**;  
  **ThickThinLine!**;  
  **ButtonTopLeftLine!**;  
  **ButtonBottomRightLine!**

HeavySingleLine!  
HeavyDoubleLine!  
TableDefaultLine!  
NoLine!}}

[Specify a separator line style in a paragraph or column border]

**BorderStyleSpacing** (State:Enumeration {No!;Yes!})

[Turn on Auto Spacing in a border style]

**BorderStyleCopy** (

Style:Any {  
SpacingOnly!  
SingleBorder!  
DoubleBorder!  
DashedBorder!  
DottedBorder!  
ThickBorder!  
ExtraThickBorder!  
ThinThickBorder!  
ThickThinBorder!  
ThickTopBottomBorder!  
ButtonBorder!  
ColumnBorderBetween!  
ColumnBorderAll!  
HeavySingleBorder!  
HeavyDoubleBorder!  
NoBorder!};

[FromLibrary:Enumeration {  
CurrentDocument!  
PersonalLibrary!  
SharedLibrary!};

[ToLibrary:Enumeration {  
CurrentDocument!  
PersonalLibrary!  
SharedLibrary!};

[DestName:String])

[Copy a border style]

**BorderStyleCreate** (

Style:String;  
[Library:Enumeration {  
CurrentDoc!  
PersonalLibrary!  
SharedLibrary!};

[Create a border style]

**BorderStyleDelete** (

Style:Any {  
SpacingOnly!  
SingleBorder!  
DoubleBorder!  
DashedBorder!  
DottedBorder!  
ThickBorder!  
ExtraThickBorder!  
ThinThickBorder!  
ThickThinBorder!  
ThickTopBottomBorder!  
ButtonBorder!  
ColumnBorderBetween!  
ColumnBorderAll!  
HeavySingleBorder!  
HeavyDoubleBorder!  
NoBorder!};

[Library:Enumeration {  
CurrentDoc!  
PersonalLibrary!  
SharedLibrary!};

[Delete a border style]

**BorderStyleEdit** (

Style:Any {

SpacingOnly!  
SingleBorder!  
DoubleBorder!  
DashedBorder!  
DottedBorder!  
ThickBorder!  
ExtraThickBorder!  
ThinThickBorder!  
ThickThinBorder!  
ThickTopBottomBorder!  
ButtonBorder!  
ColumnBorderBetween!  
ColumnBorderAll!  
HeavySingleBorder!  
HeavyDoubleBorder!  
NoBorder!};

[Library:Enumeration {  
CurrentDoc!  
PersonalLibrary!  
SharedLibrary!};

[Specify a border style to edit]

**BorderStyleEnd** (State:Enumeration {Cancel!; Save!})

[Save border style changes]

**BorderStyleName** (Style:String)

[Rename a border style]

**BorderStyleRetrieve** (

Filename:String;  
FromLibrary:Enumeration {  
CurrentDoc!  
PersonalLibrary!  
SharedLibrary!};

[Retrieve all border graphics styles from the specified document or template]

**BorderStyleSave** (

Filename:String;  
Library:Enumeration {  
CurrentDoc!  
PersonalLibrary!  
SharedLibrary!};

[Save border styles to a specified library]

**BorderTopLine** (Style:Any {

SingleLine!  
DoubleLine!  
DashedLine!  
DottedLine!  
ThickLine!  
ExtraThickLine!  
ThinThickLine!  
ThickThinLine!  
ButtonTopLeftLine!  
ButtonBottomRightLine!  
HeavySingleLine!  
HeavyDoubleLine!  
TableDefaultLine!  
NoLine!});

[Specify the top line style of a border]

**BorderUseBorderColor** (State:Enumeration {No!; Yes!})

[Turn on Border Color]

**BoxAddText** ()

[Add a text box to a draw object box]

**BoxAttachTo** (Anchor:Enumeration {

Page!  
Paragraph!  
Character!});

[Anchor a graphics box to a page, paragraph, or character]

**BoxBackwardOne ()**

[Move the selected object(s) back one layer in the document]

**BoxBorder (Style:Any {**

SpacingOnly!;  
 SingleBorder!;  
 DoubleBorder!;  
 DashedBorder!;  
 DottedBorder!;  
 ThickBorder!;  
 ExtraThickBorder!;  
 ThinThickBorder!;  
 ThickThinBorder!;  
 ThickTopBottomBorder!;  
 ButtonBorder!;  
 ColumnBorderBetween!;  
 ColumnBorderAll!;  
 HeavySingleBorder!;  
 HeavyDoubleBorder!;  
 NoBorder!})

[Specify a box border style]

**BoxBorderDig ()**

[Change the current box border, shadow, and fill styles]

**BoxCaptionEdit ()**

[Begin caption editing commands]

**BoxCaptionEditEnd ()**

[End caption editing commands]

**BoxCaptionFormatting (SizeAuto:Any {AutoWidth!})**

[Specify caption width]

**BoxCaptionInitialStyle (Style:Any {**

NormalStyle!;  
 FootnoteNumberInDocumentStyle!;  
 EndnoteNumberInDocumentStyle!;  
 FigureNumberStyle!;  
 TableBoxNumberStyle!;  
 TextBoxNumberStyle!;  
 UserBoxNumberStyle!;  
 EquationNumberStyle!;  
 HypertextStyle!;  
 ToC1Style!;  
 ToC2Style!;  
 ToC3Style!;  
 ToC4Style!;  
 ToC5Style!;  
 Index1Style!;  
 Index2Style!;  
 ListStyle!;  
 ToAStyle!;  
 DocStyle!;  
 BoxTextStyle!;  
 CaptionStyle!;  
 FootnoteStyle!;  
 EndnoteStyle!;  
 CommentStyle!;  
 HeaderAStyle!;  
 HeaderBStyle!;  
 FooterAStyle!;  
 FooterBStyle!;  
 WatermarkAStyle!;  
 WatermarkBStyle!;  
 ParagraphOutInStyle!;  
 OutlineStyle!;  
 LegalOutInStyle!;  
 BulletsOutInStyle!;  
 HeadingsOutInStyle!;  
 Legal\_2OutInStyle!;  
 NumbersOutInStyle!;  
 Level1Style!;

[Specify an initial caption style]

Level2Style!;  
 Level3Style!;  
 Level4Style!;  
 Level5Style!;  
 Level6Style!;  
 Level7Style!;  
 Level8Style!;  
 Legal1Style!;  
 Legal2Style!;  
 Legal3Style!;  
 Legal4Style!;  
 Legal5Style!;  
 Legal6Style!;  
 Legal7Style!;  
 Legal8Style!;  
 Heading1Style!;  
 Heading2Style!;  
 Heading3Style!;  
 Heading4Style!;  
 Heading5Style!;  
 Heading6Style!;  
 Heading7Style!;  
 Heading8Style!;  
 MonospacedStyle!;  
 CustomHTMLStyle!;  
 PreformattedStyle!;  
 AddressStyle!;  
 QuotationsOutInStyle!;  
 DefinitionsOutInStyle!;  
 Quotation1Style!;  
 Quotation2Style!;  
 Quotation3Style!;  
 Quotation4Style!;  
 Quotation5Style!;  
 Quotation6Style!;  
 Quotation7Style!;  
 Quotation8Style!;  
 Definition1Style!;  
 Definition2Style!;  
 Definition3Style!;  
 Definition4Style!;  
 Definition5Style!;  
 Definition6Style!;  
 Definition7Style!;  
 Definition8Style!;  
 Level9Style!;  
 Legal9Style!;  
 Heading9Style!;  
 Quotation9Style!;  
 Definition9Style!})

**BoxCaptionNumberStyle (Style:Any {**  
 NormalStyle!;  
 FootnoteNumberInDocumentStyle!;  
 EndnoteNumberInDocumentStyle!;  
 FigureNumberStyle!;  
 TableBoxNumberStyle!;  
 TextBoxNumberStyle!;  
 UserBoxNumberStyle!;  
 EquationNumberStyle!;  
 HypertextStyle!;  
 ToC1Style!;  
 ToC2Style!;  
 ToC3Style!;  
 ToC4Style!;  
 ToC5Style!;  
 Index1Style!;  
 Index2Style!;  
 ListStyle!;  
 ToAStyle!;  
 DocStyle!;



BoxTextStyle!  
 CaptionStyle!  
 FootnoteStyle!  
 EndnoteStyle!  
 CommentStyle!  
 HeaderAStyle!  
 HeaderBStyle!  
 FooterAStyle!  
 FooterBStyle!  
 WatermarkAStyle!  
 WatermarkBStyle!  
 ParagraphOutInStyle!  
 OutlineStyle!  
 LegalOutInStyle!  
 BulletOutInStyle!  
 HeadingsOutInStyle!  
 Legal\_2OutInStyle!  
 NumbersOutInStyle!  
 Level1Style!  
 Level2Style!  
 Level3Style!  
 Level4Style!  
 Level5Style!  
 Level6Style!  
 Level7Style!  
 Level8Style!  
 Legal1Style!  
 Legal2Style!  
 Legal3Style!  
 Legal4Style!  
 Legal5Style!  
 Legal6Style!  
 Legal7Style!  
 Legal8Style!  
 Heading1Style!  
 Heading2Style!  
 Heading3Style!  
 Heading4Style!  
 Heading5Style!  
 Heading6Style!  
 Heading7Style!  
 Heading8Style!  
 MonospacedStyle!  
 CustomHTMLStyle!  
 PreformattedStyle!  
 AddressStyle!  
 QuotationsOutInStyle!  
 DefinitionsOutInStyle!  
 Quotation1Style!  
 Quotation2Style!  
 Quotation3Style!  
 Quotation4Style!  
 Quotation5Style!  
 Quotation6Style!  
 Quotation7Style!  
 Quotation8Style!  
 Definition1Style!  
 Definition2Style!  
 Definition3Style!  
 Definition4Style!  
 Definition5Style!  
 Definition6Style!  
 Definition7Style!  
 Definition8Style!  
 Level9Style!  
 Legal9Style!  
 Heading9Style!  
 Quotation9Style!  
 Definition9Style!)

[Specify a caption numbering style]

**BoxCaptionPosition** (

Side:Enumeration {  
   Left!  
   Top!  
   Right!  
   Bottom!};  
 Border:Enumeration {  
   OutsideBorder!  
   InsideBorder!  
   OnBorder!};  
 [Alignment:Enumeration {  
   Left!  
   Top!  
   Center!  
   Right!  
   Bottom!});  
 [AlignmentOffset:Any])  
 [Specify caption position]

**BoxCaptionRotation** (Rotation:Enumeration {  
   None!  
   Degrees90!  
   Degrees180!  
   Degrees270!})  
 [Rotate captions]

**BoxChangeLineHeight** (State:Enumeration {No!; Yes!})  
 [Specify whether line height changes when a box is attached to a character]

**BoxChangeStyle** (BoxType:Any {  
   FigureBox!  
   TableBox!  
   TextBox!  
   UserBox!  
   EquationBox!  
   ButtonBox!  
   WatermarkImageBox!  
   InlineEquationBox!  
   NoBox!})  
 [Specify a box style]

**BoxContentEdit** ()  
 [Edit the object with the OLE server]

**BoxContentPosition** (  
   Horizontal:Enumeration {  
     Left!  
     Center!  
     Right!};  
   Vertical:Enumeration {  
     Top!  
     Center!  
     Bottom!})  
 [Position the box contents]

**BoxContentPreserveAspectRatio** (State:Enumeration {No!; Yes!})  
 [Preserve the image width/height ratio when the box changes]

**BoxContentType** (Content:Enumeration {  
   Text!  
   LinkedText!  
   Image!  
   Equation!  
   Presentation!  
   Video!  
   Macro!  
   External!  
   Empty!})  
 [Specify the box content type]

**BoxCounter** (CounterName:Any {  
   None!  
   FigureCounter!;

TableBoxCounter!;  
TextBoxCounter!;  
UserBoxCounter!;  
EquationCounter!}}

[Assign a new counter to the current box or box style]

**BoxCreate** (BoxType:Any {  
FigureBox!;  
TableBox!;  
TextBox!;  
UserBox!;  
EquationBox!;  
ButtonBox!;  
WatermarkImageBox!;  
InlineEquationBox!;  
NoBox!})

[Create a box]

**BoxDelete** (  
BoxType:Enumeration {  
FigureBox!;  
TableBox!;  
TextBox!;  
UserBox!;  
EquationBox!;  
ButtonBox!;  
WatermarkImageBox!;  
InlineEquationBox!;  
NoBox!};  
BoxNumber:Numeric)

[Delete a box]

**BoxEdit** (BoxNumber:Numeric)  
[Specify the number of a box to edit]

**BoxEditByCounter** (  
CounterName:Any {  
None!;  
FigureCounter!;  
TableBoxCounter!;  
TextBoxCounter!;  
UserBoxCounter!;  
EquationCounter!};  
{{CounterLevelNumber:Numeric}})

[Specify the counter name and level number of a box to edit]

**BoxEditNext** ([BoxType:Enumeration {  
FigureBox!;  
TableBox!;  
TextBox!;  
UserBox!;  
EquationBox!;  
ButtonBox!;  
WatermarkImageBox!;  
InlineEquationBox!;  
NoBox!}])

[Begin editing commands for the next box]

**BoxEditPrevious** ([BoxType:Enumeration {  
FigureBox!;  
TableBox!;  
TextBox!;  
UserBox!;  
EquationBox!;  
ButtonBox!;  
WatermarkImageBox!;  
InlineEquationBox!;  
NoBox!}])

[Begin editing commands for the previous box]

**BoxEnd** (  
[State:Enumeration {Cancel!; Save!}];

[BoxOptionsBar:Enumeration {Hide!; DontHide!}])  
[Save the editing changes to a box and display or hide the Graphics feature bar]

**BoxEquationColor** (  
[ColorName:String];  
[Red:Numeric];  
[Green:Numeric];  
[Blue:Numeric];  
[Shade:Numeric])

[Specify a box equation color and shading value]

**BoxEquationDefaultFont** (FontSize:Any {DefaultSize!})  
[Specify a default font and size]

**BoxEquationFilename** (Filename:String)  
[Specify the name of an equation file and designate box contents as an equation]

**BoxEquationFont** (  
FontName:String;  
[FontSize:Numeric Measurement];  
[Family:Enumeration {  
FamilyTCBenquiat!;  
FamilyTimesRoman!;  
FamilyTCGaramond!;  
FamilyCenturySchoolbook!;  
FamilyBodoni!;  
FamilyTCLubalinGraph!;  
FamilyAlbertus!;  
FamilyCourier!;  
FamilyHelvetica!;  
FamilyTCRonda!;  
FamilyMicrostyle!;  
FamilyOptima!;  
FamilyLetterGothic!;  
FamilyBrush!;  
FamilyParkAvenue!;  
FamilySignetRoundhand!;  
FamilyTekton!;  
FamilyUncial!;  
FamilyOldEnglish!;  
FamilyTCZapfDingbats!;  
FamilyBroadway!;  
FamilyComputer!;  
FamilyPTBarnum!;  
FamilyPonderosa!;  
FamilyMadrone!;  
FamilyHobol!;  
FamilyRevue!;  
FamilyAestheticOrnamented!});  
[Attributes:Enumeration {  
FontMatchNormal!;  
FontMatchItalic!;  
FontMatchOutline!;  
FontMatchShadow!;  
FontMatchSmallCaps!});  
[Weight:Enumeration {  
WeightUnknown!;  
WeightUltraThin!;  
WeightUltraLight!;  
WeightExtraThin!;  
WeightThin!;  
WeightExtraLight!;  
WeightLight!;  
WeightDemiLight!;  
WeightSemiLight!;  
WeightBook!;  
WeightRegular!;  
WeightRoman!;  
WeightMedium!;  
WeightDemiBold!};

**WeightSemiBold!**;  
**WeightBold!**;  
**WeightExtraBold!**;  
**WeightHeavy!**;  
**WeightExtraHeavy!**;  
**WeightBlack!**;  
**WeightExtraBlack!**;  
**WeightUltraHeavy!**;  
**WeightUltraBlack!**;  
**WeightMaximum!}}**;  
**[Width:Enumeration {**  
**WidthUnknown!**;  
**WidthUltraCondensed!**;  
**WidthUltraCompressed!**;  
**WidthCompressed!**;  
**WidthExtraCondensed!**;  
**WidthCondensed!**;  
**WidthSemiCondensed!**;  
**WidthNormal!**;  
**WidthSemiExpanded!**;  
**WidthExpanded!**;  
**WidthExtraExpanded!**;  
**WidthUltraExpanded!**;  
**WidthDoubleWide!**;  
**WidthTripleWide!}}**;  
**[Source:Enumeration {**  
**PRSFile!**;  
**DRSFile!}}**;  
**[Type:Enumeration {**  
**Speedo!**;  
**Intellifont!**;  
**Type1!**;  
**TrueType!}}**;  
**[CharacterSet:Enumeration {**  
**FontMatchASCII!**;  
**FontMatchMultinational1!**;  
**FontMatchMultinational2!**;  
**FontMatchBoxDrawing!**;  
**FontMatchTypographicSymbols!**;  
**FontMatchIconicSymbols!**;  
**FontMatchMath!**;  
**FontMatchMathExtension!**;  
**FontMatchGreek!**;  
**FontMatchHebrew!**;  
**FontMatchCyrillic!**;  
**FontMatchJapaneseKana!**;  
**FontMatchUserDefined!**;  
**FontMatchArabic!**;  
**FontMatchArabicScript!}}**)  
[Specify an equation font]

**BoxEquationFontSize (Size:Numeric Measurement)**  
[Specify an equation font size]

**BoxEquationRedisplay (Zoom:Numeric)**  
[Enlarge or reduce an equation display]

**BoxEquationSaveAsGraphic (**  
**Filename:String;**  
**[Format:Enumeration {**  
**WordPerfectGraphic\_10!**;  
**WordPerfectGraphic\_20!**;  
**Presentations\_20!}}**)  
[Save an equation box as a graphics file]

**BoxFill (FillStyleName:Any {**  
**Fill10!**;  
**Fill120!**;  
**Fill130!**;  
**Fill140!**;  
**Fill150!**;  
**Fill160!**;  
**Fill170!**;  
**Fill180!**;  
**Fill190!**;  
**Fill100!**;  
**FillButton!**;  
**Fill5!**;  
**NoFill!}}**)  
[Select a box fill style]

**BoxForwardOne (**  
[Move the selected object(s) forward one layer in the document]

**BoxHeight (Height:Any {**  
**AutoHeight!**;  
**MinHeight!}}**)  
[Specify box height]

**BoxHorizontalAlignment (**  
**Alignment:Enumeration {AlignMargins!; AlignColumns!;**  
**Position:Enumeration {**  
**Left!**;  
**Center!**;  
**Right!**;  
**FullAlign!};**  
**[Offset:Numeric Measurement];**  
**[LeftColumn:Numeric];**  
**[RightColumn:Numeric]**)  
[Specify the alignment of a box attached to a page or paragraph]

**BoxHorizontalPosition (Position:Numeric Measurement)**  
[Specify the horizontal position of a box relative to the left edge of the page]

**BoxImageBlackWhiteThreshold (Threshold:Numeric)**  
[Specify the point at which colors and/or grays in the current image convert to black and white]

**BoxImageBrightness (Amount:Numeric)**  
[Set the brightness level for the image]

**BoxImageContrast (Contrast:Numeric)**  
[Set the contrast level for the image]

**BoxImageDitherMethod (**  
**Method:Enumeration {**  
**Halftoning!**;  
**OrderedDither!**;  
**ErrorDiffusionDither!**;  
**Default!};**  
**Source:Enumeration {**  
**WordPerfect!**;  
**Printer!**;  
**Default!**;  
**External!}}**)  
[Specify a dither method and source]

**BoxImageFill (ImageFillType:Enumeration {**  
**Normal!**;  
**Transparent!**;  
**White!}}**)  
[Select normal, no fill, or white fill]

**BoxImageFlipX (State:Enumeration {No!; Yes!})**  
[Mirror the image around its horizontal axis]

**BoxImageFlipY (State:Enumeration {No!; Yes!})**  
[Mirror the image around its vertical axis]

**BoxImageHalfToneOptions (**  
**HalfTone:Enumeration {**  
**DefaultValues!**;  
**SetValues!};**  
**[LPI:Numeric];**

[Plane0:Numeric];  
[Plane1:Numeric];  
[Plane2:Numeric];  
[Plane3:Numeric]

[Specify the default or user-defined halftone values for the active printer]

**BoxImageInvertColors** (State:Enumeration {No!; Yes!})

[Change the colors in the image to their complementary colors]

**BoxImageMonochrome** (State:Enumeration {No!; Yes!})

[Display the image in black and white only, and set the threshold for blackness]

**BoxImageMove** (Xposition:Numeric; YPosition:Numeric)

[Move the image within the box]

**BoxImageRetrieve** (

Action:Enumeration {

LeaveExternal!;

MakeInternal!;

UseInternal!};

Filename:String;

[FileFormat:Enumeration {

MacPaint!;

GEMIMG!;

FAX\_TIFF!;

CorelVentura!;

AdobePhotoShop!;

CorelWavelen!;

ScitexCT!;

WordPerfectGraphic\_10!;

WordPerfectGraphic\_20!;

TaggedImageFileFormat!;

PCPaintbrushX!;

WindowsMetaFile!;

Bitmap!;

CGM!;

AutoCAD\_DXF!;

EncapsulatedPostScript!;

HewlettPackardGraphicPlotter!;

Macintosh\_PICT!;

Micrografx!;

Presentations\_20!;

LotusPIC!;

CompuserveGIF!;

TarGA!;

EnhancedWindowsMetaFile!;

WPWorksPaint\_20!;

JPEG!;

KodakPhotoCD!;

CompuserveGIF24!;

CorelPhotoPaint!;

MicrografxPicturePublisher4!;

CALSBBitmap!;

BIFF Charts!;

WPChartingStyle\_10!;

WPChartingData\_10!;

WPChartingStyle\_10E!;

WPChartingData\_10E!;

WPChartingStyle\_20!;

WPChartingData\_20!;})

[Retrieve an image]

**BoxImageRotate** (Rotation:Numeric)

[Rotate the image around a selected point]

**BoxImageSave** (

Filename:String;

Format:Enumeration {

WordPerfectGraphic\_10!;

WordPerfectGraphic\_20!;

**Presentations\_20!})**

[Save an image, and optionally rename it and specify a graphics format]

**BoxImageScaling** (Xscale:Numeric; YScale:Numeric)

[Scale the image (selected area, about image center, or reset)]

**BoxImageShowBackground** (State:Enumeration {No!; Yes!})

[Display the background colors or gradient saved with an image]

**BoxInitialStyleDlg** ()

[Change text box settings]

**BoxOLEEditObject** ()

[Edit the box contents with the OLE server]

**BoxOverlap** (State:Enumeration {No!; Yes!})

[Specify whether to overlap the current box over other boxes]

**BoxPosDlg** ()

[Attach the current box to a page, paragraph, or character, and specify its position]

**BoxResetBorder** ()

[Reset the border values for the active box to the values of the corresponding box style]

**BoxResetCaption** ()

[Reset the caption values for the active box to the caption values of the corresponding box style]

**BoxResetContent** ()

[Reset the content values for the active box to the content values of the corresponding box style]

**BoxResetContentRender** ()

[Reset the content render values for the active box to the values of the corresponding box style]

**BoxResetCounter** ()

[Reset the counter values for the active box to the values of the corresponding box style]

**BoxResetFill** ()

[Reset the fill values for the active box to the values of the corresponding box style]

**BoxResetPosition** ()

[Reset the position values for the active box to the position values for the corresponding box style]

**BoxResetTextFlow** ()

[Reset the text wrapping values for the active box to the values of the corresponding box style]

**BoxSaveLookAsStyle** (

Style:String;

Library:Enumeration {

CurrentDoc!;

PersonalLibrary!;

SharedLibrary!};)

[Save the settings for the active box as a style]

**BoxSizeNotLimited** (State:Enumeration {No!; Yes!})

[Specify if the box is limited by table cell boundaries]

**BoxStaysOnPage** (State:Enumeration {No!; Yes!})

[Specify whether a graphics box is allowed to move with the surrounding text]

**BoxStyleCopy** (

Style:Any {

FigureBox!;

TableBox!;

```

    TextBox!;
    UserBox!;
    EquationBox!;
    ButtonBox!;
    WatermarkImageBox!;
    InlineEquationBox!;
    NoBox!;
FromLibrary:Enumeration {
    CurrentDocument!;
    PersonalLibrary!;
    SharedLibrary!;
ToLibrary:Enumeration {
    CurrentDocument!;
    PersonalLibrary!;
    SharedLibrary!;
[DestName:String]

```

[Copy a box style from one library to another and/or rename the style]

```

BoxStyleCreate (
    Style:Any;
    [Library:Enumeration {
        CurrentDoc!;
        PersonalLibrary!;
        SharedLibrary!}]

```

[Create a box style and specify the library where it resides.]

```

BoxStyleDelete (
    Style:Any {
        FigureBox!;
        TableBox!;
        TextBox!;
        UserBox!;
        EquationBox!;
        ButtonBox!;
        WatermarkImageBox!;
        InlineEquationBox!;
        NoBox!;
    [Library:Enumeration {
        CurrentDoc!;
        PersonalLibrary!;
        SharedLibrary!}]

```

[Delete a user-defined box style]

```

BoxStyleEdit (
    Style:Any {
        FigureBox!;
        TableBox!;
        TextBox!;
        UserBox!;
        EquationBox!;
        ButtonBox!;
        WatermarkImageBox!;
        InlineEquationBox!;
        NoBox!;
    [Library:Enumeration {
        CurrentDoc!;
        PersonalLibrary!;
        SharedLibrary!}]

```

[Begin box style editing]

```

BoxStyleEnd (State:Enumeration {Cancel!; Save!})

```

[Save a new box style, or save changes to an existing style]

```

BoxStyleName (StyleName:String)

```

[Specify a box style name]

```

BoxStyleRetrieve (
    Filename:String;
    Library:Enumeration {
        CurrentDoc!;
        PersonalLibrary!;

```

```

    SharedLibrary!})

```

[Retrieve all box graphics styles from the specified document or template]

```

BoxStyleSave (
    Filename:String;
    Library:Enumeration {
        CurrentDoc!;
        PersonalLibrary!;
        SharedLibrary!})

```

[Save all styles in the current document or a specified style library to another style library]

```

BoxSuppress (State:Enumeration {No!; Yes!})

```

[Specify if the box is suppressed]

```

BoxTextAngle (Angle:Enumeration {
    None!;
    Degrees90!;
    Degrees180!;
    Degrees270!})

```

[Rotate text in a text box]

```

BoxTextDefaultStyle (Style:Any {
    NormalStyle!;
    FootnoteNumberInDocumentStyle!;
    EndnoteNumberInDocumentStyle!;
    FigureNumberStyle!;
    TableBoxNumberStyle!;
    TextBoxNumberStyle!;
    UserBoxNumberStyle!;
    EquationNumberStyle!;
    HypertextStyle!;
    ToC1Style!;
    ToC2Style!;
    ToC3Style!;
    ToC4Style!;
    ToC5Style!;
    Index1Style!;
    Index2Style!;
    ListStyle!;
    ToAStyle!;
    DocStyle!;
    BoxTextStyle!;
    CaptionStyle!;
    FootnoteStyle!;
    EndnoteStyle!;
    CommentStyle!;
    HeaderAStyle!;
    HeaderBStyle!;
    FooterAStyle!;
    FooterBStyle!;
    WatermarkAStyle!;
    WatermarkBStyle!;
    ParagraphOutInStyle!;
    OutlineStyle!;
    LegalOutInStyle!;
    BulletsOutInStyle!;
    HeadingsOutInStyle!;
    Legal_2OutInStyle!;
    NumbersOutInStyle!;
    Level1Style!;
    Level2Style!;
    Level3Style!;
    Level4Style!;
    Level5Style!;
    Level6Style!;
    Level7Style!;
    Level8Style!;
    Legal1Style!;
    Legal2Style!;
    Legal3Style!;

```

**Legal4Style!**;  
**Legal5Style!**;  
**Legal6Style!**;  
**Legal7Style!**;  
**Legal8Style!**;  
**Heading1Style!**;  
**Heading2Style!**;  
**Heading3Style!**;  
**Heading4Style!**;  
**Heading5Style!**;  
**Heading6Style!**;  
**Heading7Style!**;  
**Heading8Style!**;  
**MonospacedStyle!**;  
**CustomHTMLStyle!**;  
**PreformattedStyle!**;  
**AddressStyle!**;  
**QuotationsOutlnStyle!**;  
**DefinitionsOutlnStyle!**;  
**Quotation1Style!**;  
**Quotation2Style!**;  
**Quotation3Style!**;  
**Quotation4Style!**;  
**Quotation5Style!**;  
**Quotation6Style!**;  
**Quotation7Style!**;  
**Quotation8Style!**;  
**Definition1Style!**;  
**Definition2Style!**;  
**Definition3Style!**;  
**Definition4Style!**;  
**Definition5Style!**;  
**Definition6Style!**;  
**Definition7Style!**;  
**Definition8Style!**;  
**Level9Style!**;  
**Legal9Style!**;  
**Heading9Style!**;  
**Quotation9Style!**;  
**Definition9Style!**

[Specify the default text box style]

**BoxTextEdit ()**

[Begin text box editing]

**BoxTextEditEnd ()**

[End text box editing]

**BoxTextFilename (Filename:String)**

[Specify a file to insert into a text box]

**BoxTextFlow (Wrap:Enumeration {**

**LargestSide!**;  
**LeftSide!**;  
**RightSide!**;  
**NeitherSide!**;  
**BothSides!**;  
**Columnar!**;  
**NoWrapping!})**

[Specify the side(s) of a box to wrap text around]

**BoxTextFlowContour ()**

[Contour the surrounding text to the shape of graphics image]

**BoxTextFlowSquare ()**

[Contour the text evenly around all the sides of a box]

**BoxToBack ()**

[Move the selected object(s) to the back layer of the document]

**BoxToFront ()**

[Move the selected object(s) to the front layer of the document]

**BoxUpdateDisplay ()**

[Update the appearance of a box after editing changes]

**BoxVerticalAlignment (**

**Alignment:Enumeration {**  
**Top!**;  
**Center!**;  
**Bottom!**;  
**Baseline!**;  
**FullAlign!**;  
**BaseOffset!**;  
**ForceLeftMargin!**;  
**ForceRightMargin!**;  
**CenteredOnBaseline!**;  
**BottomOnBottom!};**

**[Offset:Any]**

[Align a box with the page or character it is attached to]

**BoxVerticalPosition (Alignment:Any)**

[Specify the vertical position of a box attached to a page or paragraph]

**BoxWidth (Width:Any {**  
**AutoWidth!})**

[Base the box width on the width of box contents or specify a fixed measurement]

**BoxWrapDlg ()**

[Specify how text wraps around graphics boxes]

**BrowsePrevious ()**

[Browse to the previous object]

**BrowseNext ()**

[Browse to the next object]

**BulletDlg ()**

[Create a bulleted list, numbered list or outline]

**BulletSetCounter (CounterName:String; Value:Numeric)**

[Specify a bullet style name and counter value]

**ButtonBarAddItem (**

**ButtonBarName:String;**  
**[Template:Enumeration {**  
**CurrentDoc!**;  
**PersonalLibrary!**;  
**SharedLibrary!});**  
**[ItemName:String];**  
**[Offset:Numeric];**  
**Type:Enumeration {**  
**Macro!**;  
**Separator!**;  
**Feature!**;  
**Script!**;  
**Launch!};**  
**[MacroName:String];**  
**[ItemText:String];**  
**[Prompt:String];**  
**[HelpFile:String];**  
**[HelpId:String];**  
**[Script:String])**

[Add items such as macros, buttons, and separators to a toolbar]

**ButtonBarCreateDlg ()**

[Create a toolbar]

**ButtonBarDeleteItem (**

**ButtonBarName:String;**  
**[Template:Enumeration {**  
**CurrentDoc!**;  
**PersonalLibrary!**;  
**SharedLibrary!});**

**[ItemName:String];**  
**[Offset:Numeric]**  
[Remove an item from a toolbar]

**BarButtonEditDlg ()**  
[Change the items on the toolbar]

**BarButtonOptionsDlg ()**  
[Change the position or style of the active toolbar]

**BarButtonScrollbarToggle (**  
**[State:Enumeration {Off!; On!}];**  
**BarButtonName:String;**  
**LoadedFromDefaultTemplate:Enumeration { No!; Yes!})**  
[Turn the toolbar scroll bar on or off]

**BarButtonSelect (**  
**BarButtonName:String;**  
**LoadedFromDefaultTemplate:Enumeration { No!; Yes!})**  
[Select a toolbar]

**BarButtonSetFont (Font:String; Size:Numeric)**  
[Specify a toolbar font face and size]

**BarButtonSetLocation (**  
**Location:Enumeration {**  
**Palette!;**  
**Left!;**  
**Top!;**  
**Right!;**  
**Bottom!};**  
**BarButtonName:String;**  
**LoadedFromDefaultTemplate:Enumeration {No!; Yes!})**  
[Specify a toolbar position]

**BarButtonSetRows (**  
**Rows:Numeric;**  
**BarButtonName:String;**  
**LoadedFromDefaultTemplate:Enumeration {No!; Yes!})**  
[Specify the number of rows on a toolbar]

**BarButtonSetStyle (**  
**Style:Enumeration {**  
**Picture!;**  
**Text!;**  
**PictureAndText!};**  
**BarButtonName:String;**  
**LoadedFromDefaultTemplate:Enumeration {No!; Yes!})**  
[Specify a toolbar style]

**BarButtonShow (**  
**State:Enumeration {Off!; On!};**  
**BarButtonName:String;**  
**LoadedFromDefaultTemplate:Enumeration {No!; Yes!})**  
[Display the toolbar]

**BarButtonShowToggle ()**  
[Toggle the toolbars]

**- C -**

**CapBeginSentence (State:Enumeration {Off!;On!})**  
[Automatically capitalize the first letter of each sentence as it is typed]

**CaseToggle ()**  
[Change the selected text to uppercase or lowercase]

**Center ()**  
[Center the selected text, the current line only, or all text from the insertion point forward]

**CenterCurrentPage (State:Enumeration {Off!; On!})**  
[Center the current page vertically]

**CenterPageDlg ()**  
[Center the text vertically on the current page or on the current and succeeding pages]

**CenterPages (State:Enumeration {Off!; On!})**  
[Center the active and succeeding pages vertically]

**ChangeWatermarkGraphicShade (Amount:Numeric)**  
[Change the shading of a Watermark image]

**ChangeWatermarkTextShade (Amount:Numeric)**  
[Change the shading of Watermark text]

**ChapterNumber (Chapter:Numeric)**  
[Specify a chapter number]

**ChapterNumberDecrement ()**  
[Decrease the chapter number by one]

**ChapterNumberDisplay ()**  
[Insert the chapter number at the insertion point]

**ChapterNumberIncrement ()**  
[Increase the chapter number by one]

**ChapterNumberMethod (Method:Enumeration {**  
**Numbers!;**  
**LowerLetters!;**  
**UpperLetters!;**  
**LowerRoman!;**  
**UpperRoman!})**  
[Specify a chapter numbering method]

**Char (Character:Variable;[Prompt:String])**  
[Prompt the user for a keystroke, then assign the ASCII equivalent to a variable]

**CharMappingDlg ()**  
[Select a character map]

**CheckBoxShippingMacro ()**  
[Insert a checkbox at the insertion point]

**CleanUpSpaces (State:Enumeration {Off!;On!})**  
[Turn on Cut and Paste Cleanup]

**ClearDoc ()**  
[Close the active document without saving but leave the active window open]

**ClearUndoLevels ()**  
[Delete all Undo actions performed on the active document]

**ClipboardAppendTo ()**  
[Add the active document to the Clipboard]

**ClipboardRetrieve ()**  
[Insert the Clipboard contents at the insertion point]

**ClipboardSaveTo ()**  
[Save the selected text or graphics to the Clipboard]

**Close ([Save:Enumeration {No!; Yes!}])**  
[Close File]

**CloseAllShippingMacro ()**  
[Close all open documents; prompt to save changes]

**CloseNoSave ([Verify:Enumeration {No!; Yes!}])**  
[Close the document window without saving changes]

**CodePage (**  
**[Dos:Numeric];**  
**[Windows:Numeric];**

**[InputFile:Numeric];**  
**[OutputFile:Numeric]**  
[Specify code page settings]

**CodePageDlg ()**  
[Display the Code Page Settings dialog box]

**ColumnBorderDlg ()**  
[Select, create, or edit a column border and background]

**ColumnDefineDlg ()**  
[Define, create, and edit text columns and columns border/fill]

**ColumnsDefinition (**  
    **ColumnsType:Enumeration {**  
        **Newspaper!;**  
        **NewspaperBalanced!;**  
        **Paralle!;**  
        **ParallelBlockProtect!;**  
    **VerticalSpacing:Numeric;**  
    **{{Spacing:Numeric Measurement};**  
    **[SpacingDef:Enumeration {NotFixed!; Fixed!}]}}**  
[Define columns]

**ColumnsOff ()**  
[Discontinue Columns]

**ColumnsOn ()**  
[Turn on Columns]

**CommentConvert ()**  
[Convert a comment to text]

**CommentCreate ()**  
[Create a comment]

**CommentEdit ()**  
[Edit a comment]

**CommentInformation ()**  
[Display the Comment Information dialog box]

**CommentInsertTime ()**  
[Insert the current time into a comment]

**CommentInsertUserInitials ()**  
[Insert the user initials specified in Environment Settings into a comment]

**CommentInsertUserName ()**  
[Insert the user name specified in Environment Settings into a comment]

**CommentNext ()**  
[Go to the next comment]

**CommentPrevious ()**  
[Go to the previous comment]

**CommentsDisplaySetup ([OnOff:Enumeration {Off!; On!}]**  
[Turn document comment display on or off]

**ConditionalEndOfPage (LinesTogether:Numeric)**  
[Keep a specified number of lines together at a page break]

**ConvertCaseInitialCaps ()**  
[Capitalize the current word or each word in selected text]

**ConvertCaseLowercase ()**  
[Lowercase the current word or selected text]

**ConvertCaseUppercase ()**  
[Uppercase the current word or selected text]

**Copy ()**  
[Copy the selected text]

**CopyAndPaste ()**  
[Copy the selected text to the Clipboard]

**CoreVersionArchiveCurrentDlg ()**  
[Save a version of the active document]

**CoreVersionRetrieveCurrentDlg ()**  
[Retrieve any version already saved for the active document]

**CoreVersionRetrieveDocumentDlg ()**  
[Retrieve any version already saved for any document]

**CounterCreate (**  
    **CounterName:Any {**  
        **None!;**  
        **FigureCounter!;**  
        **TableBoxCounter!;**  
        **TextBoxCounter!;**  
        **UserBoxCounter!;**  
        **EquationCounter!;**  
    **{{[Method:Enumeration {**  
        **Numbers!;**  
        **LowLetter!;**  
        **UpLetter!;**  
        **LowRoman!;**  
        **UpRoman!}]}}**  
[Specify a box reference number]

**CounterDecrement (**  
    **CounterName:Any {**  
        **None!;**  
        **FigureCounter!;**  
        **TableBoxCounter!;**  
        **TextBoxCounter!;**  
        **UserBoxCounter!;**  
        **EquationCounter!;**  
    **Level:Numeric)**  
[Decrease a counter level by one]

**CounterDelete (CounterName:Any {**  
    **None!;**  
    **FigureCounter!;**  
    **TableBoxCounter!;**  
    **TextBoxCounter!;**  
    **UserBoxCounter!;**  
    **EquationCounter!})**  
[Delete a user-defined counter or reset a program-defined counter]

**CounterDisplay (**  
    **CounterName:Any {**  
        **None!;**  
        **FigureCounter!;**  
        **TableBoxCounter!;**  
        **TextBoxCounter!;**  
        **UserBoxCounter!;**  
        **EquationCounter!;**  
    **Level:Numeric)**  
[Display the counter value of a box or user-defined counter]

**CounterDlg ()**  
[Create and edit counters]

**CounterEditLevel (**  
    **CounterName:Any {**  
        **None!;**  
        **FigureCounter!;**  
        **TableBoxCounter!;**  
        **TextBoxCounter!;**  
        **UserBoxCounter!;**  
        **EquationCounter!;**



**NewNumLvs:Numeric)**  
[Specify the number of levels a counter contains]

**CounterEditLevelMethod (**

**CounterName: Any {**  
None!;  
FigureCounter!;  
TableBoxCounter!;  
TextBoxCounter!;  
UserBoxCounter!;  
EquationCounter!};  
**LevelNum: Numeric;**  
**NewMethod: Enumeration {**  
Numbers!;  
LowLetter!;  
UpLetter!;  
LowRoman!;  
UpRoman!}}

[Change the numbering method for a multi-level counter]

**CounterEditLevelName (**

**OldCounterName: Any {**  
None!;  
FigureCounter!;  
TableBoxCounter!;  
TextBoxCounter!;  
UserBoxCounter!;  
EquationCounter!};  
**OldLevelName: String;**  
**NewLevelName: String)**

[Rename the specified level of a multi-level counter]

**CounterEditMethod (**

**CounterName: Any {**  
None!;  
FigureCounter!;  
TableBoxCounter!;  
TextBoxCounter!;  
UserBoxCounter!;  
EquationCounter!};  
**NewMethod: Enumeration {**  
Numbers!;  
LowLetter!;  
UpLetter!;  
LowRoman!;  
UpRoman!}}

[Change a counter numbering method]

**CounterEditName (**

**OldName: String;**  
**NewName: String)**

[Rename a user-defined counter]

**CounterIncrement (**

**CounterName: Any {**  
None!;  
FigureCounter!;  
TableBoxCounter!;  
TextBoxCounter!;  
UserBoxCounter!;  
EquationCounter!};  
**Level: Numeric)**

[Increase a counter level by one]

**CounterSetMethod (**

**CounterName: Any {**  
None!;  
FigureCounter!;  
TableBoxCounter!;  
TextBoxCounter!;  
UserBoxCounter!;  
EquationCounter!};  
**CounterLevel: Numeric;**

**Method: Enumeration {**

Numbers!;  
LowerLetters!;  
UpperLetters!;  
LowerRoman!;  
UpperRoman!}}

[Specify a counter numbering method]

**CounterSetValue (**

**CounterName: Any {**  
None!;  
FigureCounter!;  
TableBoxCounter!;  
TextBoxCounter!;  
UserBoxCounter!;  
EquationCounter!};  
**{[CounterLevel: Numeric]}**

[Set a counter value]

**CreateBoxToggle ([State: Enumeration {Off!; On!}])**

[Create and size boxes by clicking and dragging]

**CreateMapFile ([FileName: String])**

[Create a .MAP file containing the active document's character mapping and save it with the specified filename]

**CreateRegionLink (Region: Enumeration {OLERegion!; OBEXRegion!})**

[Indicate that a portion of the active document was copied to the Clipboard, and specify the link type]

**CrossRefMarkReference (**

**TargetName: String;**  
**RefType: Enumeration {**  
RefPage!;  
RefSecondaryPage!;  
RefChapter!;  
RefVolume!;  
RefParagraph!;  
RefFootnote!;  
RefEndnote!;  
RefBox!;  
RefCounter!};  
**[CounterName: Any {**  
None!;  
FigureCounter!;  
TableBoxCounter!;  
TextBoxCounter!;  
UserBoxCounter!;  
EquationCounter!}];  
**[QuickReference: Numeric]**

[Mark location where reference item will be displayed]

**CrossRefMarkTarget ([TagName: String])**

[Mark item to be referenced to]

**CtrlBreak ()**

[Cancel the current action]

**CustomBoxDlg ()**

[Create a custom box]

**Cut ()**

[Remove the selected text and copy it to the Clipboard]

**CutAndPaste ()**

[Remove the selected text and copy it to the Clipboard]

**- D -**

**DateCode ()**

[Insert a date code that changes to reflect the current date]

**DateFormat (DateFormatString:String)**

[Specify a date format]

**DateFormatDlg ()**

[Change the appearance of dates when they are inserted into a document]

**DateText ()**

[Insert the current date]

**DDEChangeUpdateMode (**

**LinkName:String;**  
**[UpdateMode:Enumeration {**  
**Manual!;**  
**Automatic!}]**)

[Specify a manual or automatic DDE link update method]

**DDECreateLink (**

**LinkName:String;**  
**Source:String;**  
**[UpdateMode:Enumeration {**  
**Manual!;**  
**Automatic!}]**;  
**[Format:Enumeration {**  
**WP6DOC!;**  
**WPDOC!;**  
**WP6TEXT!;**  
**WPG2!;**  
**RTF!;**  
**TEXT!;**  
**OEMTEXT!;**  
**METAFILE!;**  
**WB1!;**  
**WP7DOC!;**  
**WP7TEXT!;**  
**WP8DOC!;**  
**WP8TEXT!;**  
**WP9DOC!;**  
**WP9TEXT!}]**)

[Establish a link between a WordPerfect 10 document and a file in another application]

**DDECreateLinkDlg ()**

[Display the Links dialog box]

**DDEDeleteLink ({[LinkName:String]})**

[Remove a DDE Link]

**DDEDeleteLinkDlg ()**

[Display the Links dialog box]

**DDEEditLink (**

**LinkName:String;**  
**[NewName:String];**  
**[NewSource:String]**)

[Change the properties of a DDE Link]

**DDEEditLinkDlg ()**

[Display the Change Links dialog box]

**DDEPasteLink ({[Format:Enumeration {**

**WP6DOC!;**  
**WPDOC!;**  
**WP6TEXT!;**  
**WPG2!;**  
**RTF!;**  
**TEXT!;**  
**OEMTEXT!;**  
**METAFILE!;**  
**WB1!;**  
**WP7DOC!;**  
**WP7TEXT!;**  
**WP8DOC!;**

**WP8TEXT!;**  
**WP9DOC!;**  
**WP9TEXT!}]**)

[Establish a link between a WordPerfect 10 document and a file in another application]

**DDEUpdateLink ({[LinkName:String]})**

[Update DDE links]

**DDEUpdateLinkDlg ()**

[Display the Links dialog box]

**String := DecimalAlignmentCharacter ([Character:String])**

[Specify a decimal character]

**DelayCodes (PagesToDelay:Numeric)**

[Postpone the effects of codes for one or more pages]

**DelayCodesDlg ()**

[Delay features (codes) for one or more pages]

**DeleteAppend ()**

[Delete a selection and append it to the Clipboard]

**DeleteCharNext ()**

[Delete the character to the right of the insertion point]

**DeleteCharPrevious ()**

[Delete the character to the left of the insertion point]

**DeleteToBeginningOfWord ()**

[Delete the text from the insertion point to the beginning of the current word]

**DeleteToEndOfLine ()**

[Delete the text from the insertion point to the end of the line]

**DeleteToEndOfPage ()**

[Delete the text from the insertion point to the end of the page]

**DeleteToEndOfWord ()**

[Delete the text from the insertion point to the end of the current word]

**DeleteWord ()**

[Delete the word at the insertion point]

**DirectoryCreate (Directory:String)**

[Create a folder]

**DirectoryExists (Variable:Variable;Directory:String)**

[Return whether a specified folder exists]

**Boolean := Display ([State:Enumeration {Off!;On!}])**

[Display macro execution when a macro is played]

**DisplayMode (DisplayMode:Enumeration {**

**Text!;**  
**Graphics!;**  
**FullPage!;**  
**WebPage!}]**)

[Display the current document in Draft or Page mode]

**Boolean := DisplaySuppressedBoxes ([State:Enumeration {**

**No!;**

**Yes!}]**)

[Specify if suppressed boxes will display]

**DisplayZoom (Percent:Numeric)**

[Specify the current zoom setting]

**DocCompareAddMarkingsDlg ()**

[Compare two documents using Redline and Strikeout]

**DocCompareOptions** (  
 [Precision:Enumeration {Character!; Word!}];  
 [SkipChars:String];  
 [CmpHdrsFtrs:Enumeration {No!; Yes!}];  
 [CmpComments:Enumeration {No!; Yes!}];  
 [ExpandMstrDocs:Enumeration {No!; Yes!}];  
 [SummaryOfCmp:Enumeration {  
   None!;  
   AtBeginning!;  
   AtEnd!}];  
 [ListOfChanges:Enumeration {  
   None!;  
   ShowChgOnly!;  
   ShowContext!}];  
 [InsertionBoldAttr:Enumeration {Off!; On!}];  
 [InsertionItalicAttr:Enumeration {Off!; On!}];  
 [InsertionUndAttr:Enumeration {Off!; On!}];  
 [InsertionDUndAttr:Enumeration {Off!; On!}];  
 [InsertionRedlineAttr:Enumeration {Off!; On!}];  
 [InsColorRed:Numeric];  
 [InsColorGreen:Numeric];  
 [InsColorBlue:Numeric];  
 [EncloseInsWith:Enumeration {  
   None!;  
   Brackets!;  
   Braces!;  
   Bkslashes!;  
   Quotes!;  
   GTLT!}];  
 [DeletionBoldAttr:Enumeration {Off!; On!}];  
 [DeletionItalicAttr:Enumeration {Off!; On!}];  
 [DeletionUndAttr:Enumeration {Off!; On!}];  
 [DeletionDUndAttr:Enumeration {Off!; On!}];  
 [DeletionRedlineAttr:Enumeration {Off!; On!}];  
 [DeletionStkoutAttr:Enumeration {Off!; On!}];  
 [DelColorRed:Numeric];  
 [DelColorGreen:Numeric];  
 [DelColorBlue:Numeric];  
 [EncloseDelWith:Enumeration {  
   None!;  
   Brackets!;  
   Braces!;  
   Bkslashes!;  
   Quotes!;  
   GTLT!}];  
 [ShowDelTextAs:Enumeration {  
   FullText!;  
   AbbrevText!;  
   NoText!;  
   Character!}];  
 [DelTextChar:String];  
 [ShowMoves:Enumeration {  
   NotShown!;  
   AtNewLoc!;  
   AtBothLocs!}];  
 [MoveColorRed:Numeric];  
 [MoveColorGreen:Numeric];  
 [MoveColorBlue:Numeric];  
 [ReviewOpt:Enumeration {  
   CompareOnly!;  
   CompareThenReview!}])  
 [Specify options for the document compare]

**DocCompareRemoveMarkingsDlg** (  
 [Remove the markings used to compare documents]

**DocCompareRemoveRedline** (Flag:Enumeration {  
 RemoveAll!;  
 LeaveRedline!})  
 [Remove Strikeout and Redline codes and Strikeout text, or remove compare markings]

**DocInitialFont** (  
 Name:String;  
 [PointSize:Numeric Measurement];  
 [Family:Enumeration {  
   FamilyITCBenguiat!;  
   FamilyTimesRoman!;  
   FamilyITCGaramond!;  
   FamilyCenturySchoolbook!;  
   FamilyBodoni!;  
   FamilyITCLubalinGraph!;  
   FamilyAlbertus!;  
   FamilyCourier!;  
   FamilyHelvetica!;  
   FamilyITCRonda!;  
   FamilyMicrostyle!;  
   FamilyOptima!;  
   FamilyLetterGothic!;  
   FamilyBrush!;  
   FamilyParkAvenue!;  
   FamilySignetRoundhand!;  
   FamilyTekton!;  
   FamilyUncial!;  
   FamilyOldEnglish!;  
   FamilyITCZapfDingbats!;  
   FamilyBroadway!;  
   FamilyComputer!;  
   FamilyPTBarnum!;  
   FamilyPonderosa!;  
   FamilyMadrone!;  
   FamilyHobo!;  
   FamilyRevue!;  
   FamilyAestheticOrnamented!}];  
 [Attributes:Enumeration {  
   FontMatchNormal!;  
   FontMatchItalic!;  
   FontMatchOutline!;  
   FontMatchShadow!;  
   FontMatchSmallCaps!}];  
 [Weight:Enumeration {  
   WeightUnknown!;  
   WeightUltraThin!;  
   WeightUltraLight!;  
   WeightExtraThin!;  
   WeightThin!;  
   WeightExtraLight!;  
   WeightLight!;  
   WeightDemiLight!;  
   WeightSemiLight!;  
   WeightBook!;  
   WeightRegular!;  
   WeightRoman!;  
   WeightMedium!;  
   WeightDemiBold!;  
   WeightSemiBold!;  
   WeightBold!;  
   WeightExtraBold!;  
   WeightHeavy!;  
   WeightExtraHeavy!;  
   WeightBlack!;  
   WeightExtraBlack!;  
   WeightUltraHeavy!;  
   WeightUltraBlack!;  
   WeightMaximum!}];  
 [Width:Enumeration {  
   WidthUnknown!;  
   WidthUltraCondensed!;  
   WidthUltraCompressed!;  
   WidthCompressed!;  
   WidthExtraCondensed!;  
   WidthCondensed!;  
   WidthSemiCondensed!;  
   WidthNormal!};

```

WidthSemiExpanded!;
WidthExpanded!;
WidthExtraExpanded!;
WidthUltraExpanded!;
WidthDoubleWide!;
WidthTripleWide!});
[Source:Enumeration {
  PRSFile!;
  DRSFile!});
[Type:Enumeration {
  Speedo!;
  Intellifont!;
  Type1!;
  TrueType!});
[CharacterSet:Enumeration {
  FontMatchASCII!;
  FontMatchMultinational1!;
  FontMatchMultinational2!;
  FontMatchBoxDrawing!;
  FontMatchTypographicSymbols!;
  FontMatchIconicSymbols!;
  FontMatchMath!;
  FontMatchMathExtension!;
  FontMatchGreek!;
  FontMatchHebrew!;
  FontMatchCyrillic!;
  FontMatchJapaneseKana!;
  FontMatchUserDefined!;
  FontMatchArabic!;
  FontMatchArabicScript!});
[Set the default font for the active document]

```

**DocInitialFontDlg ()**  
[Specify the default font for the active document]

**DocMaximize ()**  
[Expand the document window to its maximum size]

**DocMinimize ()**  
[Reduce the document window to an icon]

**DocMove ()**  
[Move the document window to another location on the screen]

**DocNext ()**  
[Edit the next open document]

**DocPrevious ()**  
[Edit the previous open document]

**DocRedlineMethodDlg ()**  
[Change the appearance of redlined text]

**DocRestore ()**  
[Restore the document window to its previous size]

**DocSize ()**  
[Change the size of the document window]

```

String := DocSumGet (DocSumFieldName:Enumeration {
  Abstract!;
  Account!;
  Address!;
  Attachments!;
  Author!;
  Authorization!;
  BillTo!;
  BlindCopy!;
  CarbonCopy!;
  Category!;
  CheckedBy!;
  Client!;
  Comments!;

```

```

CreationDate!;
DateCompleted!;
Department!;
DescriptiveName!;
DescriptiveType!;
Destination!;
Disposition!;
Division!;
DocumentNumber!;
Editor!;
ForwardTo!;
Group!;
Keywords!;
Language!;
MailStop!;
Matter!;
Office!;
Owner!;
Project!;
Publisher!;
Purpose!;
ReceivedFrom!;
RecordedBy!;
RecordedDate!;
Reference!;
RevisionDate!;
RevisionNotes!;
RevisionNumber!;
Section!;
Security!;
Source!;
Status!;
Subject!;
TelephoneNumber!;
Typist!;
VersionDate!;
VersionNotes!;
VersionNumber!});
[Returns content of the specified document summary field]

```

```

DocSummaryConfig ({{DocSumFieldName:Enumeration {
  Abstract!;
  Account!;
  Address!;
  Attachments!;
  Author!;
  Authorization!;
  BillTo!;
  BlindCopy!;
  CarbonCopy!;
  Category!;
  CheckedBy!;
  Client!;
  Comments!;
  CreationDate!;
  DateCompleted!;
  Department!;
  DescriptiveName!;
  DescriptiveType!;
  Destination!;
  Disposition!;
  Division!;
  DocumentNumber!;
  Editor!;
  ForwardTo!;
  Group!;
  Keywords!;
  Language!;
  MailStop!;
  Matter!;
  Office!;
  Owner!;

```

Project!;  
 Publisher!;  
 Purpose!;  
 ReceivedFrom!;  
 RecordedBy!;  
 RecordedDate!;  
 Reference!;  
 RevisionDate!;  
 RevisionNotes!;  
 RevisionNumber!;  
 Section!;  
 Security!;  
 Source!;  
 Status!;  
 Subject!;  
 TelephoneNumber!;  
 Typist!;  
 VersionDate!;  
 VersionNotes!;  
 VersionNumber!}})

[Specify the fields to include in the active document summary]

**DocSummaryDelete ()**

[Delete the active document summary information]

**DocSummaryPrint ()**

[Print the active document summary]

**DocSummarySaveAs (**

Filename:String;  
 [ExportType:Enumeration {  
 WordPerfect\_42!;  
 WordPerfect\_50!;  
 WordPerfect\_51!;  
 WordPerfect\_60!;  
 WordPerfect\_7!;  
 WordStar\_33!;  
 WordStar\_331!;  
 WordStar\_34!;  
 WordStar\_40!;  
 WordStar\_50!;  
 WordStar\_55!;  
 WordStar\_60!;  
 WordStar2000\_10!;  
 WordStar2000\_20!;  
 WordStar2000\_30!;  
 MicrosoftWord\_40!;  
 MicrosoftWord\_50!;  
 MicrosoftWord\_55!;  
 WordForWindows\_10!;  
 WordForWindows\_11!;  
 WordForWindows\_1A!;  
 WordForWindows\_20!;  
 WordForWindows\_20a!;  
 WordForWindows\_20b!;  
 RFT!;  
 FFT!;  
 DisplayWrite\_40!;  
 DisplayWrite\_42!;  
 DisplayWrite\_50!;  
 MultiMate\_33!;  
 MultiMateAdvantage\_36!;  
 MultiMateAdvantage\_10!;  
 MultiMate\_40!;  
 OfficeWriter\_60!;  
 OfficeWriter\_61!;  
 OfficeWriter\_611!;  
 OfficeWriter\_62!;  
 RTF!;  
 ASCIIStandard!;  
 ANSIText\_Windows!;  
 DelimitedText!;

ASCIIStripped!;  
 Kermit\_7bit\_Transfer!;  
 NavyDIF!;  
 XyWriteIIPlus\_355!;  
 XyWriteIIPlus\_356!;  
 AmiPro\_10!;  
 AmiPro\_12!;  
 AmiPro\_12A!;  
 AmiPro\_12B!;  
 AmiPro\_30!;  
 WindowsWrite!;  
 WordPerfect\_5xA!;  
 WordForWindows\_12!;  
 WordForWindows\_50!;  
 RTFJ!;  
 ANSIDelimitedTextWindows!;  
 ANSIGenericWordProcessor\_Windows!;  
 Excel\_40!});  
 [AppendReplace:Enumeration {  
 Append!;  
 Replace!})]

[Save the active document summary with a filename]

**DocSumSet (**

{[DocSumFieldName:Enumeration {  
 Abstract!;  
 Account!;  
 Address!;  
 Attachments!;  
 Author!;  
 Authorization!;  
 BillTo!;  
 BlindCopy!;  
 CarbonCopy!;  
 Category!;  
 CheckedBy!;  
 Client!;  
 Comments!;  
 CreationDate!;  
 DateCompleted!;  
 Department!;  
 DescriptiveName!;  
 DescriptiveType!;  
 Destination!;  
 Disposition!;  
 Division!;  
 DocumentNumber!;  
 Editor!;  
 ForwardTo!;  
 Group!;  
 Keywords!;  
 Language!;  
 MailStop!;  
 Matter!;  
 Office!;  
 Owner!;  
 Project!;  
 Publisher!;  
 Purpose!;  
 ReceivedFrom!;  
 RecordedBy!;  
 RecordedDate!;  
 Reference!;  
 RevisionDate!;  
 RevisionNotes!;  
 RevisionNumber!;  
 Section!;  
 Security!;  
 Source!;  
 Status!;  
 Subject!;  
 TelephoneNumber!;

**Typist!;**  
**VersionDate!;**  
**VersionNotes!;**  
**VersionNumber!;};**  
**[DocSumFieldData:String]})**  
[Modify the data of the specified document summary field]

**DocumentCompare (**  
**Filename:String;**  
**MarkIn:Enumeration {**  
**NewDoc!;**  
**CurrentDoc!})**  
[Compare two versions of a document and mark differences]

**Numeric := DocumentInfo ([ItemType:Enumeration {**  
**Characters!;**  
**Words!;**  
**Lines!;**  
**Sentences!;**  
**Paragraphs!;**  
**Pages!;**  
**AveWordLength!;**  
**AveNumOfWords!;**  
**MaxNumOfWords!;})**  
[Returns document information such as number of characters, paragraphs, etc.]

**DotLeader (Character:String; SpaceCount:Numeric)**  
[Specify a dot leader character, and the number of spaces between each character]

**DoubleSidedPrint (Flags:Enumeration {**  
**None!;**  
**LongEdge!;**  
**ShortEdge!})**  
[Turn double-sided printing (duplex) on or off, and specify a binding edge]

**DoubleSmartQuote (**  
**[State:Enumeration {Off!; On!};**  
**[OpenDoubleQuote:Numeric];**  
**[CloseDoubleQuote:Numeric])**  
[Turn Double SmartQuotes on or off and specify the quote character]

**DraftZoom100 ()**  
[Display the document without headers, footnotes, or other substructures, at 100%]

**DraftZoomMargin ()**  
[Display the document without headers, footnotes, or other substructures, at margin width]

**DropCapAdjustDlg ()**  
[Specify Drop Cap options, such as how many characters to include]

**DropCapBegin ()**  
[Begin a drop cap]

**DropCapStart ()**  
[Start a drop cap definition]

**DropCapBorderFillDlg ()**  
[Specify border and fill styles for drop caps]

**DropCapBorderStyle (BorderStyleName:Any {**  
**SpacingOnly!;**  
**SingleBorder!;**  
**DoubleBorder!;**  
**DashedBorder!;**  
**DottedBorder!;**  
**ThickBorder!;**  
**ExtraThickBorder!;**  
**ThinThickBorder!;**

**ThickThinBorder!;**  
**ThickTopBottomBorder!;**  
**ButtonBorder!;**  
**ColumnBorderBetween!;**  
**ColumnBorderAll!;**  
**HeavySingleBorder!;**  
**HeavyDoubleBorder!;**  
**NoBorder!})**  
[Specify a drop cap border style]

**DropCapCharacters (Number:Numeric)**  
[Specify the number of characters in a drop cap]

**DropCapCharacterSize (Size:Numeric)**  
[Specify the size of drop cap character(s)]

**DropCapContourWrap (State:Any {No!; Yes!})**  
[Wrap text around a drop cap]

**DropCapControlBar ()**  
[Change the first letter in a paragraph to a drop cap]

**DropCapDialog ([Page:Enumeration {**  
**Format!;**  
**Font!;**  
**Options!;})**  
[Change the first letter in a paragraph to a drop cap]

**DropCapEnd (State:Enumeration {Cancel!; Save!})**  
[End a drop cap and save the changes]

**DropCapFillStyle (FillStyleName:Any {**  
**Fill10!;**  
**Fill20!;**  
**Fill30!;**  
**Fill40!;**  
**Fill50!;**  
**Fill60!;**  
**Fill70!;**  
**Fill80!;**  
**Fill90!;**  
**Fill100!;**  
**FillButton!;**  
**Fill5!;**  
**NoFill!})**  
[Specify a drop cap fill style]

**DropCapFontDlg ()**  
[Specify the font for drop caps]

**DropCapHorizontalAdjustment (Adjustment:Numeric)**  
[Specify what percentage of a drop cap is located in the margin]

**DropCapIncludeDescender (State:Any {No!; Yes!})**  
[Allow room for descending characters in a drop cap]

**DropCapIncludeDiacritic (State:Enumeration {No!; Yes!})**  
[Allow room for diacritic marks in a drop cap]

**DropCapKerning (Adjustment:Numeric Measurement)**  
[Specify the spacing between drop cap characters]

**DropCapNoWrap (State:Any {No!; Yes!})**  
[Specify whether text wraps around a drop cap]

**DropCapOff ()**  
[Turn off the Drop Cap in the current paragraph]

**DropCapPositionOtherDlg ()**  
[Specify the vertical and horizontal position of drop caps]

**DropCapSizeDlg ()**  
[Specify the size of drop caps]

**DropCapVerticalAdjustment (Adjustment:Numeric)**

[Specify the number of blank lines between a drop cap and the text that follows it]

- E -

**EditBoxDlg ()**

[Change box appearance, position, and content]

**EditBoxDlgClose ()**

[Close the Edit Box Dialog]

**EditCopy ()**

[Copy the selection to the Clipboard]

**EditCut ()**

[Move the selection to the Clipboard]

**EditPaste (**

**[ClipboardFormat:String];**  
**[FormatId:Numeric];**  
**[LinkType:Enumeration {**  
**DDE!;**  
**OLE!}]])**

[Insert the Clipboard contents at the insertion point]

**EditPasteSimple ()**

[Insert the Clipboard contents at the insertion point]

**EndCenterOrAlignment ()**

[Turn off Center, Flush Right, and alignment tabs at the insertion point for the active line only]

**EndnoteCreate ()**

[Create an endnote]

**EndnoteEdit ([EndnoteNumber:String])**

[Open a specific endnote window]

**EndnoteEditDlg ()**

[Edit a specified endnote]

**EndnoteFootnoteConvert (Action:Enumeration {**

**FootnoteToEndnote!;**  
**EndnoteToFootnote!})**

[Convert footnotes to endnotes in the active document]

**EndnoteFootnoteNext ()**

[Go to the next endnote/footnote]

**EndnoteFootnotePrev ()**

[Go to the previous endnote/footnote]

**Numeric Measurement := EndnoteMinimum ([Size:Numeric Measurement])**

[Specify the amount of endnote text to keep together]

**EndnoteNewNumber (EndnoteNumber:String)**

[Begin new numbering for subsequent endnotes]

**EndnoteNumberDecrement ()**

[Decrease the active endnote number by one]

**EndnoteNumberDisplay ()**

[Display the active endnote number]

**EndnoteNumberIncrement ()**

[Increase the active endnote number by one]

**EndnoteNumberMethod (**

**NumberingMethod:Enumeration {**  
**Numbers!;**  
**LowerLetters!;**  
**UpperLetters!;**

**LowerRoman!;**

**UpperRoman!;**

**Characters!);**

**[Characters:String])**

[Specify an endnote numbering method]

**EndnoteOptions (**

**[NumberingMethod:Enumeration {**  
**Numbers!;**  
**LowerLetters!;**  
**UpperLetters!;**  
**LowerRoman!;**  
**UpperRoman!;**  
**Characters!});**

**[Characters:String];**

**[BetweenSpacing:Numeric Measurement];**

**[MinimumHeight:Numeric Measurement])**

[Specify endnote options beginning at the insertion point]

**EndnoteOptionsDlg ()**

[Change the endnote numbering method, format, and style]

**EndnotePlacement ([Restart:Enumeration {No!; Yes!}])**

[Mark where to generate an endnote list, and reset numbering for subsequent endnotes]

**EndnotePlacementDlg ()**

[Place the endnote text at the insertion point]

**EndnoteSetNumberDlg ()**

[Start the numbering of subsequent endnotes with a new number]

**Numeric Measurement := EndnotesSpaceBetween ([Width:Numeric Measurement])**

[Specify the distance between endnotes]

**EndnoteToFootnoteShippingMacro ()**

[Convert endnotes to footnotes in the active document or selection]

**EndOfSentenceSpacing (Type:Enumeration {**

**None!;**  
**SingleToDouble!;**  
**DoubleToSingle!})**

[Specify the number of spaces at the end of a sentence]

**EnterKey ()**

[Insert a hard return, perform a hyperlink jump, etc.]

**EnvelopeAddressPositionUpdate ()**

[Update the envelope address positions]

**EnvelopeCreate (**

**[Name:String];**  
**[MergeEnvelope:Enumeration {**  
**CreateMrgEnvelope!;**  
**EditMrgEnvelope!})])**

[Create an envelope]

**EnvelopeDlg ()**

[Create an envelope]

**EnvelopePositionsDlg ()**

[Modify return and mailing address position]

**EnvelopeSavePaperSize ()**

[Save the Envelope Paper Size]

**PrintEnvelope ()**

[Print Current Envelope]

**EquationBaseFontDlg ()**

[Select the font for the Equation Editor]

**EquationCreate ()**  
[Create an equation]

**EquationPalette ([State:Enumeration {Off!; On!}])**  
[Display the Equation Palette in the Equation Editor]

**EquationPaletteToggle ()**  
[Toggle the equation palette]

**EquationRetrieveDlg ()**  
[Retrieve an existing equation]

**EquationZoomDlg ()**  
[Change the magnification of the equation window]

**ExecScrapBook ()**  
[Insert a Scrapbook item into your document]

**ExitWordPerfect ()**  
[Exit WordPerfect 10]

**ExpandAllAbbrevShippingMacro ()**  
[Expand all QuickWords in the document]

**ExpandDlg ()**  
[Select the subdocuments to open in a master document]

**- F -**

**FancyPageBorder (Filename:String)**  
[Create a border from a predefined graphic]

**Boolean := FancyPageBorderInParagraph ()**  
[Return whether a Fancy Border code is in the current paragraph]

**Boolean := FancyPageBorderOccur ()**  
[Return whether fancy page borders are active]

**FancyPageBorderOff ()**  
[Turn Fancy Page Border off]

**FigurePaletteDlg ()**  
[Edit the current image]

**FigurePaletteDlgEnd ()**  
[Close the Image Tools palette]

**FigureRetrieveDlg ()**  
[Put a clipart or graphics image in your document]

**FileChangeAttributes (**  
    **Filename:String;**  
    **{[Attributes:Enumeration {**  
        **Normal!;**  
        **ReadOnly!;**  
        **Hidden!;**  
        **System!;**  
        **Archive!}]})**  
[Change file attributes.]

**FileChangeDir (Directory:String)**  
[Change the Documents or Graphics files folder]

**FileCopy (**  
    **SourceFile:String;**  
    **DestinationFile:String)**  
[Make a copy of the active file]

**FileDelete ([Filename:String])**  
[Delete a specified file]

**FileExists (**  
    **Variable:Variable;**

**Filename:String)**  
[Return whether a specified file exists]

**FileInsert (**  
    **[Filename:String];**  
    **[AutoDetect:Enumeration {No!; Yes!}];**  
    **[InsertIntoDoc:Enumeration {Prompt!; Insert!}])**  
[Insert a specified file]

**FileInsertDlg ()**  
[Retrieve an existing file]

**FileMove ([SourceFile:String]; [DestinationFile:String])**  
[Rename and move a file]

**FileNew ()**  
[Create a blank document in a new window]

**FileOpen (**  
    **Filename:String;**  
    **[Format:Enumeration {**  
        **WordPerfect\_42!;**  
        **WordPerfect\_50!;**  
        **WordPerfect\_51!;**  
        **WordPerfect\_6\_7\_8!;**  
        **WordPerfect\_60!;**  
        **WordPerfect\_90!;**  
        **WordPerfect\_CompoundFile!;**  
        **WordPerfect\_7!;**  
        **WordStar\_33!;**  
        **WordStar\_331!;**  
        **WordStar\_34!;**  
        **WordStar\_40!;**  
        **WordStar\_50!;**  
        **WordStar\_55!;**  
        **WordStar\_60!;**  
        **WordStar2000\_10!;**  
        **WordStar2000\_20!;**  
        **WordStar2000\_30!;**  
        **MicroSoftWord\_40!;**  
        **MicroSoftWord\_50!;**  
        **MicroSoftWord\_55!;**  
        **WordForWindows\_10!;**  
        **WordForWindows\_11!;**  
        **WordForWindows\_1A!;**  
        **WordForWindows\_20!;**  
        **WordForWindows\_20a!;**  
        **WordForWindows\_20b!;**  
        **WordForWindows\_20c!;**  
        **WordForWindows\_60!;**  
        **RFT!;**  
        **FFT!;**  
        **DisplayWrite\_40!;**  
        **DisplayWrite\_42!;**  
        **DisplayWrite\_50!;**  
        **MultiMate\_33!;**  
        **MultiMateAdvantage\_36!;**  
        **MultiMateAdvantagell\_10!;**  
        **MultiMate\_40!;**  
        **OfficeWriter\_60!;**  
        **OfficeWriter\_61!;**  
        **OfficeWriter\_611!;**  
        **OfficeWriter\_62!;**  
        **RTF!;**  
        **ASCIIStandard!;**  
        **ANSIText\_Windows!;**  
        **DelimitedText!;**  
        **ASCIIstripped!;**  
        **Kermit\_7bit\_Transfer!;**  
        **NavyDIF!;**  
        **XyWritellIPlus\_355!;**  
        **XyWritellIPlus\_356!;**



AmiPro\_12!;  
 AmiPro\_12A!;  
 AmiPro\_12B!;  
 AmiPro\_20!;  
 AmiPro\_30!;  
 WindowsWrite!;  
 XML\_UTF16Big!;  
 XML\_UTF16Lit!;  
 XML\_UTF8!;  
 HTML!;  
 SGML!;  
 WordPerfect\_5xA!;  
 WordForWindows\_12!;  
 WordForWindows\_50!;  
 AreAHangoll!;  
 Hana!;  
 WPS\_Chinese!;  
 WaiSing\_251!;  
 RTFJ!;  
 Ichitaro\_3x!;  
 Ichitaro\_4x!;  
 Ichitaro\_5x!;  
 P1EXE!;  
 Aruga\_23!;  
 Matu\_4!;  
 Matu\_5!;  
 Matu\_6!;  
 ANSIDelimitedTextWindows!;  
 ANSIGenericWordProcessor\_Windows!;  
 Excel\_40!}}))

[Retrieve a specified file into a new document window]

#### **FileOpenDlg ()**

[Open an existing document into a new window]

#### **FileRemoveDirectory (Directory:String)**

[Remove a folder]

#### **FileSave (**

[Filename:String];  
 [ExportType:Enumeration {  
   WordPerfect\_42!;  
   WordPerfect\_50!;  
   WordPerfect\_51!;  
   WordPerfect\_6\_7\_8!;  
   WordPerfect\_60!;  
   WordPerfect\_90!;  
   WordPerfect\_CompoundFile!;  
   WordPerfect\_7!;  
   WordStar\_33!;  
   WordStar\_331!;  
   WordStar\_34!;  
   WordStar\_40!;  
   WordStar\_50!;  
   WordStar\_55!;  
   WordStar\_60!;  
   WordStar2000\_10!;  
   WordStar2000\_20!;  
   WordStar2000\_30!;  
   MicroSoftWord\_40!;  
   MicroSoftWord\_50!;  
   MicroSoftWord\_55!;  
   WordForWindows\_10!;  
   WordForWindows\_11!;  
   WordForWindows\_1A!;  
   WordForWindows\_20!;  
   WordForWindows\_20a!;  
   WordForWindows\_20b!;  
   WordForWindows\_20c!;  
   WordForWindows\_60!;  
   RFT!;  
   FFT!;

DisplayWrite\_40!;  
 DisplayWrite\_42!;  
 DisplayWrite\_50!;  
 MultiMate\_33!;  
 MultiMateAdvantage\_36!;  
 MultiMateAdvantagell\_10!;  
 MultiMate\_40!;  
 OfficeWriter\_60!;  
 OfficeWriter\_61!;  
 OfficeWriter\_611!;  
 OfficeWriter\_62!;  
 RTF!;  
 ASCIIStandard!;  
 ANSIText\_Windows!;  
 DelimitedText!;  
 ASCIIStripped!;  
 Kermit\_7bit\_Transfer!;  
 NavyDIF!;  
 XyWriteIIPlus\_355!;  
 XyWriteIIPlus\_356!;  
 AmiPro\_12!;  
 AmiPro\_12A!;  
 AmiPro\_12B!;  
 AmiPro\_20!;  
 AmiPro\_30!;  
 WindowsWrite!;  
 XML\_UTF16Big!;  
 XML\_UTF16Lit!;  
 XML\_UTF8!;  
 EDGAR!;  
 HTML!;  
 SGML!;  
 WordPerfect\_5xA!;  
 WordForWindows\_12!;  
 WordForWindows\_50!;  
 AreAHangoll!;  
 Hana!;  
 WPS\_Chinese!;  
 WaiSing\_251!;  
 RTFJ!;  
 Ichitaro\_3x!;  
 Ichitaro\_4x!;  
 Ichitaro\_5x!;  
 P1EXE!;  
 Aruga\_23!;  
 Matu\_4!;  
 Matu\_5!;  
 Matu\_6!;  
 ANSIDelimitedTextWindows!;  
 ANSIGeneratedWordProcessor\_Windows!;  
 Excel\_40!;  
 QuattroPro\_30!;  
 QuattroPro\_40!}});  
 [Overwrite:Enumeration {No!; Yes!; Prompt!}]

[Save the current document]

#### **FileSaveAsDlg ()**

[Save file]

#### **FileStampShippingMacro ()**

[Insert the filename of the current document in a header or footer]

#### **FillColor (**

[ForeColorName:String];  
 [ForeColor:Numeric];  
 [ForeGreen:Numeric];  
 [ForeBlue:Numeric];  
 [ForeShade:Numeric];  
 [BackColorName:String];  
 [BackRed:Numeric];  
 [BackGreen:Numeric];  
 [BackBlue:Numeric];

**[BackShade:Numeric]**  
[Specify a fill color for boxes and tables]

**FillSetGradient (**  
  **Gradient:Enumeration {**  
    **LinearGradient!;**  
    **RadialGradient!;**  
    **RectangularGradient!};**  
  **[Angle:Numeric];**  
  **[XLocation:Numeric];**  
  **[YLocation:Numeric];**  
  **[Calculate:Numeric];**  
  **[State:Enumeration {No!; Yes!}]]**  
[Customize a gradient fill]

**FillSetWPG2 (**  
  **PatternSet:Numeric;**  
  **Index:Enumeration {**  
    **WPG2\_Solid!;**  
    **WPG2\_50Gray!;**  
    **WPG2\_25Gray!;**  
    **WPG2\_12Gray!;**  
    **WPG2\_Crosshatch1!;**  
    **WPG2\_Crosshatch2!;**  
    **WPG2\_Crosshatch3!;**  
    **WPG2\_Crosshatch4!;**  
    **WPG2\_Lines1!;**  
    **WPG2\_Lines2!;**  
    **WPG2\_Lines3!;**  
    **WPG2\_Lines4!;**  
    **WPG2\_Lines5!;**  
    **WPG2\_Lines6!;**  
    **WPG2\_Lines7!;**  
    **WPG2\_Lines8!;**  
    **WPG2\_Checkboard1!;**  
    **WPG2\_Checkboard2!;**  
    **WPG2\_Bricks1!;**  
    **WPG2\_Bricks2!;**  
    **WPG2\_Fishscale!;**  
    **WPG2\_Honeycomb!;**  
    **WPG2\_Patio!;**  
    **WPG2\_Chainlink!;**  
    **WPG2\_Weave!;**  
    **WPG2\_Plaid!;**  
    **WPG2\_Balls!;**  
    **WPG2\_Mesh!;**  
    **WPG2\_Crosses!;**  
    **WPG2\_Triangles!;**  
    **WPG2\_Waves!;**  
    **WPG2\_Arch!};**  
  **[Specify a pattern set and a pattern from the set]**

**FillStyleCopy (**  
  **StyleName:Any {**  
    **Fill10!;**  
    **Fill20!;**  
    **Fill30!;**  
    **Fill40!;**  
    **Fill50!;**  
    **Fill60!;**  
    **Fill70!;**  
    **Fill80!;**  
    **Fill90!;**  
    **Fill100!;**  
    **FillButton!;**  
    **Fill5!;**  
    **NoFill!};**  
  **FromLibrary:Enumeration {**  
    **CurrentDocument!;**  
    **PersonalLibrary!;**  
    **SharedLibrary!};**  
  **ToLibrary:Enumeration {**

**CurrentDocument!;**  
    **PersonalLibrary!;**  
    **SharedLibrary!};**  
  **[DestName:String]**  
[Copy a fill style]

**FillStyleCreate (**  
  **TemplateName:String;**  
  **Library:Enumeration {**  
    **CurrentDoc!;**  
    **PersonalLibrary!;**  
    **SharedLibrary!};**  
  **[Create a fill style]**

**FillStyleDelete (**  
  **StyleName:Any {**  
    **Fill10!;**  
    **Fill20!;**  
    **Fill30!;**  
    **Fill40!;**  
    **Fill50!;**  
    **Fill60!;**  
    **Fill70!;**  
    **Fill80!;**  
    **Fill90!;**  
    **Fill100!;**  
    **FillButton!;**  
    **Fill5!;**  
    **NoFill!};**  
  **[Library:Enumeration {**  
    **CurrentDoc!;**  
    **PersonalLibrary!;**  
    **SharedLibrary!};**  
  **[Delete a specified fill style]**

**FillStyleEdit (**  
  **Style:Any {**  
    **Fill10!;**  
    **Fill20!;**  
    **Fill30!;**  
    **Fill40!;**  
    **Fill50!;**  
    **Fill60!;**  
    **Fill70!;**  
    **Fill80!;**  
    **Fill90!;**  
    **Fill100!;**  
    **FillButton!;**  
    **Fill5!;**  
    **NoFill!};**  
  **[Library:Enumeration {**  
    **CurrentDoc!;**  
    **PersonalLibrary!;**  
    **SharedLibrary!};**  
  **[Begin a FillStyleEdit-FillStyleEnd statement]**

**FillStyleEnd (State:Enumeration {Cancel!; Save!})**  
[Save or cancel changes to a fill style]

**FillStyleName (FillName:String)**  
[Name a user-created fill style]

**FillStyleRetrieve (**  
  **Filename:String;**  
  **Library:Enumeration {**  
    **CurrentDoc!;**  
    **PersonalLibrary!;**  
    **SharedLibrary!};**  
  **[Retrieve all fill graphics styles from the specified document or**  
  **template]**

**FillStyleSave (**  
  **Filename:String;**

**Library:Enumeration {**  
**CurrentDoc!;**  
**PersonalLibrary!;**  
**SharedLibrary!}**)

[Save a fill style]

**FindDig ()**

[Search for words or codes and replace them with new words or codes]

**FirstLineIndent (Width:Numeric Measurement)**

[Specify the distance between the left margin and the first word of indented text]

**FloatingCellBegin ()**

[Move the insertion point in a floating cell to the beginning of the cell]

**FloatingCellCreate ()**

[Create a floating cell]

**FloatingCellEdit ()**

[Search for a floating cell and move the insertion point to the first one found]

**FloatingCellEnd ()**

[Move the insertion point to the end of the current floating cell]

**FloatingCellFormula (Formula:String)**

[Specify a floating cell formula]

**FloatingCellNumAlignCurrency (Align:Enumeration {No!;Yes!})**

[Align currency symbols]

**FloatingCellNumberDateFormat (Format:Numeric)**

[Specify a floating cell date format]

**FloatingCellNumberDecDigits (Number:Numeric)**

[Specify the number of digits to display after the decimal point in a floating cell]

**FloatingCellNumberFormat (CellNumType:Enumeration {**

**General!;**  
**Integer!;**  
**Fixed2!;**  
**Percent!;**  
**Currency!;**  
**Accounting!;**  
**Commas!;**  
**Scientific!;**  
**DateFormat!;**  
**TextOnly!})**

[Specify a number type for floating cells]

**FloatingCellNumberNegNumber (NegDisplay:Enumeration {**

**Minus!;**  
**Parentheses!;**  
**CRDR!})**

[Specify how to display negative numbers in a floating cell]

**FloatingCellNumberRound (Round:Enumeration {No!; Yes!})**

[Round numbers for calculation in a floating cell]

**FloatingCellNumberUseCommas (Commas:Enumeration {No!; Yes!})**

[Display commas in floating cell numbers]

**FloatingCellNumberUseCurrency (Currency:Enumeration {No!; Yes!})**

[Use currency symbols in a floating cell]

**FloatingCellNumSelectCurrency (IntlCurrency:Enumeration {**

**DefaultCurrency!;**  
**Dollar!;**

**Peso!;**  
**FrenchCanada!;**  
**Canada!;**  
**Brazil!;**  
**Denmark!;**  
**Peseta!;**  
**Croatia!;**  
**CzechRepublic!;**  
**SlovakRepublic!;**  
**Finland!;**  
**FinlandIntl!;**  
**France!;**  
**Germany!;**  
**Greece!;**  
**Hungary!;**  
**Iceland!;**  
**Italy!;**  
**Japan!;**  
**Netherlands!;**  
**Norway!;**  
**NorwayIntl!;**  
**Ruble!;**  
**RussianRuble!;**  
**SouthAfrica!;**  
**Sweden!;**  
**Turkey!;**  
**UnitedKingdom!;**  
**Switzerland!;**  
**Austria!;**  
**Belgium!;**  
**Ecuador!;**  
**ElSalvador!;**  
**Guatemala!;**  
**Honduras!;**  
**Paraguay!;**  
**Venezuela!;**  
**Poland!;**  
**Slovenia!;**  
**Israel!;**  
**Korea!;**  
**China!;**  
**Taiwan!;**  
**HongKong!;**  
**Singapore!;**  
**Portugal!;**  
**BelgiumFrench!;**  
**Euro!})**

[Specify a currency symbol for floating cells]

**FlushRight ()**

[Right justify a line of text]

**FlushRightWithDotLeaders ()**

[Right justify a line of text with dot leaders]

**Font (**

**Name:String;**  
**[Family:Enumeration {**  
**FamilyITCBenguiat!;**  
**FamilyTimesRoman!;**  
**FamilyITCGaramond!;**  
**FamilyCenturySchoolbook!;**  
**FamilyBodon!;**  
**FamilyITCLubalinGraph!;**  
**FamilyAlbertus!;**  
**FamilyCourier!;**  
**FamilyHelvetica!;**  
**FamilyITCRonda!;**  
**FamilyMicrostyle!;**  
**FamilyOptima!;**  
**FamilyLetterGothic!;**  
**FamilyBrush!;**

FamilyParkAvenue!;  
 FamilySignetRoundhand!;  
 FamilyTekton!;  
 FamilyUncial!;  
 FamilyOldEnglish!;  
 FamilyITCZapfDingbats!;  
 FamilyBroadway!;  
 FamilyComputer!;  
 FamilyPTBarnum!;  
 FamilyPonderosa!;  
 FamilyMadrone!;  
 FamilyHobo!;  
 FamilyRevue!;  
 FamilyAestheticOrnamented!});  
 [Attributes:Enumeration {  
   FontMatchNormal!;  
   FontMatchItalic!;  
   FontMatchOutline!;  
   FontMatchShadow!;  
   FontMatchSmallCaps!});  
 [Weight:Enumeration {  
   WeightUnknown!;  
   WeightUltraThin!;  
   WeightUltraLight!;  
   WeightExtraThin!;  
   WeightThin!;  
   WeightExtraLight!;  
   WeightLight!;  
   WeightDemiLight!;  
   WeightSemiLight!;  
   WeightBook!;  
   WeightRegular!;  
   WeightRoman!;  
   WeightMedium!;  
   WeightDemiBold!;  
   WeightSemiBold!;  
   WeightBold!;  
   WeightExtraBold!;  
   WeightHeavy!;  
   WeightExtraHeavy!;  
   WeightBlack!;  
   WeightExtraBlack!;  
   WeightUltraHeavy!;  
   WeightUltraBlack!;  
   WeightMaximum!});  
 [Width:Enumeration {  
   WidthUnknown!;  
   WidthUltraCondensed!;  
   WidthUltraCompressed!;  
   WidthCompressed!;  
   WidthExtraCondensed!;  
   WidthCondensed!;  
   WidthSemiCondensed!;  
   WidthNormal!;  
   WidthSemiExpanded!;  
   WidthExpanded!;  
   WidthExtraExpanded!;  
   WidthUltraExpanded!;  
   WidthDoubleWide!;  
   WidthTripleWide!});  
 [Source:Enumeration {  
   PRSFile!;  
   DRSFile!});  
 [Type:Enumeration {  
   Speedo!;  
   IntelliFont!;  
   Type1!;  
   TrueType!});  
 [CharacterSet:Enumeration {  
   FontMatchASCII!;  
   FontMatchMultinational1!;  
   FontMatchMultinational2!;  
   FontMatchBoxDrawing!;  
   FontMatchTypographicSymbols!;  
   FontMatchIconicSymbols!;  
   FontMatchMath!;  
   FontMatchMathExtension!;  
   FontMatchGreek!;  
   FontMatchHebrew!;  
   FontMatchCyrillic!;  
   FontMatchJapaneseKana!;  
   FontMatchUserDefined!;  
   FontMatchArabic!;  
   FontMatchArabicScript!});  
 [Change the typeface of printed characters]

**FontDig ()**  
 [Change the font]

**FontDnShippingMacro ()**  
 [Reduce the font size by 2 points]

**FontDoubleUnderlineToggle ()**  
 [Turn on/off double underline font]

**FontDown ()**  
 [Reduce the font size by 1 point]

**FontExtraLargeToggle ()**  
 [Turn on/off Extra Large]

**FontFineToggle ()**  
 [Turn on/off Fine]

**FontHiddenToggle ()**  
 [Turn on/off Hidden]

**FontItalicToggle ()**  
 [Turn on/off Italic]

**FontLargeToggle ()**  
 [Turn on/off Large]

**FontOutlineToggle ()**  
 [Turn on/off the Outline attribute]

**FontRedlineToggle ()**  
 [Turn on/off Redline]

**FontShadowToggle ()**  
 [Turn on/off Shadow]

**FontSize (FontSize:Numeric Measurement)**  
 [Change the size of printed characters]

**FontSmallCapsToggle ()**  
 [Turn on/off Small Caps]

**FontSmallToggle ()**  
 [Turn on/off Small]

**FontStrikeoutToggle ()**  
 [Turn on/off Strikeout]

**FontSubscriptToggle ()**  
 [Turn on/off Subscript]

**FontSuperscriptToggle ()**  
 [Turn on/off Superscript]

**FontUnderlineToggle ()**  
 [Turn on/off Underline]

**FontUp ()**  
 [Increase the font size by 1 point]

**FontUpShippingMacro ()**  
[Increase the font size by 2 points]

**FontVeryLargeToggle ()**  
[Turn on/off Very Large]

**FooterA (**  
  **Action:Enumeration {Off!; Create!; Edit!};**  
  **[Occurrence:Enumeration {**  
    **NoPages!;**  
    **OddPages!;**  
    **EvenPages!;**  
    **AllPages!}]])**  
[Create, edit, or discontinue Footer A, and specify footer position]

**FooterB (**  
  **Action:Enumeration {Off!; Create!; Edit!};**  
  **[Occurrence:Enumeration {**  
    **NoPages!;**  
    **OddPages!;**  
    **EvenPages!;**  
    **AllPages!}]])**  
[Create, edit, or discontinue Footer B, and specify footer position]

**FooterSeparationDistance (Distance:Numeric Measurement)**  
[Specify the distance between text and footers]

**FootnoteContinuedMessage (State:Enumeration {No!; Yes!})**  
[Print (continued...) when a footnote continues on the next page]

**FootnoteCreate ()**  
[Create a footnote]

**FootnoteEdit ([FootnoteNumber:String])**  
[Open a specified footnote window]

**FootnoteEditDlg ()**  
[Edit a specified footnote]

**FootnoteEndnoteDlg ()**  
[Create and edit footnotes and endnotes]

**Numeric Measurement := FootnoteMinimum ([Width:Numeric Measurement])**  
[Specify the amount of footnote text to keep together on a page]

**FootnoteNewNumber (FootnoteNumber:String)**  
[Specify a new footnote number and increment subsequent footnotes from that number]

**FootnoteNewNumberDlg ()**  
[Start numbering of subsequent footnotes with a new number]

**FootnoteNumberDecrement ()**  
[Decrease the last footnote number by one]

**FootnoteNumberDisplay ()**  
[Insert the current footnote number at the insertion point]

**FootnoteNumberIncrement ()**  
[Increase the last footnote number by one]

**FootnoteNumberMethod (**  
  **NumberingMethod:Enumeration {**  
    **Numbers!;**  
    **LowerLetters!;**  
    **UpperLetters!;**  
    **LowerRoman!;**  
    **UpperRoman!;**  
    **Characters!};**  
  **[Character:String])**  
[Specify a footnote numbering method]

**FootnoteOptions (**  
  **[NumberingMethod:Enumeration {**  
    **Numbers!;**  
    **LowerLetters!;**  
    **UpperLetters!;**  
    **LowerRoman!;**  
    **UpperRoman!;**  
    **Characters!}];**  
  **[Characters:String];**  
  **[BetweenSpacing:Numeric Measurement];**  
  **[MinimumHeight:Numeric Measurement];**  
  **[Restart:Enumeration {Off!; On!}];**  
  **[PrintContinued:Enumeration {No!; Yes!}];**  
  **[FootnotePosition:Enumeration {**  
    **BottomOfPage!;**  
    **AfterText!}];**  
  **[Separator:Enumeration {**  
    **NoLine!;**  
    **ShortLine!;**  
    **MarginToMargin!}])**  
[Specify footnote options from the insertion point forward]

**FootnoteOptionsDlg ()**  
[Change the footnote numbering method, format and style]

**FootnotePosition (Position:Enumeration {**  
  **BottomOfPage!;**  
  **AfterText!})**  
[Print footnotes at the bottom of the page or just below the last line of text]

**FootnoteRestartEachPage (State:Enumeration {No!; Yes!})**  
[Restart footnote numbering on each page]

**FootnoteSeparatorLine (**  
  **Style:Any {**  
    **SingleLine!;**  
    **DoubleLine!;**  
    **DashedLine!;**  
    **DottedLine!;**  
    **ThickLine!;**  
    **ExtraThickLine!;**  
    **ThinThickLine!;**  
    **ThickThinLine!;**  
    **ButtonTopLeftLine!;**  
    **ButtonBottomRightLine!;**  
    **HeavySingleLine!;**  
    **HeavyDoubleLine!;**  
    **TableDefaultLine!;**  
    **NoLine!};**  
  **SpaceAboveLine:Numeric Measurement;**  
  **SpaceBelowLine:Numeric Measurement;**  
  **Alignment:Enumeration {**  
    **Left!;**  
    **Full!;**  
    **Center!;**  
    **Right!;**  
    **SetPosition!};**  
  **[Width:Numeric Measurement];**  
  **[Position:Numeric Measurement])**  
[Specify a separator line between text and footnotes]

**Numeric Measurement := FootnoteSpaceBetween ([Space:Numeric Measurement])**  
[Specify the amount of blank space between footnotes]

**FootnoteToEndnoteShippingMacro ()**  
[Convert footnotes to endnotes in the current document or selection]

**ForcePage (EvenOddNew:Enumeration {Even!;Odd!;New!})**  
[Force a page break or specify that a certain page always has an odd or even number]

**ForcePageDlg ()**

[Make a page be an odd or even numbered page]

**FormatMarginsDlg ()**

[Change the left, right, top, and bottom margins]

**FormElementsToolbar ({State:Enumeration {On!;Off!}})**

[Obsolete Command]

- G -

**GenerateDlg ()**

[Create an updated index, table of contents, etc.]

**GenerateDoc (**

**[SaveSubDocs:Enumeration {DontSave!; Save!}];**

**[UndoAction:Enumeration {NoUndo!; Undo!}];**

**[Hypertext:Enumeration {CreateHypertext!}])**

[Generate indexes, tables of contents, lists, cross references, and tables of authorities]

**GetAcceleratorText (Variable:Variable; Token:Command Name)**

[Assign the accelerator text of a macro command to a variable]

**GraphicBoxStyleDlg ()**

[Change the style of the current box]

**GraphicCreateBox ([BoxStyleName:Any])**

[Display the Insert Image dialog box]

**GraphicCreateChart ()**

[Create a chart]

**GraphicCreateTextArt ()**

[Add special effects to text]

**GraphicEditBox ([BoxNumber:Numeric])**

[Change box appearance, position, and content]

**GraphicsBoxCaptionDlg ()**

[Create or edit a box caption]

**GraphicsBoxContentDlg ()**

[Change the current box content type and position, and rotate text]

**GraphicsBoxFindDlg ()**

[Find a graphics box]

**GraphicsBoxSizeDlg ()**

[Change the height and width of the current box]

**GraphicsFigureEditAIDlg ()**

[Move, scale, rotate, set color and fill attributes, etc. for the current image]

**GraphicsLineColor (**

**[ColorName:String];**

**[Red:Numeric];**

**[Green:Numeric];**

**[Blue:Numeric];**

**[Shade:Numeric])**

[Specify the color of a graphics line]

**GraphicsLineCreate ([LineType:Enumeration { HorizontalLine!; VerticalLine!}])**

[Create a graphics line]

**GraphicsLineDelete (Line:Numeric)**

[Delete a graphics line]

**GraphicsLineEdit (Line:Numeric)**

[Specify a graphics line to edit]

**GraphicsLineEditNext ([LineType:Enumeration { HorizontalLine!; VerticalLine!}])**

[Begin editing of the next graphics line]

**GraphicsLineEditPrevious ([LineType:Enumeration { HorizontalLine!; VerticalLine!}])**

[Begin editing of the previous graphics line]

**GraphicsLineEnd (State:Enumeration {Cancel!;Save!})**

[Save or cancel changes made to a graphics line]

**GraphicsLineHorizontalPosition ( Position:Enumeration {**

**Left!;**

**Center!;**

**Right!;**

**FullAlign!;**

**Set!;**

**BetweenColumns!;**

**[AbsPos:Numeric Measurement];**

**[LeftColNum:Numeric])**

[Position a graphics line horizontally]

**GraphicsLineLength (Length:Numeric Measurement)**

[Specify the length of graphics lines]

**GraphicsLineSpacing (**

**TopSpace:Numeric Measurement;**

**[BottomSpace:Numeric Measurement])**

[Specify the amount of white space between a graphics line and text]

**GraphicsLineStyle (Style:Any {**

**SingleLine!;**

**DoubleLine!;**

**DashedLine!;**

**DottedLine!;**

**ExtraThickLine!;**

**ThickLine!;**

**ThinThickLine!;**

**ThickThinLine!;**

**ButtonTopLeft!;**

**ButtonBottomRight!;**

**HeavySingleLine!;**

**HeavyDoubleLine!;**

**TableDefaultLine!;**

**NoLine!})**

[Specify a graphics line style]

**GraphicsLineThickness (Thickness:Any)**

[Specify the thickness of a graphics line]

**GraphicsLineType (LineType:Enumeration {**

**Horizontal!;**

**Vertical!})**

[Specify whether a graphics line is horizontal or vertical]

**GraphicsLineUpdateDisplay (State:Enumeration {Cancel!;Save!})**

[Update a graphics line display after editing]

**GraphicsLineUseColor (State:Enumeration {No!;Yes!})**

[Select or deselect the default line style color for a graphics line]

**GraphicsLineVerticalPosition (**

**Position:Enumeration {**

**Top!;**

**Center!;**

**Bottom!;**

**Baseline!;**

**FullAlign!;**

**Set!};**

**[Where:Numeric Measurement]**  
[Specify the vertical position of a graphics line]

**GraphicsStylesDlg ()**  
[Create and edit graphics styles]

**Numeric := Guidelines (**  
    **[ShowTableGuides:Enumeration {No!; Yes!};**  
    **[ShowMarginGuides:Enumeration {No!; Yes!};**  
    **[ShowColumnGuides:Enumeration {No!; Yes!};**  
    **[ShowHeaderFooterGuides:Enumeration {No!; Yes!};**  
    **[EnableGuidesDrag:Enumeration {No!; Yes!};]**  
[Show or hide guidelines and return the current state]

**GuidelinesDlg ()**  
[Select guidelines to display]

- H -

**HardColumnBreak ()**  
[Begin a new column]

**HardPageBreak ()**  
[Begin a new page]

**HardPageBreakInsert ()**  
[Begin a new page]

**HardPageBreakKey ()**  
[Insert a hard page break at the insertion point]

**HardReturn ()**  
[Begin a new line (hard return)]

**HardReturnInsert ()**  
[Insert a hard return in an outline]

**HardSpace ()**  
[Insert a space that holds two words together]

**HeaderA (**  
    **Action:Enumeration {Off!; Create!; Edit!};**  
    **[Occurrence:Enumeration {**  
        **NoPages!;**  
        **OddPages!;**  
        **EvenPages!;**  
        **AllPages!};]**  
[Specify whether to create, edit, or discontinue Header A]

**HeaderB (**  
    **Action:Enumeration {Off!; Create!; Edit!};**  
    **[Occurrence:Enumeration {**  
        **NoPages!;**  
        **OddPages!;**  
        **EvenPages!;**  
        **AllPages!};]**  
[Specify whether to create, edit, or discontinue Header B]

**HeaderFooterDistanceDlg ()**  
[Set the distance between text and the header or footer]

**HeaderFooterDlg ()**  
[Create and edit headers and footers]

**HeaderFooterNext ()**  
[Go to the next header/footer]

**HeaderFooterPlacement ([State:Enumeration {**  
    **NoPages!;**  
    **OddPages!;**  
    **EvenPages!;**  
    **AllPages!};])**  
[Specify on which pages headers and footers occur]

**HeaderFooterPlacementDlg ()**  
[Select odd/even pages or every page for header and footer display]

**HeaderFooterPrev ()**  
[Go to the previous header/footer]

**HeaderSeparationDistance (Distance:Numeric Measurement)**  
[Specify the distance between text and headers]

**HelpContents ()**  
[Display Help Contents]

**HelpHelpTopics ()**  
[Display a list of Help topics]

**HelpHowDoI ()**  
[Get instructions on how to use Help]

**HelpHowToUseHelp ()**  
[Get instructions on how to use Help]

**HelpKeystrokes ()**  
[Display templates and keystroke information]

**HelpMacroManual ()**  
[Get online Help on using macros]

**HelpSearchForHelpOn ()**  
[Display the Help Index]

**HelpWhatIs ()**  
[Use the mouse and keyboard to select the item you want help on]

**HiddenText (State:Enumeration {Off!;On!})**  
[Turn Hidden Text on or off for selected text]

**HiddenTextShowAll (State:Enumeration {Off!;On!})**  
[Display or hide hidden text]

**HideBars ([State:Enumeration {Off!;On!};])**  
[Hide all bars]

**HighlightColor (**  
    **Red:Numeric;**  
    **Green:Numeric;**  
    **Blue:Numeric;**  
    **Shade:Numeric)**  
[Set the color and shade of the highlighter]

**HighlightColorDlg ()**  
[Select a highlight color]

**HighlightPrint ()**  
[Print highlighting]

**HighlightRemove ()**  
[Remove highlighting from the selected text]

**HighlightToggle ()**  
[Turn on/off Highlight]

**HLineCreate ()**  
[Put a line in your document from the left margin to the right margin]

**HTMLPublish (Filename:String)**  
[Save the current document to HTML format]

**Hypertext ([State:Enumeration {Inactive!;Active!};])**  
[Activate or deactivate all hyperlinks]

**HypertextCreate (**  
    **Action:Enumeration {**  
        **GoToBookmark!;**  
        **RunMacro!};**

**[Macro:String];**  
**[Bookmark:String];**  
**[Display:Enumeration {**  
    **HighlightedText!**  
    **Button!}]**)  
[Create and specify the appearance of a hyperlink]

**HypertextCreateDlg ()**  
[Create a hyperlink]

**HypertextDelete ()**  
[Remove the current hyperlink]

**HypertextEdit (**  
    **Action:Enumeration {**  
        **GoToBookmark!**  
        **RunMacro!**;  
    **[Macro:String];**  
    **[Bookmark:String];**  
    **[Appearance:Enumeration {**  
        **HighlightedText!**  
        **Button!}]**)  
[Edit a hyperlink]

**HypertextEditDlg ()**  
[Edit a hyperlink]

**HypertextExecute ()**  
[Perform a hyperlink]

**Any := HypertextInfo (Data:Enumeration {**  
    **Action!**;  
    **Name!**;  
    **Bookmark!**;  
    **Appearance!})**  
[Return information about the current hyperlink]

**HypertextNext ()**  
[Find the next hyperlink]

**HypertextPrevious ()**  
[Find the previous hyperlink]

**HypertextReturnFrom ()**  
[Return the insertion point from a bookmark to the bookmark's hyperlink]

**HypertextStyleEdit ()**  
[Edit the Hyperlink Style]

**Hyphenation (State:Enumeration {Off!;On!})**  
[Turn Hyphenation on or off]

**HyphenationIgnore ()**  
[Cancel hyphenation of the current word]

**HyphenationSoftReturn ()**  
[Insert a soft return to break the word, if necessary, at the insertion point]

**HyphenationZoneLeft (HZone:Numeric)**  
[Specify the lower limit of a hyphenation zone]

**HyphenationZoneRight (HZone:Numeric)**  
[Specify the upper limit of a hyphenation zone]

- 1 -

**ImportCreateDlg ()**  
[Create a data link to a spreadsheet, database, or data file]

**ImportDlg ()**  
[Import data from a spreadsheet, database, or data file]

**ImportDoEdit ()**  
[Modify and perform a link between a spreadsheet or database and the current document.]

**ImportDoImport ()**  
[Import ASCII text, ANSI text, table data, spreadsheet data, or database data]

**ImportDoLink ()**  
[Create and perform a link between a spreadsheet or database and the current document]

**ImportDoUpdate ()**  
[Update linked data from the currently defined spreadsheet or database]

**ImportEditDlg ()**  
[Edit a data link to a spreadsheet, database, or data file]

**ImportOptionsDlg ()**  
[Change options for data links]

**ImportSetAsciiEncap (EncapsulationChar:String)**  
[Specify an ASCII encapsulation character for importing data saved as ASCII delimited text]

**ImportSetAsciiFieldDelimiter (FieldDelimiter:String)**  
[Specify the characters to mark the beginning or end of a field for data saved as ASCII delimited text]

**ImportSetAsciiRecordDelimiter (RecordDelimiter:String)**  
[Specify the characters to mark the beginning or end of a record for data saved as ASCII delimited text]

**ImportSetAsciiStrip (StripChars:String)**  
[Delete extra characters when importing spreadsheet or database data saved as ASCII delimited text]

**ImportSetDataSource (Source:String)**  
[Specify the data source for an ODBC import]

**ImportSetDestination (DestinationType:Enumeration {**  
    **WPTable!**;  
    **TabbedText!**;  
    **MergeData!})**  
[Specify a format for linked or imported spreadsheet or database data]

**ImportSetDictPath (DictionaryPath:String)**  
[Specify the NetWare SQL dictionary path for imported spreadsheet or database data]

**ImportSetFields (FieldNames:String)**  
[Specify the database fields to import]

**ImportSetFileName (Filename:String)**  
[Specify a source file to link to or import from]

**ImportSetFilePath (FilePath:String)**  
[Specify a NetWare SQL file path]

**ImportSetFirstRecFieldnames (State:Enumeration {No!;Yes!})**  
[The first record is fieldnames]

**ImportSetIncludeNames (State:Enumeration {Off!;On!})**  
[Use field names as headings for imported database data]

**ImportSetPanel (PanelName:String)**  
[Specify a DataPerfect panel to retrieve]

**ImportSetPassword (Password:String)**  
[Specify a password to import SQL data]



**ImportsetQuery** (  
[FieldName1:String];  
[FieldName2:String];  
[FieldName3:String];  
[Condition1Field1:String];  
[Condition1Field2:String];  
[Condition1Field3:String];  
[Condition2Field1:String];  
[Condition2Field2:String];  
[Condition2Field3:String];  
[Condition3Field1:String];  
[Condition3Field2:String];  
[Condition3Field3:String];  
[Condition4Field1:String];  
[Condition4Field2:String];  
[Condition4Field3:String])  
[Specify field conditions for a database import query]

**ImportSetRange** (Range:String)  
[Specify a range of spreadsheet cells to import]

**ImportSetServer** (ServerName:String)  
[Specify a server for a NetWare SQL database import]

**ImportSetSizeToFit** (State:Enumeration {No!;Yes!})  
[Size table columns to fit data after it is imported]

**ImportSetSource** (SourceType:Enumeration {  
Spreadsheet!;  
dBase!;  
Paradox!;  
FoxPro!;  
Clipper!;  
DataPerfect!;  
ASCII!;  
ANSIText!;  
NetWare!;  
Oracle!;  
DB2!;  
SQLServer!;  
Sybase!;  
SQLBase!;  
XDB!;  
InFormix!;  
ODBC!;  
ODBC\_NO\_SQL!})  
[Specify the format of spreadsheet or database data to import]

**ImportSetSQLQuery** (SQLQuery:String)  
[Specify conditions for a database link or import query]

**ImportSetTable** (TableName:String)  
[Specify a table to import]

**ImportSetUserID** (UserId:String)  
[Specify a NetWare SQL database user identification]

**ImportUpdateDlg** ()  
[Update linked data from spreadsheets, databases, and data files]

**Indent** ()  
[Indent the current paragraph one tab stop]

**IndentLeftRight** ()  
[Indent the current paragraph equally from both margins]

**IndexDefineDlg** ()  
[Select an index format and specify a concordance file]

**IndexDefinition** (  
Heading:Any {  
NormalStyle!;  
FootnoteNumberInDocumentStyle!;

EndnoteNumberInDocumentStyle!;  
FigureNumberStyle!;  
TableBoxNumberStyle!;  
TextBoxNumberStyle!;  
UserBoxNumberStyle!;  
EquationNumberStyle!;  
HypertextStyle!;  
ToC1Style!;  
ToC2Style!;  
ToC3Style!;  
ToC4Style!;  
ToC5Style!;  
Index1Style!;  
Index2Style!;  
ListStyle!;  
ToAStyle!;  
DocStyle!;  
BoxTextStyle!;  
CaptionStyle!;  
FootnoteStyle!;  
EndnoteStyle!;  
CommentStyle!;  
HeaderAStyle!;  
HeaderBStyle!;  
FooterAStyle!;  
FooterBStyle!;  
WatermarkAStyle!;  
WatermarkBStyle!;  
ParagraphOutInStyle!;  
OutlineStyle!;  
LegalOutInStyle!;  
BulletsOutInStyle!;  
HeadingsOutInStyle!;  
Legal\_2OutInStyle!;  
NumbersOutInStyle!;  
Level1Style!;  
Level2Style!;  
Level3Style!;  
Level4Style!;  
Level5Style!;  
Level6Style!;  
Level7Style!;  
Level8Style!;  
Legal1Style!;  
Legal2Style!;  
Legal3Style!;  
Legal4Style!;  
Legal5Style!;  
Legal6Style!;  
Legal7Style!;  
Legal8Style!;  
Heading1Style!;  
Heading2Style!;  
Heading3Style!;  
Heading4Style!;  
Heading5Style!;  
Heading6Style!;  
Heading7Style!;  
Heading8Style!;  
MonospacedStyle!;  
CustomHTMLStyle!;  
PreformattedStyle!;  
AddressStyle!;  
QuotationsOutInStyle!;  
DefinitionsOutInStyle!;  
Quotation1Style!;  
Quotation2Style!;  
Quotation3Style!;  
Quotation4Style!;  
Quotation5Style!;  
Quotation6Style!;  
Quotation7Style!;

Quotation8Style!  
 Definition1Style!  
 Definition2Style!  
 Definition3Style!  
 Definition4Style!  
 Definition5Style!  
 Definition6Style!  
 Definition7Style!  
 Definition8Style!  
 Level9Style!  
 Legal9Style!  
 Heading9Style!  
 Quotation9Style!  
 Definition9Style!};  
 SubHeading:Any {  
   NormalStyle!  
   FootnoteNumberInDocumentStyle!  
   EndnoteNumberInDocumentStyle!  
   FigureNumberStyle!  
   TableBoxNumberStyle!  
   TextBoxNumberStyle!  
   UserBoxNumberStyle!  
   EquationNumberStyle!  
   HypertextStyle!  
   ToC1Style!  
   ToC2Style!  
   ToC3Style!  
   ToC4Style!  
   ToC5Style!  
   Index1Style!  
   Index2Style!  
   ListStyle!  
   ToAStyle!  
   DocStyle!  
   BoxTextStyle!  
   CaptionStyle!  
   FootnoteStyle!  
   EndnoteStyle!  
   CommentStyle!  
   HeaderAStyle!  
   HeaderBStyle!  
   FooterAStyle!  
   FooterBStyle!  
   WatermarkAStyle!  
   WatermarkBStyle!  
   ParagraphOutInStyle!  
   OutlineStyle!  
   LegalOutInStyle!  
   BulletsOutInStyle!  
   HeadingsOutInStyle!  
   Legal\_2OutInStyle!  
   NumbersOutInStyle!  
   Level1Style!  
   Level2Style!  
   Level3Style!  
   Level4Style!  
   Level5Style!  
   Level6Style!  
   Level7Style!  
   Level8Style!  
   Legal1Style!  
   Legal2Style!  
   Legal3Style!  
   Legal4Style!  
   Legal5Style!  
   Legal6Style!  
   Legal7Style!  
   Legal8Style!  
   Heading1Style!  
   Heading2Style!  
   Heading3Style!  
   Heading4Style!

Heading5Style!  
 Heading6Style!  
 Heading7Style!  
 Heading8Style!  
 MonospacedStyle!  
 CustomHTMLStyle!  
 PreformattedStyle!  
 AddressStyle!  
 QuotationsOutInStyle!  
 DefinitionsOutInStyle!  
 Quotation1Style!  
 Quotation2Style!  
 Quotation3Style!  
 Quotation4Style!  
 Quotation5Style!  
 Quotation6Style!  
 Quotation7Style!  
 Quotation8Style!  
 Definition1Style!  
 Definition2Style!  
 Definition3Style!  
 Definition4Style!  
 Definition5Style!  
 Definition6Style!  
 Definition7Style!  
 Definition8Style!  
 Level9Style!  
 Legal9Style!  
 Heading9Style!  
 Quotation9Style!  
 Definition9Style!};  
 NumMode:Enumeration {  
   DefNone!  
   DefFollows!  
   DefParentheses!  
   DefFlushRight!  
   DefLeader!};  
 PgCombine:Enumeration {  
   NoCombine!  
   Combine!};  
 [Concordance:String];  
 [UserFormat:String]  
 [Create and insert an index definition]

**IndexMark** ([Heading:String]; [Subheading:String])  
 [Mark where current heading and subheading will be inserted]

**Boolean := InhibitInput** ([State:Enumeration {Off!; On!}])  
 [Enable or disable keyboard and menu input during macro execution]

**InitialCodesDoc** ()  
 [Begin initial codes. All codes between InitialCodesDoc and SubstructureExit are placed in initial codes]

**InitialCodesStyleDlg** ()  
 [Edit the current document style]

**InsertDefaultBullet** ()  
 [Begin a list with the current Bullets \& Numbers Style]

**InsertFilename** ()  
 [Insert the filename of the current document]

**InsertFilenameWithPath** ()  
 [Insert the path and filename of the current document]

**InsertMergeField** ()  
 [Insert a merge field]

**InsertOtherCodesDlg** ()  
 [Insert other formatting codes at the insertion point]

**InternetPublisherDlg ()**

[Create, browse, and publish web documents]

**InvokeDialog (Dialog:Command Name)**

[Display a specified dialog box but do not pause the macro for input]

**IrregularCap (State:Enumeration {Off!; On!})**

[Lowercase the second of two initial caps at the beginning of a word]

**Boolean := IsTokenValid (****[Variable:Variable;  
Token:Command Name)**

[Indicate whether a specified menu item or dialog is available or a macro command is executable]

**- J -****Justification (Justification:Enumeration {****Left!;  
Full!;  
Center!;  
Right!;  
FullAll!;  
DecAlign!})**

[Align text]

**JustifyAll ()**

[Align text (including the last line) on the left and right margins]

**JustifyCenter ()**

[Center text between the margins]

**JustifyFull ()**

[Align text on the left and right margins]

**JustifyLeft ()**

[Align text on the left margin]

**JustifyRight ()**

[Align text on the right margin]

**- K -****KeepTextTogetherDlg ()**

[Specify the lines of text to be kept together on a page]

**Kerning (State:Enumeration {Off!; On!})**

[Turn Automatic Kerning on or off]

**KerningSpacing ([KerningSpacingValue:Numeric Measurement])**

[Move the character to the right of the insertion point a specified distance to the left or right]

**KeyboardSelect (****Keyboard:String;  
[LoadedFromDefaultTemplate:Enumeration {No!; Yes!}])**

[Select a keyboard]

**KeyboardSelectDlg ()**

[Create and edit keyboard layouts]

**- L -****LabelsDefine (****Filename:String;  
LabelName:String;  
LabelFlags:Enumeration {  
Laser!;  
TractorFed!;  
LaserandTractorFed!};  
SheetWidth:Numeric Measurement;  
SheetLength:Numeric Measurement;  
ColumnsPerPage:Numeric;****RowsPerPage:Numeric;  
LabelWidth:Numeric Measurement;  
LabelLength:Numeric Measurement;  
LeftOffset:Numeric Measurement;  
TopOffset:Numeric Measurement;  
BetweenColumns:Numeric Measurement;  
BetweenRows:Numeric Measurement;  
MarginLeft:Numeric Measurement;  
MarginRight:Numeric Measurement;  
MarginTop:Numeric Measurement;  
MarginBottom:Numeric Measurement)**

[Create a customized labels definition]

**LabelsDelete (Filename:String;LabelName:String)**

[Delete a labels definition]

**LabelsDlg ()**

[Create, edit, and select labels]

**LabelsFileCreate (Filename:String)**

[Create a labels file]

**LabelsFileDescription (Filename:String;Description:String)**

[Create a description for a labels file]

**LabelsFileSelect (Filename:String)**

[Select a labels file]

**LabelsOff ()**

[Turn off labels mode]

**LabelsSelect (Filename:String;LabelName:String)**

[Select a label definition]

**Language (Language:String)**

[Specify a language for a document or a selection of text]

**LanguageDlg ()**

[Specify the language of text, and disable Spell Checker and Grammatik]

**LaunchOnlineService (ProgramPath:String;Address:String)**

[Start an online service]

**Leading (Adjustment:Numeric Measurement)**

[Specify a leading adjustment]

**LineBreak ()**

[Begin a new line without ending the paragraph]

**LineCreateDlg ()**

[Create a horizontal or vertical line with custom attributes]

**LineEditDlg ()**

[Edit a graphics line]

**LineHeight (Height:Any {Auto!};[Minimum:Numeric Measurement])**

[Specify the line height, measured from baseline to baseline in single-spaced text]

**LineHeightDlg ()**

[Specify line height for each line]

**LineHyphenationDlg ()**

[Turn Hyphenation on or off and set zones]

**LineNumbering (****State:Enumeration {Off!; On!};  
[RestartAtNewPage:Enumeration {  
DontRestartNumbersEachPage!;  
RestartNumbersEachPage!}];  
[CountBlankLines:Enumeration {**

**DontCountBlankLines!**;  
**CountBlankLines!**];  
**[NumberColumns:Enumeration {**  
    **DontNumberAllNewspaperColumns!**;  
    **NumberAllNewspaperColumns!}}**];  
**[PositionRelativeToLeftEdge:Enumeration {**  
    **PosNumberFromLeftEdgeOfPaper!**;  
    **PosNumberFromLeftMargin!}}**];  
**[Position:Numeric Measurement!]**;  
**[FirstPrintingNumber:Numeric!]**;  
**[NumberInterval:Numeric!]**;  
**[Default:Enumeration {**  
    **UseDefaultValues!**;  
    **DontUseDefaultValues!}}**];  
**[FontFace:String!]**;  
**[FontSize:Numeric Measurement!]**;  
**[Red:Numeric!]**;  
**[Green:Numeric!]**;  
**[Blue:Numeric!]**;  
**[Shade:Numeric!]**;  
**{[AttributesOn:Enumeration {**  
    **ExtraLarge!**;  
    **VeryLarge!**;  
    **Large!**;  
    **Small!**;  
    **Fine!**;  
    **Superscript!**;  
    **Subscript!**;  
    **Outline!**;  
    **Italics!**;  
    **Shadow!**;  
    **Redline!**;  
    **DoubleUnderline!**;  
    **Bold!**;  
    **Strikeout!**;  
    **Underline!**;  
    **SmallCaps!}}**)}  
[Turn Line Numbering on or off, and specify appearance]

**LineNumberingDlg ()**  
[Turn Line Numbering on or off, and specify appearance]

**LineNumberingMethod (NumberingMethod:Enumeration {**  
    **Numbers!**;  
    **LowerLetters!**;  
    **UpperLetters!**;  
    **LowerRoman!**;  
    **UpperRoman!}}**)  
[Specify a method for numbering lines]

**LineNumberSet (Number:Numeric)**  
[Specify the first line number on a page]

**LineStyleColor (**  
    **[ColorName:String!]**;  
    **[Red:Numeric!]**;  
    **[Green:Numeric!]**;  
    **[Blue:Numeric!]**;  
    **[Shade:Numeric!]**)  
[Specify the color of a line segment]

**LineStyleCreate (Position:Numeric)**  
[Create a custom line style of stacked lines]

**LineStyleDelete (Position:Numeric)**  
[Delete a single line of a stacked custom line]

**LineStyleEdit (Position:Numeric)**  
[Specify a line segment to edit]

**LineStyleEnd (State:Enumeration {Cancel!; Save!})**  
[Save or cancel changes to a line segment]

**LineStyleSetUser (Onoffs:Numeric)**  
[Begin a custom line pattern definition, and specify the number of dashes and spaces in the pattern]

**LineStyleSetUserData (SegSize:Numeric Measurement)**  
[Specify a dash or space length in a custom line pattern]

**LineStyleSetUserEnd ()**  
[End a custom line pattern definition]

**LineStyleSetWPG2 (**  
    **Style:Numeric;**  
    **LineStyle:Enumeration {**  
        **SingleLine!**;  
        **DoubleLine!**;  
        **DashedLine!**;  
        **DottedLine!**;  
        **ThickLine!**;  
        **ExtraThickLine!**;  
        **ThinThickLine!**;  
        **ThickThinLine!**;  
        **ButtonTopLeftLine!**;  
        **ButtonBottomRightLine!**;  
        **HeavySingleLine!**;  
        **HeavyDoubleLine!**;  
        **TableDefaultLine!**;  
        **NoLine!}}**)  
[Set a line segment pattern to a predefined WPG2 line pattern]

**LineStyleSpacing (Spacing:Numeric Measurement)**  
[Specify the distance between lines in a stacked custom line]

**LineStyleWidth (Width:Numeric Measurement)**  
[Set the thickness of a line segment]

**LineStyleWavy (LineWavy:Enumeration {**  
    **NoWavy!**;  
    **SquareWavy!**;  
    **PointyWavy!**;  
    **BothWavy!}}**)  
[Set the wavyness of a line segment]

**LineStyleWavyWidth (Index:Numeric)**  
[Set the thickness of a wavy line segment]

**LineSpacing (Spacing:Numeric)**  
[Specify line spacing for the current and subsequent paragraphs]

**LineSpacingDlg ()**  
[Change the distance between lines of text in the document]

**LineStyleCopy (**  
    **Style:Any {**  
        **SingleLine!**;  
        **DoubleLine!**;  
        **DashedLine!**;  
        **DottedLine!**;  
        **ThickLine!**;  
        **ExtraThickLine!**;  
        **ThinThickLine!**;  
        **ThickThinLine!**;  
        **ButtonTopLeftLine!**;  
        **ButtonBottomRightLine!**;  
        **HeavySingleLine!**;  
        **HeavyDoubleLine!**;  
        **TableDefaultLine!**;  
        **NoLine!}}**;  
    **FromLibrary:Enumeration {**  
        **CurrentDocument!**;  
        **PersonalLibrary!**;  
        **SharedLibrary!}}**;  
    **ToLibrary:Enumeration {**  
        **CurrentDocument!**;

```
PersonalLibrary!;  
SharedLibrary!);  
[DestName:String])  
[Copy a graphics line style]
```

```
LineStyleCreate (  
Name:String;  
[Library:Enumeration {  
CurrentDoc!;  
PersonalLibrary!;  
SharedLibrary!})]  
[Create a graphics line style]
```

```
LineStyleDelete (  
Name:Any {  
SingleLine!;  
DoubleLine!;  
DashedLine!;  
DottedLine!;  
ThickLine!;  
ExtraThickLine!;  
ThinThickLine!;  
ThickThinLine!;  
ButtonTopLeftLine!;  
ButtonBottomRightLine!;  
HeavySingleLine!;  
HeavyDoubleLine!;  
TableDefaultLine!;  
NoLine!};  
[Library:Enumeration {  
CurrentDoc!;  
PersonalLibrary!;  
SharedLibrary!})]  
[Delete a user-defined graphics line style]
```

```
LineStyleEdit (  
Style:Any {  
SingleLine!;  
DoubleLine!;  
DashedLine!;  
DottedLine!;  
ThickLine!;  
ExtraThickLine!;  
ThinThickLine!;  
ThickThinLine!;  
ButtonTopLeftLine!;  
ButtonBottomRightLine!;  
HeavySingleLine!;  
HeavyDoubleLine!;  
TableDefaultLine!;  
NoLine!};  
[Library:Enumeration {  
CurrentDoc!;  
PersonalLibrary!;  
SharedLibrary!})]  
[Edit a line style]
```

```
LineStyleEnd (State:Enumeration {Cancel!; Save!})  
[Save or cancel changes to a graphics line style]
```

```
LineStyleName (LineName:String)  
[Rename a user-defined graphics line style]
```

```
LineStyleRetrieve (  
Filename:String;  
Style:Enumeration {  
CurrentDoc!;  
PersonalLibrary!;  
SharedLibrary!})  
[Retrieve all line graphics styles from the specified document or  
template]
```

```
LineStyleSave (  
[Save a line style library]
```

```
FileName:String;  
Style:Enumeration {  
CurrentDoc!;  
PersonalLibrary!;  
SharedLibrary!})  
[Save a line style library]
```

```
ListDefineDlg ()  
[Define a list]
```

```
ListDefinition (  
ListName:String;  
[Style:Any {  
NormalStyle!;  
FootnoteNumberInDocumentStyle!;  
EndnoteNumberInDocumentStyle!;  
FigureNumberStyle!;  
TableBoxNumberStyle!;  
TextBoxNumberStyle!;  
UserBoxNumberStyle!;  
EquationNumberStyle!;  
HypertextStyle!;  
ToC1Style!;  
ToC2Style!;  
ToC3Style!;  
ToC4Style!;  
ToC5Style!;  
Index1Style!;  
Index2Style!;  
ListStyle!;  
ToAStyle!;  
DocStyle!;  
BoxTextStyle!;  
CaptionStyle!;  
FootnoteStyle!;  
EndnoteStyle!;  
CommentStyle!;  
HeaderAStyle!;  
HeaderBStyle!;  
FooterAStyle!;  
FooterBStyle!;  
WatermarkAStyle!;  
WatermarkBStyle!;  
ParagraphOutInStyle!;  
OutlineStyle!;  
LegalOutInStyle!;  
BulletsOutInStyle!;  
HeadingsOutInStyle!;  
Legal_2OutInStyle!;  
NumbersOutInStyle!;  
Level1Style!;  
Level2Style!;  
Level3Style!;  
Level4Style!;  
Level5Style!;  
Level6Style!;  
Level7Style!;  
Level8Style!;  
Legal1Style!;  
Legal2Style!;  
Legal3Style!;  
Legal4Style!;  
Legal5Style!;  
Legal6Style!;  
Legal7Style!;  
Legal8Style!;  
Heading1Style!;  
Heading2Style!;  
Heading3Style!;  
Heading4Style!;  
Heading5Style!;  
Heading6Style!;
```

```

Heading7Style!;
Heading8Style!;
MonospacedStyle!;
CustomHTMLStyle!;
PreformattedStyle!;
AddressStyle!;
QuotationsOutInStyle!;
DefinitionsOutInStyle!;
Quotation1Style!;
Quotation2Style!;
Quotation3Style!;
Quotation4Style!;
Quotation5Style!;
Quotation6Style!;
Quotation7Style!;
Quotation8Style!;
Definition1Style!;
Definition2Style!;
Definition3Style!;
Definition4Style!;
Definition5Style!;
Definition6Style!;
Definition7Style!;
Definition8Style!;
Level9Style!;
Legal9Style!;
Heading9Style!;
Quotation9Style!;
Definition9Style!});
[Numbering:Enumeration {
  DefNone!;
  DefFollows!;
  DefParentheses!;
  DefFlushRight!;
  DefLeader!});
[Counter:Any {
  FigureBox!;
  TableBox!;
  TextBox!;
  UserBox!;
  EquationBox!;
  ButtonBox!;
  WatermarkImageBox!;
  InlineEquationBox!;
  NoBox!});
[NumberingFormat:String])
[Select a list definition and specify where to generate the list]

ListDefinitionCreate (
  ListName:String;
  [Style:Any {
    NormalStyle!;
    FootnoteNumberInDocumentStyle!;
    EndnoteNumberInDocumentStyle!;
    FigureNumberStyle!;
    TableBoxNumberStyle!;
    TextBoxNumberStyle!;
    UserBoxNumberStyle!;
    EquationNumberStyle!;
    HypertextStyle!;
    ToC1Style!;
    ToC2Style!;
    ToC3Style!;
    ToC4Style!;
    ToC5Style!;
    Index1Style!;
    Index2Style!;
    ListStyle!;
    ToAStyle!;
    DocStyle!;
    BoxTextStyle!;
    CaptionStyle!;
    FootnoteStyle!;
    EndnoteStyle!;
    CommentStyle!;
    HeaderAStyle!;
    HeaderBStyle!;
    FooterAStyle!;
    FooterBStyle!;
    WatermarkAStyle!;
    WatermarkBStyle!;
    ParagraphOutInStyle!;
    OutlineStyle!;
    LegalOutInStyle!;
    BulletsOutInStyle!;
    HeadingsOutInStyle!;
    Legal_2OutInStyle!;
    NumbersOutInStyle!;
    Level1Style!;
    Level2Style!;
    Level3Style!;
    Level4Style!;
    Level5Style!;
    Level6Style!;
    Level7Style!;
    Level8Style!;
    Legal1Style!;
    Legal2Style!;
    Legal3Style!;
    Legal4Style!;
    Legal5Style!;
    Legal6Style!;
    Legal7Style!;
    Legal8Style!;
    Heading1Style!;
    Heading2Style!;
    Heading3Style!;
    Heading4Style!;
    Heading5Style!;
    Heading6Style!;
    Heading7Style!;
    Heading8Style!;
    MonospacedStyle!;
    CustomHTMLStyle!;
    PreformattedStyle!;
    AddressStyle!;
    QuotationsOutInStyle!;
    DefinitionsOutInStyle!;
    Quotation1Style!;
    Quotation2Style!;
    Quotation3Style!;
    Quotation4Style!;
    Quotation5Style!;
    Quotation6Style!;
    Quotation7Style!;
    Quotation8Style!;
    Definition1Style!;
    Definition2Style!;
    Definition3Style!;
    Definition4Style!;
    Definition5Style!;
    Definition6Style!;
    Definition7Style!;
    Definition8Style!;
    Level9Style!;
    Legal9Style!;
    Heading9Style!;
    Quotation9Style!;
    Definition9Style!});
[Numbering:Enumeration {
  DefNone!;
  DefFollows!;
  DefParentheses!;
  DefFlushRight!;

```

DefLeader!});  
**[Counter:Any {**  
 FigureBox!;  
 TableBox!;  
 TextBox!;  
 UserBox!;  
 EquationBox!;  
 ButtonBox!;  
 WatermarkImageBox!;  
 InlineEquationBox!;  
 NoBox!});  
**[NumberingFormat:String]**  
 [Create a list definition]

**ListDefinitionDelete (ListName:String)**  
 [Delete a list definition]

**ListDefinitionEditBox (**  
 Name:String;  
 BoxName:Any {  
 FigureBox!;  
 TableBox!;  
 TextBox!;  
 UserBox!;  
 EquationBox!;  
 ButtonBox!;  
 WatermarkImageBox!;  
 InlineEquationBox!;  
 NoBox!})  
 [Change the box type of a list definition]

**ListDefinitionMode (**  
 ListName:String;  
**[Numbering:Enumeration {**  
 DefNone!;  
 DefFollows!;  
 DefParentheses!;  
 DefFlushRight!;  
 DefLeader!})  
 [Change the numbering mode of a list definition]

**ListDefinitionRename (OldName:String; NewName:String)**  
 [Rename a list definition]

**ListDefinitionRetrieve (Filename:String; ListName:String)**  
 [Retrieve a list definition from a document]

**ListDefinitionStyle (**  
 ListName:String;  
 Style:Any {  
 NormalStyle!;  
 FootnoteNumberInDocumentStyle!;  
 EndnoteNumberInDocumentStyle!;  
 FigureNumberStyle!;  
 TableBoxNumberStyle!;  
 TextBoxNumberStyle!;  
 UserBoxNumberStyle!;  
 EquationNumberStyle!;  
 HypertextStyle!;  
 ToC1Style!;  
 ToC2Style!;  
 ToC3Style!;  
 ToC4Style!;  
 ToC5Style!;  
 Index1Style!;  
 Index2Style!;  
 ListStyle!;  
 ToAStyle!;  
 DocStyle!;  
 BoxTextStyle!;  
 CaptionStyle!;  
 FootnoteStyle!;  
 EndnoteStyle!;  
 CommentStyle!;  
 HeaderAStyle!;  
 HeaderBStyle!;  
 FooterAStyle!;  
 FooterBStyle!;  
 WatermarkAStyle!;  
 WatermarkBStyle!;  
 ParagraphOutInStyle!;  
 OutlineStyle!;  
 LegalOutInStyle!;  
 BulletsOutInStyle!;  
 HeadingsOutInStyle!;  
 Legal\_2OutInStyle!;  
 NumbersOutInStyle!;  
 Level1Style!;  
 Level2Style!;  
 Level3Style!;  
 Level4Style!;  
 Level5Style!;  
 Level6Style!;  
 Level7Style!;  
 Level8Style!;  
 Legal1Style!;  
 Legal2Style!;  
 Legal3Style!;  
 Legal4Style!;  
 Legal5Style!;  
 Legal6Style!;  
 Legal7Style!;  
 Legal8Style!;  
 Heading1Style!;  
 Heading2Style!;  
 Heading3Style!;  
 Heading4Style!;  
 Heading5Style!;  
 Heading6Style!;  
 Heading7Style!;  
 Heading8Style!;  
 MonospacedStyle!;  
 CustomHTMLStyle!;  
 PreformattedStyle!;  
 AddressStyle!;  
 QuotationsOutInStyle!;  
 DefinitionsOutInStyle!;  
 Quotation1Style!;  
 Quotation2Style!;  
 Quotation3Style!;  
 Quotation4Style!;  
 Quotation5Style!;  
 Quotation6Style!;  
 Quotation7Style!;  
 Quotation8Style!;  
 Definition1Style!;  
 Definition2Style!;  
 Definition3Style!;  
 Definition4Style!;  
 Definition5Style!;  
 Definition6Style!;  
 Definition7Style!;  
 Definition8Style!;  
 Level9Style!;  
 Legal9Style!;  
 Heading9Style!;  
 Quotation9Style!;  
 Definition9Style!})  
 [Edit a list definition and change the list style]

EndnoteStyle!;  
 CommentStyle!;  
 HeaderAStyle!;  
 HeaderBStyle!;  
 FooterAStyle!;  
 FooterBStyle!;  
 WatermarkAStyle!;  
 WatermarkBStyle!;  
 ParagraphOutInStyle!;  
 OutlineStyle!;  
 LegalOutInStyle!;  
 BulletsOutInStyle!;  
 HeadingsOutInStyle!;  
 Legal\_2OutInStyle!;  
 NumbersOutInStyle!;  
 Level1Style!;  
 Level2Style!;  
 Level3Style!;  
 Level4Style!;  
 Level5Style!;  
 Level6Style!;  
 Level7Style!;  
 Level8Style!;  
 Legal1Style!;  
 Legal2Style!;  
 Legal3Style!;  
 Legal4Style!;  
 Legal5Style!;  
 Legal6Style!;  
 Legal7Style!;  
 Legal8Style!;  
 Heading1Style!;  
 Heading2Style!;  
 Heading3Style!;  
 Heading4Style!;  
 Heading5Style!;  
 Heading6Style!;  
 Heading7Style!;  
 Heading8Style!;  
 MonospacedStyle!;  
 CustomHTMLStyle!;  
 PreformattedStyle!;  
 AddressStyle!;  
 QuotationsOutInStyle!;  
 DefinitionsOutInStyle!;  
 Quotation1Style!;  
 Quotation2Style!;  
 Quotation3Style!;  
 Quotation4Style!;  
 Quotation5Style!;  
 Quotation6Style!;  
 Quotation7Style!;  
 Quotation8Style!;  
 Definition1Style!;  
 Definition2Style!;  
 Definition3Style!;  
 Definition4Style!;  
 Definition5Style!;  
 Definition6Style!;  
 Definition7Style!;  
 Definition8Style!;  
 Level9Style!;  
 Legal9Style!;  
 Heading9Style!;  
 Quotation9Style!;  
 Definition9Style!})  
 [Edit a list definition and change the list style]

**ListEditPageFormat (ListName:String; NumberFormat:Any {**  
 DocFormat!})  
 [Edit the page numbering format of a list definition]

**ListMark** ([ListName:String])

[Mark the selected text as an item in the current list]

– M –

**MacroCommandInserter** (State:Enumeration {Off!;On!})

[Turn the Macro Command Inserter on or off]

**MacroConverterShippingMacro** ()

[Convert a 5.2 macro to a 6.0 macro]

**MacroEditControlBar** ([State:Enumeration {Off!;On!}])

[Display the macro toolbar to edit macros]

**MacroExecuteQuick** ()

[Play back the current QuickMacro]

**MacroFileChain** (Filename:String)

[Specify a macro to play upon completion of the current macro]

**MacroFilePlay** (Filename:String)

[Play a macro]

**MacroMenuAdd** (

UpdateBIF:Enumeration {No!; Yes!};

MenuPos:Numeric;

Location:Enumeration {

FileOnDisk!;

CurrentTemplate!;

StandardTemplate!};

[FileName:String];

[TemplateName:String])

[Add a macro to the macro submenu]

**MacroMenuDelete** (

UpdateBIF:Enumeration {No!; Yes!};

{[MenuPos:Numeric]})

[Delete a macro from a menu]

**MacroPause** ()

[Pause while recording/executing a macro]

**MacroStatusPrompt** (

State:Enumeration {Off!; On!};

[Prompt:String])

[Turn the macro status prompt line on or off]

**MacroTemplatePlay** (

Location:Enumeration {

DefaultTemplate!;

CurrentTemplate!};

MacroName:String)

[Play a macro located on a template]

**MailDlg** ()

[Send a selection as an e-mail message or document as an attachment]

**MakeItFit** (

TargetPage:Numeric;

{[Adjust:Enumeration {

FitLeftMargin!;

FitRightMargin!;

FitTopMargin!;

FitBottomMargin!;

FitFontSize!;

FitLineSpacing!}}])

[Shrink or expand a document to fill a specified number of pages]

**MakeItFitDlg** ()

[Shrink or expand the document to fill a specified number of pages]

**Numeric Measurement := MarginAdjustLeft**

([AmtToAdjust:Numeric Measurement])

[Move the left margin to the right or left]

**Numeric Measurement := MarginAdjustRight**

([AmtToAdjust:Numeric Measurement])

[Move the right margin to the right or left]

**Numeric Measurement := MarginBottom** ([MarginWidth:Numeric Measurement])

[Set the bottom margin]

**Numeric Measurement := MarginLeft** ([MarginWidth:Numeric Measurement])

[Set the left margin]

**MarginReleaseInsert** ()

[Insert a hard back tab (margin release) into the document]

**Numeric Measurement := MarginRight** ([MarginWidth:Numeric Measurement])

[Set the right margin]

**Numeric Measurement := MarginTop** ([MarginWidth:Numeric Measurement])

[Set the top margin]

**MarkCrossRef** ()

[Mark cross-references]

**MarkCrossRefUpdateList** (Target:String)

[Add a specified target name to the Target name list]

**MarkIndex** ()

[Define and mark entries for an index]

**MarkIndexUpdateList** (Heading:String; SubHeading:String)

[Create a reference to a specified heading and subheading in an index]

**MarkList** ()

[Define and mark items for a list]

**MarkListUpdateList** (ListName:String)

[Create a reference to a specified list name]

**MarkTableOfAuthorities** ()

[Define, mark items, and create a table of authorities]

**MarkTableOfContents** ()

[Define, mark items, and create a table of contents]

**MarkTOAUpdateList** (ShortForm:String)

[Add a name to the Table of Authorities Short form list]

**MasterDocCondense** (

[Filename:String];

Action:Enumeration {SkipSub!; SaveSub!})

[Condense selected subdocuments and save changes]

**MasterDocCondenseDlg** ()

[Select subdocuments to close in a master document]

**MasterDocExpand** ([Filename:String])

[Expand a subdocument]

**MasterDocSubdocDlg** ()

[Select a file to include in a master document]

**MatchElement** (

[Attribute:String];

[Value:String];

Type:Enumeration {Start!; End!};



**[EditAttribute:Enumeration {No!; Yes!}]**  
[Specify element-specific constraints for a search]

**MatchExtendSelection ()**  
[Select the text between the insertion point and the matched text of a Find and Replace search]

**MatchFindInElements (**  
  **State:Enumeration {No!; Yes!};**  
  **[Action:Enumeration {**  
    **ListedOrder!;**  
    **AnyOrder!};**  
  **[[Element:String]])**  
[Find the selection only if the match is in XML elements]

**MatchLimit (**  
  **State:Enumeration {No!; Yes!};**  
  **[LimitValue:Numeric]**  
[Limit a Replace All search to a specified number of matches]

**MatchPositionAfter ()**  
[Place the insertion point after matched text in a Find and Replace search]

**MatchPositionBefore ()**  
[Place the insertion point before matched text in a Find and Replace search]

**MatchSelection ()**  
[Select matched text in a Find and Replace search]

**MatchSkipInElements ([[Element:String]])**  
[Do not find matches in XML elements]

**MatchWithAttributes (**  
  **State:Enumeration {No!; Yes!};**  
  **[[Attribute:Enumeration {**  
    **ExtraLarge!;**  
    **VeryLarge!;**  
    **Large!;**  
    **Small!;**  
    **Fine!;**  
    **Superscript!;**  
    **Subscript!;**  
    **Outline!;**  
    **Italics!;**  
    **Shadow!;**  
    **Redline!;**  
    **DoubleUnderline!;**  
    **Bold!;**  
    **Strikeout!;**  
    **Underline!;**  
    **SmallCaps!}]])**  
[Match only text with specified attributes in a Find and Replace search]

**MatchWithFont (**  
  **State:Enumeration {No!; Yes!};**  
  **[Typeface:String];**  
  **[Family:Enumeration {**  
    **FamilyITCBenguiat!;**  
    **FamilyTimesRoman!;**  
    **FamilyITCGaramon!;**  
    **FamilyCenturySchoolbook!;**  
    **FamilyBodoni!;**  
    **FamilyITCLubalinGraph!;**  
    **FamilyAlbertus!;**  
    **FamilyCourier!;**  
    **FamilyHelvetica!;**  
    **FamilyITCRonda!;**  
    **FamilyMicrostyle!;**  
    **FamilyOptima!;**  
    **FamilyLetterGothic!;**

**FamilyBrush!;**  
    **FamilyParkAvenue!;**  
    **FamilySignetRoundhand!;**  
    **FamilyTekton!;**  
    **FamilyUncial!;**  
    **FamilyOldEnglish!;**  
    **FamilyITCZapfDingbats!;**  
    **FamilyBroadway!;**  
    **FamilyComputer!;**  
    **FamilyPTBarnum!;**  
    **FamilyPonderosa!;**  
    **FamilyMadrone!;**  
    **FamilyHobol!;**  
    **FamilyRevue!;**  
    **FamilyAestheticOrnamented!});**  
  **[Attributes:Enumeration {**  
    **FontMatchNormal!;**  
    **FontMatchItalic!;**  
    **FontMatchOutline!;**  
    **FontMatchShadow!;**  
    **FontMatchSmallCaps!});**  
  **[Weight:Enumeration {**  
    **WeightUnknown!;**  
    **WeightUltraThin!;**  
    **WeightUltraLight!;**  
    **WeightExtraThin!;**  
    **WeightThin!;**  
    **WeightExtraLight!;**  
    **WeightLight!;**  
    **WeightDemiLight!;**  
    **WeightSemiLight!;**  
    **WeightBook!;**  
    **WeightRegular!;**  
    **WeightRoman!;**  
    **WeightMedium!;**  
    **WeightDemiBold!;**  
    **WeightSemiBold!;**  
    **WeightBold!;**  
    **WeightExtraBold!;**  
    **WeightHeavy!;**  
    **WeightExtraHeavy!;**  
    **WeightBlack!;**  
    **WeightExtraBlack!;**  
    **WeightUltraHeavy!;**  
    **WeightUltraBlack!;**  
    **WeightMaximum!});**  
  **[Width:Enumeration {**  
    **WidthUnknown!;**  
    **WidthUltraCondensed!;**  
    **WidthUltraCompressed!;**  
    **WidthCompressed!;**  
    **WidthExtraCondensed!;**  
    **WidthCondensed!;**  
    **WidthSemiCondensed!;**  
    **WidthNormal!;**  
    **WidthSemiExpanded!;**  
    **WidthExpanded!;**  
    **WidthExtraExpanded!;**  
    **WidthUltraExpanded!;**  
    **WidthDoubleWide!;**  
    **WidthTripleWide!});**  
  **[Source:Enumeration {**  
    **PRSTFile!;**  
    **DRSFile!});**  
  **[Type:Enumeration {**  
    **Speedo!;**  
    **IntelliFont!;**  
    **Type1!;**  
    **TrueType!});**  
  **[CharacterSet:Enumeration {**  
    **FontMatchASCII!;**  
    **FontMatchMultinational1!;**

**FontMatchMultinational2!;**  
**FontMatchBoxDrawing!;**  
**FontMatchTypographicSymbols!;**  
**FontMatchIconicSymbols!;**  
**FontMatchMath!;**  
**FontMatchMathExtension!;**  
**FontMatchGreek!;**  
**FontMatchHebrew!;**  
**FontMatchCyrillic!;**  
**FontMatchJapaneseKana!;**  
**FontMatchUserDefined!;**  
**FontMatchArabic!;**  
**FontMatchArabicScript!}}**

[Match only text with specified font attributes in a Find and Replace search]

**MatchWithFontSize (**  
**State:Enumeration {No!; Yes!};**  
**[FontSize:Numeric Measurement])**

[Find and Replace Text matches only text with the specified font size]

**Boolean := MenuAddFeature (**  
**Menu:String;**  
**Name:Command Name;**  
**[Offset:Numeric])**

[Add a feature to the current WordPerfect menu]

**Boolean := MenuAddMacro (**  
**Menu:String;**  
**Name:String;**  
**MacroName:String;**  
**[Hint:String];**  
**[Offset:Numeric])**

[Add a macro to the current WordPerfect menu]

**Boolean := MenuAddProgram (**  
**Menu:String;**  
**Name:String;**  
**ProgramName:String;**  
**[Hint:String];**  
**[Offset:Numeric])**

[Add a program to the current WordPerfect menu]

**Boolean := MenuAddScript (**  
**Menu:String;**  
**Name:String;**  
**Script:String;**  
**[Hint:String];**  
**[Offset:Numeric])**

[Add a keyboard script to the current WordPerfect menu]

**Boolean := MenuAddSeparator (**  
**Menu:String;**  
**[Offset:Numeric])**

[Add a separator to the current WordPerfect menu]

**Boolean := MenuAddSubMenu (**  
**Menu:String;**  
**Name:String;**  
**[Hint:String];**  
**[Offset:Numeric])**

[Add a submenu to the current WordPerfect menu]

**Boolean := MenuDeleteItemEx (**  
**Menu:String;**  
**[Name:String];**  
**[Offset:Numeric])**

[Delete an item from the current WordPerfect menu]

**MenuSelect (**  
**MenuName:String;**  
**[LoadedFromDefaultTemplate:Enumeration {No!; Yes!}])**

[Choose a menu]

**MenuSelectDlg ()**

[Create or edit a menu bar]

**MergeAddressBook (AddressBook:String)**

[Specify an address book to associate with the current form file]

**MergeAddressBookSaveAs (**

**OutputFile:String;**  
**[AddressBook:String])**

[Save address book records as a merge data file]

**MergeBlankField (State:Enumeration {Leave!; Remove!})**

[Specify whether to leave or remove a blank field]

**MergeCode (**

**Codes:Enumeration {**

**Beep!;**

**Break!;**

**CancelOff!;**

**CancelOn!;**

**Continue!;**

**Date!;**

**Default!;**

**DisplayStop!;**

**Else!;**

**EndSwitch!;**

**EndField!;**

**EndFor!;**

**EndIf!;**

**EndRecord!;**

**EndWhile!;**

**Next!;**

**NextRecord!;**

**PageOff!;**

**PageOn!;**

**Print!;**

**Quit!;**

**ProcessOn!;**

**Return!;**

**ReturnCancel!;**

**ReturnError!;**

**Rewrite!;**

**Step Off!;**

**StepOn!;**

**Stop!;**

**ProcessOff!;**

**RepeatRow!;**

**Assign!;**

**Call!;**

**Caps!;**

**Switch!;**

**CaseOf!;**

**ChainMacro!;**

**ChainData!;**

**ChainForm!;**

**Char!;**

**Codes!;**

**Comment!;**

**CtoN!;**

**Document!;**

**EmbedMacro!;**

**Field!;**

**FieldNames!;**

**FirstCap!;**

**ForNext!;**

**ForEach!;**

**GetString!;**

**Go!;**

**If!;**

**IfBlank!;**

**IfExists!;**

**IfNotBlank!;**

**Insert!;**  
**Keyboard!;**  
**Label!;**  
**AssignLocal!;**  
**Look!;**  
**MrgCmd!;**  
**NestData!;**  
**NestForm!;**  
**NestMacro!;**  
**NtoC!;**  
**OnCancel!;**  
**OnError!;**  
**Prompt!;**  
**StatusPrompt!;**  
**StrLen!;**  
**StrPos!;**  
**SubstData!;**  
**SubstForm!;**  
**SubStr!;**  
**System!;**  
**ToLower!;**  
**ToUpper!;**  
**Variable!;**  
**Wait!;**  
**While!;**  
**PostNet!;**  
**[String:String]**  
 [Insert a merge code]

**MergeCodesDisplayEdit (Display:Enumeration {**  
**Hide!;**  
**Show!;**  
**Marker!})**  
 [Display or hide merge codes, or display them as markers]

**MergeCodesDisplayRun (Display:Enumeration {**  
**Hide!;**  
**Show!;**  
**Marker!})**  
 [Display or hide merge codes, or display them as markers in a merged document]

**MergeCodesDlg ()**  
 [Insert merge codes into the document]

**MergeCondition (**  
**[Condition1:String];**  
**[Condition2:String];**  
**[Condition3:String];**  
**[Condition4:String];**  
**[Condition5:String];**  
**[Condition6:String];**  
**[Condition7:String];**  
**[Condition8:String];**  
**[Condition9:String];**  
**[Condition10:String];**  
**[Condition11:String];**  
**[Condition12:String];**  
**[Condition13:String];**  
**[Condition14:String];**  
**[Condition15:String])**  
 [Define a condition to select records from a data file]

**MergeDataFile ([DataFile:String])**  
 [Specify the data file to associate with a form file]

**MergeDlg ()**  
 [Create and combine data files and form files]

**MergeEndField ()**  
 [End a field in a merge data file]

**MergeEndRecord ()**  
 [End a record in a merge data file]

**MergeEnvelope (State:Enumeration {Off!;On!})**  
 [Create an envelope for merged records]

**MergeEnvelopeDefinitionBegin ()**  
 [Begin a merge envelope definition]

**MergeEnvelopeDefinitionEnd ()**  
 [End a merge envelope definition]

**MergeFileAssociate (FormFile:String; [DataFile:String])**  
 [Associate form and data files]

**MergeFileAssociateAddressBook (**  
**FormFile:String;**  
**[AddressBook:String])**  
 [Associate a form file with an address book]

**MergeFileAssociateODBC (**  
**FormFile:String;**  
**[DataSource:String];**  
**[TableName:String])**  
 [Associate a form file with an ODBC data source and table]

**MergeFilesDlg ()**  
 [Combine data with a form file]

**MergeFileType (FileType:Enumeration {**  
**None!;**  
**Form!;**  
**TableData!;**  
**TextData!})**  
 [Specify the file type to merge]

**MergeFormDlg ()**  
 [Display the Create Merge File dialog box]

**MergeMark (RecordNum:Numeric)**  
 [Mark a record to include in a merge]

**MergeODBC (DataSource:String; TableName:String)**  
 [Specify an ODBC data source and table to associate with the current form file]

**MergePageBreak (State:Enumeration {Off!;On!})**  
 [Separate merged documents with a page break]

**MergePrintText ()**  
 [Print a data file without page breaks between records]

**MergeRepeat (NumberToRepeat:Numeric)**  
 [Specify the number of times to merge each record]

**MergeRun (**  
**FormFileType:Enumeration {**  
**Current!;**  
**Clipboard!;**  
**FormFile!};**  
**[FormFile:String];**  
**DataFileType:Enumeration {**  
**NoData!;**  
**Clipboard!;**  
**DataFile!;**  
**AddressBook!;**  
**ODBC!};**  
**[DataFile:String];**  
**OutputFileType:Enumeration {**  
**ToCurrentDoc!;**  
**ToNewDoc!;**  
**ToFile!;**  
**ToPrinter!;**

ToEMail!;  
[OutputFile:String];  
[AddressBook:String];  
[EmailAddress:String];  
[SubjectLine:String];  
[DataSource:String];  
[TableName:String])  
[Merge a form file and a data source]

**MergeSelect** (  
    SelectType:Enumeration {  
        All!;  
        Marked!;  
        Range!};  
    [Begin:Numeric];  
    [End:Numeric])  
[Specify the records to merge]

**MergeTableDataDlg** ()  
[Display the Create Data File dialog box]

**MergeTextDataDlg** ()  
[Display the Create Merge File dialog box]

**MiniBrowser** ()  
[Launches the Corel Connector]

**MinusSign** ()  
[Insert a hyphen character]

**MoveModeEnd** ()  
[End the command sequence to move text]

- N -

**NewHFWOccurrence** (Occurrence:Enumeration {  
    NoPages!;  
    OddPages!;  
    EvenPages!;  
    AllPages!})  
[Reset the pages on which headers, footers, and watermarks occur]

**NumberingOptionDlg** ()  
[Specify the position and appearance of page numbers, set the number]

- O -

**ObjectImageArrow** (  
    [HeadStyle:Any];  
    [TailStyle:Any])  
[Create an arrow object]

**ObjectImageCreate** ()  
[Create an image object]

**ObjectImageDropShadow** (  
    [Location:Enumeration {  
        NoShadow!;  
        UpperLeft!;  
        LowerLeft!;  
        LowerRight!;  
        UpperRight!});  
    [XSize:Numeric Measurement];  
    [YSize:Numeric Measurement];  
    [ColorName:String];  
    [Red:Numeric];  
    [Green:Numeric];  
    [Blue:Numeric];  
    [Shade:Numeric];  
    [VerticalSize:Numeric Measurement])  
[Create a drop shadow for an image object]

**ObjectImageLineColor** ([RGBValue:Numeric])  
[Change line color of image object]

**ObjectImageLineStyle** (Style:Any {  
    NoLine!;  
    SolidLine!;  
    XLDash!;  
    LDash!;  
    Dash!;  
    SDash!;  
    CloseDot!;  
    Dot!;  
    TripleDot!;  
    LDashDot!;  
    LDashDotDot!;  
    DashDashDot!;  
    DashDashDotDot!;  
    LDashDash!;  
    LDashSDashSDash!;  
    XLDashXLDashDash!;  
    XLDashXLDashSDashSDash!})  
[Change line style of image object]

**ObjectImageLineThickness** (Thickness:Any {AutoThickness!})  
[Change line thickness of image object]

**ObjectImagePolyline** ()  
[Create a polyline image object]

**ObjectImageRemove** ()  
[Remove an image object]

**ObjectImageRotation** (Degrees:Numeric)  
[Rotate an image object]

**OLE1InsertDlg** ()  
[Insert an OLE object]

**OLE1LinksDlg** ()  
[View or edit OLE or DDE links]

**OLE1PasteDlg** ()  
[Insert the Clipboard contents in a specified data type]

**OLEAcceptLinks** ()  
[Accept a link updated or initialized between a server and client application]

**OLECancelLink** (LinkName:String)  
[Cancel a link to an object]

**OLEChangeLink** (ObjectName:String; Filename:String)  
[Change the file in a link]

**OLEDeactivate** ()  
[Deactivate an OLE object that is being edited in place]

**OLEInitializeLinks** ()  
[Reinitialize a link with its server application]

**OLEInsertDlg** ()  
[Display the Insert Object dialog box]

**OLEInsertObject** (  
    ObjectType:String;  
    [DisplayMode:Enumeration {  
        DisplayNormal!;  
        DisplayAsIcon!})])  
[Specify the name of a program to insert an object into]

**OLEInsertObjectFile** (  
    Filename:String;  
    [DisplayMode:Enumeration {

**DisplayNormal!**;  
**DisplayAsIcon!**];  
**[ObjectMode:Enumeration {**  
  **Embedded!**;  
  **Linked!}]**)  
[Insert an object file into the active document]

**OLELinksDlg ()**  
[Display the Links dialog box]

**OLEObject ()**  
[The OLE option for the selected object]

**OLEPasteDlg ()**  
[Display the Paste Special dialog box]

**OLESetLinkUpdate (**  
  **ObjectName:String;**  
  **UpdateType:Enumeration {**  
    **UpdateAlways!**;  
    **UpdateManually!}]**)  
[Specify the update options for linked objects]

**OLEUpdateLink (LinkName:String)**  
[Update a specified link]

**OLEVerb (Verb:Numeric)**  
[Perform an action on an OLE object]

**OnlineHelpSupport ()**  
[Get help from the WordPerfect 10 home page]

**OutlineApply (**  
  **[OutlineName:String];**  
  **SysOrUser:Enumeration {**  
    **UserOutlineStyle!**;  
    **SystemOutlineStyle!};**  
  **[SystemStyleId:Numeric];**  
  **CreateOrResume:Enumeration {**  
    **ResumeOutline!**;  
    **CreateOutline!**;  
    **EmbedOutline!};**  
  **Prompt:Enumeration {No!; Yes!};**  
  **[TabEvery:Numeric Measurement])**  
[Apply outline definition to document]

**OutlineBarDisplay (State:Enumeration {Off!;On!})**  
[Turn on Outline mode and display the Outline feature bar]

**OutlineBarSetup ([State:Enumeration {Off!;On!}])**  
[Display or hide outline icons in Page view]

**OutlineBeginNew (Style:Enumeration {**  
  **ParagraphOutInStyle!**;  
  **OutlineStyle!**;  
  **LegalOutInStyle!**;  
  **BulletsOutInStyle!**;  
  **HeadingsOutInStyle!**;  
  **Legal\_2OutInStyle!**;  
  **NumbersOutInStyle!**;  
  **QuotationsOutInStyle!**;  
  **DefinitionsOutInStyle!**;  
  **Legal\_3OutInStyle!**;  
  **Legal\_4OutInStyle!})**  
[Begin a new outline in a specified style]

**OutlineBodyText ([State:Enumeration {**  
  **ShowBodyText!**;  
  **HideBodyText!}])**  
[Display or hide the outline body text]

**OutlineBodyTextToggle ()**  
[Apply outline to body text]

**OutlineChangeToBodyText ()**  
[Change the active outline item to body text]

**OutlineDown ()**  
[Move the insertion point to the next outline item or paragraph of body text]

**OutlineEnd ()**  
[Turn off the outline]

**OutlineFamilyCopy ()**  
[Copy the active outline family items to the Clipboard]

**OutlineFamilyCut ()**  
[Delete the active outline family items and adjust numbering]

**OutlineFamilyDemote ()**  
[Demote the current outline family items to the next level and adjust the numbering]

**OutlineFamilyDown ()**  
[Move the insertion point to the next outline family]

**OutlineFamilyDragDown ()**  
[Move the active family items below the next family and adjust numbering]

**OutlineFamilyDragUp ()**  
[Move the active outline family items above the previous family and adjust numbering]

**OutlineFamilyHide (State:Enumeration {**  
  **ShowBodyText!**;  
  **HideBodyText!})**  
[Hide all but the current level of the outline family]

**OutlineFamilyLeft ()**  
[Move the insertion point to the previous family]

**OutlineFamilyMove ()**  
[Move an outline family]

**OutlineFamilyPromote ()**  
[Raise the active outline family items one level]

**OutlineFamilyRight ()**  
[Move the insertion point to the next outline item]

**OutlineFamilySelect ()**  
[Select the active outline family]

**OutlineFamilyShow (State:Enumeration {**  
  **ShowBodyText!**;  
  **HideBodyText!})**  
[Show all levels of the outline family]

**OutlineFamilyUp ()**  
[Move the insertion point to the beginning of the previous outline family]

**OutlineHideShowingBodyText ()**  
[Hide all outline items and display the body text]

**OutlineIconsDisplay (State:Enumeration {Off!;On!})**  
[Display the outline level icons]

**OutlineInsertParagraphNumber (**  
  **Level:Numeric;**  
  **InsertType:Enumeration {**  
    **Fixed!**;  
    **Auto!}])**  
[Insert a paragraph number code into the document]

**OutlineInsertParagraphNumberDlg ()**

[Insert a paragraph number code into the document]

**OutlineNumberDlg ()**

[Set the paragraph number to a specific value]

**OutlineOff ()**

[Close the Outline Feature Bar]

**OutlineParagraphCopy ()**

[Copy the active outline item to the Clipboard]

**OutlineParagraphDelete ()**

[Delete the active outline item and adjust the numbering]

**OutlineParagraphDemote ()**

[Change the active outline item to the next level]

**OutlineParagraphDragDown ()**

[Move the current or selected item or family down; keep the same level]

**OutlineParagraphDragUp ()**

[Move the active or selected item or family up; keep the same level]

**OutlineParagraphMove ()**

[Delete the active outline item and adjust the numbering]

**OutlineParagraphNumber ([Level:Numeric])**

[Insert a paragraph number code into the document]

**OutlineParagraphNumberInsert ()**

[Insert a paragraph number code into the document]

**OutlineParagraphPromote ()**

[Change the active outline item to the previous level]

**OutlineParagraphSelect ()**

[Select the active outline paragraph]

**OutlineRight ()**

[Move the insertion point to the next outline item]

**OutlineShow (**

**Level:Numeric;**  
**State:Enumeration {**  
**ShowBodyText!;**  
**HideBodyText!})**

[Specify the lowest outline level to display and whether to show body text]

**OutlineStyleCopy (**

**Style:Any {**  
**ParagraphOutInStyle!;**  
**OutlineStyle!;**  
**LegalOutInStyle!;**  
**BulletsOutInStyle!;**  
**HeadingsOutInStyle!;**  
**Legal\_2OutInStyle!;**  
**NumbersOutInStyle!;**  
**QuotationsOutInStyle!;**  
**DefinitionsOutInStyle!;**  
**Legal\_3OutInStyle!;**  
**Legal\_4OutInStyle!};**  
**[Source:Enumeration {**  
**CurrentDoc!;**  
**PersonalLibrary!;**  
**SharedLibrary!});**  
**[Destination:Enumeration {**  
**CurrentDoc!;**  
**PersonalLibrary!;**  
**SharedLibrary!});**

**[NewName:String])**

[Copy an outline style]

**OutlineStyleCreate (**

**Style:String;**  
**[Library:Enumeration {**  
**CurrentDoc!;**  
**PersonalLibrary!;**  
**SharedLibrary!});**

[Create, name, and save an outline style]

**OutlineStyleDelete (**

**Style:Any {**  
**ParagraphOutInStyle!;**  
**OutlineStyle!;**  
**LegalOutInStyle!;**  
**BulletsOutInStyle!;**  
**HeadingsOutInStyle!;**  
**Legal\_2OutInStyle!;**  
**NumbersOutInStyle!;**  
**QuotationsOutInStyle!;**  
**DefinitionsOutInStyle!;**  
**Legal\_3OutInStyle!;**  
**Legal\_4OutInStyle!};**  
**[Library:Enumeration {**  
**CurrentDoc!;**  
**PersonalLibrary!;**  
**SharedLibrary!});**

[Delete an outline style]

**OutlineStyleDescription (Description:String)**

[Create an outline style description]

**OutlineStyleEditBegin (**

**Style:Any {**  
**ParagraphOutInStyle!;**  
**OutlineStyle!;**  
**LegalOutInStyle!;**  
**BulletsOutInStyle!;**  
**HeadingsOutInStyle!;**  
**Legal\_2OutInStyle!;**  
**NumbersOutInStyle!;**  
**QuotationsOutInStyle!;**  
**DefinitionsOutInStyle!;**  
**Legal\_3OutInStyle!;**  
**Legal\_4OutInStyle!};**  
**[Library:Enumeration {**  
**CurrentDoc!;**  
**PersonalLibrary!;**  
**SharedLibrary!});**

[Begin editing of an outline style]

**OutlineStyleEditEnd (State:Enumeration {Cancel!;Save!})**

[Save outline style changes]

**OutlineStyleLeadingZeros (Level:Numeric; Zeros:Numeric)**

[Specify the number of leading zeros for each level of a selected outline style]

**OutlineStyleLevel (**

**Level:Numeric;**  
**Style:Any {**  
**NormalStyle!;**  
**FootnoteNumberInDocumentStyle!;**  
**EndnoteNumberInDocumentStyle!;**  
**FigureNumberStyle!;**  
**TableBoxNumberStyle!;**  
**TextBoxNumberStyle!;**  
**UserBoxNumberStyle!;**  
**EquationNumberStyle!;**  
**HypertextStyle!;**  
**ToC1Style!;**  
**ToC2Style!;**

ToC3Style!  
 ToC4Style!  
 ToC5Style!  
 Index1Style!  
 Index2Style!  
 ListStyle!  
 ToAStyle!  
 DocStyle!  
 BoxTextStyle!  
 CaptionStyle!  
 FootnoteStyle!  
 EndnoteStyle!  
 CommentStyle!  
 HeaderAStyle!  
 HeaderBStyle!  
 FooterAStyle!  
 FooterBStyle!  
 WatermarkAStyle!  
 WatermarkBStyle!  
 ParagraphOutInStyle!  
 OutlineStyle!  
 LegalOutInStyle!  
 BulletsOutInStyle!  
 HeadingsOutInStyle!  
 Legal\_2OutInStyle!  
 NumbersOutInStyle!  
 Level1Style!  
 Level2Style!  
 Level3Style!  
 Level4Style!  
 Level5Style!  
 Level6Style!  
 Level7Style!  
 Level8Style!  
 Legal1Style!  
 Legal2Style!  
 Legal3Style!  
 Legal4Style!  
 Legal5Style!  
 Legal6Style!  
 Legal7Style!  
 Legal8Style!  
 Heading1Style!  
 Heading2Style!  
 Heading3Style!  
 Heading4Style!  
 Heading5Style!  
 Heading6Style!  
 Heading7Style!  
 Heading8Style!  
 MonospacedStyle!  
 CustomHTMLStyle!  
 PreformattedStyle!  
 AddressStyle!  
 QuotationsOutInStyle!  
 DefinitionsOutInStyle!  
 Quotation1Style!  
 Quotation2Style!  
 Quotation3Style!  
 Quotation4Style!  
 Quotation5Style!  
 Quotation6Style!  
 Quotation7Style!  
 Quotation8Style!  
 Definition1Style!  
 Definition2Style!  
 Definition3Style!  
 Definition4Style!  
 Definition5Style!  
 Definition6Style!  
 Definition7Style!  
 Definition8Style!

Level9Style!  
 Legal9Style!  
 Heading9Style!  
 Quotation9Style!  
 Definition9Style!

[Specify an outline level and style]

**OutlineStyleNumberFormat** (Level:Numeric;Format:String)  
 [Specify the numbering format for each level of a selected outline style]

**OutlineStyleNumbersOnly** (State:Enumeration {No!;Yes!})  
 [Apply the Numbers Only format to a selected outline]

**OutlineStyleRename** (Name:String)  
 [Rename a user-defined outline style]

**OutlineStyleRetrieve** (  
 Filename:String;  
 Type:Enumeration {  
 UserStyles!  
 SystemStyles!  
 AllStyles!};  
 [Library:Enumeration {  
 CurrentDoc!  
 PersonalLibrary!  
 SharedLibrary!}])  
 [Retrieve an outline style]

**OutlineStyleSave** (  
 Filename:String;  
 Type:Enumeration {  
 UserStyles!  
 SystemStyles!  
 AllStyles!};  
 [Library:Enumeration {  
 CurrentDoc!  
 PersonalLibrary!  
 SharedLibrary!}])  
 [Save an outline style]

**OutlineStyleSelect** (Style:Any {  
 ParagraphOutInStyle!  
 OutlineStyle!  
 LegalOutInStyle!  
 BulletsOutInStyle!  
 HeadingsOutInStyle!  
 Legal\_2OutInStyle!  
 NumbersOutInStyle!  
 QuotationsOutInStyle!  
 DefinitionsOutInStyle!  
 Legal\_3OutInStyle!  
 Legal\_4OutInStyle!})  
 [Select an outline definition]

**OutlineStyleType** (  
 Bullets:Enumeration {No!; Yes!};  
 SingleLevel:Enumeration {No!; Yes!})  
 [Specify the outline attributes]

**OutlineUp** ()  
 [Move the insertion point up one outline item up]

**OverstrikeCreate** (OvrStrkStrg:String)  
 [Create an overstrike]

**OverstrikeCreateDlg** ()  
 [Combine two or more characters to create a new character]

**OverstrikeEdit** (OvrStrkStrg:String)  
 [Replace the current overstrike with new text]

**PageBorderDlg ()**

[Select, create, or edit a page border and background]

**PageNumber (Page:Numeric)**

[Specify a new page number for the current page]

**PageNumberDecrement ()**

[Decrease the page number by one]

**PageNumberDisplay ()**

[Insert the page, section, chapter, or volume number]

**PageNumberDisplayFormat ()**

[Insert the page number and accompanying text]

**PageNumberFormat (Format:String)**

[Specify a page number format]

**PageNumberIncrement ()**

[Increase the page number by one]

**PageNumberingAdjustDlg ()**

[Change or reset the page numbering]

**PageNumberingCountDlg ()**

[Count the pages in the document]

**PageNumberingInsertDlg ()**

[Insert the active page number]

**PageNumberMethod (Method:Enumeration {**

**Numbers!;**  
**LowerLetters!;**  
**UpperLetters!;**  
**LowerRoman!;**  
**UpperRoman!})**

[Specify a page numbering method]

**PageNumberPosition (**

**Position:Enumeration {**  
**None!;**  
**TopLeft!;**  
**TopCenter!;**  
**TopRight!;**  
**TopAlternating!;**  
**BottomLeft!;**  
**BottomCenter!;**  
**BottomRight!;**  
**BottomAlternating!;**  
**InsideTopAlternating!;**  
**OutsideBottomAlternating!};**

**Default:Enumeration {**  
**UseDefaultValues!;**  
**DontUseDefaultValues!};**

**[Font:String];**  
**[FontSize:Numeric Measurement];**  
**[Red:Numeric];**  
**[Green:Numeric];**  
**[Blue:Numeric];**  
**[Shade:Numeric];**

**{[Attributes:Enumeration {**  
**ExtraLarge!;**  
**VeryLarge!;**  
**Large!;**  
**Small!;**  
**Fine!;**  
**Superscript!;**  
**Subscript!;**  
**Outline!;**  
**Italics!;**  
**Shadow!;**

**Redline!;**  
**DoubleUnderline!;**  
**Bold!;**  
**Strikeout!;**  
**Underline!;**  
**SmallCaps!;**  
**Every!})}**

[Specify the position and appearance of page numbering]

**PageSetupDlg ()**

[Change the paper size or margins, or divide the page]

**PageSuppressDlg ()**

[Turn off Page Numbering, Header, Watermark, etc., on the current page]

**PageZoomFull ()**

[Switch between Page and Full Page views]

**PaneNext ()**

[Move the focus to the next pane]

**PanePrevious ()**

[Move the focus to the previous pane]

**PaperSizeAdd (**

**Name:String;**  
**Width:Numeric Measurement;**  
**Length:Numeric Measurement;**  
**WidthAdjust:Numeric Measurement;**  
**LengthAdjust:Numeric Measurement;**  
**Flags:Enumeration {**  
**Portrait!;**  
**Landscape!;**  
**PrimaryForm!;**  
**FormCurrentlyLoaded!;**  
**PortraitLoaded!;**  
**LandscapeLoaded!;**  
**PortraitPrimaryLoaded!;**  
**LandscapePrimaryLoaded!};**

**Location:Enumeration {**  
**Continuous!;**  
**Manual!};**

**Type:Enumeration {**  
**AllOthers!;**  
**Standard!;**  
**Bond!;**  
**Letterhead!;**  
**Labels!;**  
**Envelope!;**  
**Transparency!;**  
**CardStock!;**  
**Glossy!;**  
**ClayBased!;**  
**Other!};**

**[Use:Enumeration {**  
**None!;**  
**US!;**  
**Europe!;**  
**Latin\_America!;**  
**Japan!});**  
**[ReturnAddressHoriz:Numeric Measurement];**  
**[ReturnAddressVert:Numeric Measurement];**  
**[MailingAddressHoriz:Numeric Measurement];**  
**[MailingAddressVert:Numeric Measurement]**

[Create a page definition]

**PaperSizeDelete (Name:String)**

[Delete a page definition]

**PaperSizeDlg ()**

[Change the page size and format]



**PaperSizeResave (**

**OldName:String;**  
**NewName:String;**  
**Width:Numeric Measurement;**  
**Length:Numeric Measurement;**  
**WidthAdjustment:Numeric Measurement;**  
**LengthAdjustment:Numeric Measurement;**  
**Flags:Enumeration {**  
   **Portrait!;**  
   **Landscape!;**  
   **PrimaryForm!;**  
   **FormCurrentlyLoaded!;**  
   **PortraitLoaded!;**  
   **LandscapeLoaded!;**  
   **PortraitPrimaryLoaded!;**  
   **LandscapePrimaryLoaded!;**  
**Location:Enumeration {**  
   **Continuous!;**  
   **Manual!;**  
**Type:Enumeration {**  
   **AllOthers!;**  
   **Standard!;**  
   **Bond!;**  
   **Letterhead!;**  
   **Labels!;**  
   **Envelope!;**  
   **Transparency!;**  
   **CardStock!;**  
   **Glossy!;**  
   **ClayBased!;**  
   **Other!;**  
**Use:Enumeration {**  
   **None!;**  
   **US!;**  
   **Europe!;**  
   **Latin\_America!;**  
   **Japan!;**  
**ReturnAddressHoriz:Numeric Measurement;**  
**ReturnAddressVert:Numeric Measurement;**  
**MailingAddressHoriz:Numeric Measurement;**  
**MailingAddressVert:Numeric Measurement)**  
 [Edit and save a page definition]

**PaperSizeSelect (**

**Name:String;**  
**[Width:Numeric Measurement];**  
**[Height:Numeric Measurement];**  
**[Type:Enumeration {**  
   **Standard!;**  
   **Bond!;**  
   **Letterhead!;**  
   **Labels!;**  
   **Envelope!;**  
   **Transparency!;**  
   **Cardstock!;**  
   **Glossy!;**  
   **ClayBased!;**  
   **Other!;**  
**[Orientation:Enumeration {**  
   **Portrait!;**  
   **Landscape!;**  
**}]**  
 [Specify a page size for the current document]

**ParagraphBorderDlg (**

[Select, create, or edit a paragraph border and background]

**ParagraphBreakShippingMacro (**

[Place a paragraph break symbol at the end of the active paragraph]

**ParagraphDown (**

[Move the insertion point to the beginning of the next paragraph]

**ParagraphFormatDlg (**

[Set paragraph spacing, first line indent, and margin adjustments]

**ParagraphHangingIndent (**

[Indent all but the first line of the active paragraph one tab stop]

**ParagraphNumberDefDlg (**

[Change the current outline item to the specified level]

**ParagraphSetNumber (Level:Numeric; StartValue:Numeric)**

[Specify the level and starting value for paragraph numbering]

**ParagraphSpacing (AmtSpc:Numeric; [DotSpc:Numeric Measurement])**

[Specify the spacing between paragraphs]

**ParagraphUp (**

[Move the insertion point to the beginning of the previous paragraph]

**Paste (**

[Paste the Clipboard contents at the insertion point]

**PasteLink (**

**FormatName:String;**  
**FormatId:Numeric;**  
**LinkType:Enumeration {Ddel; Ole!}**  
 [Insert a link to the Clipboard contents]

**PauseCommand (Token:Command Name)**

[Pause a macro until a specified macro command is executed]

**PauseKey (**

**Key:Enumeration {**  
   **Enter!;**  
   **Cancel!;**  
   **Close!;**  
   **Character!;**  
   **Any!;**  
**[Character:String]}**  
 [Pause a macro until a specified key or character is pressed]

**PauseSet (Token:Command Name)**

[Specify a command to replace [HRT] to end a PAUSE]

**PdfDlg (**

[Publish the current document as a (.PDF) file]

**Boolean := PerfectExpert ([State:Enumeration {Off!; On!}])**

[Display the PerfectExpert]

**Boolean := PersonalInfoDlg (**

[Create, edit, and set default personal information for templates]

**String := PersonalInfoFormatAddr (Format:String)**

[Retrieve a formatted version of the current Personal Information]

**String := PersonalInfoGetField (Field:Any)**

[Retrieves a field from the current Personal Information]

**PlayClipboardShippingMacro (**

[Play text from the Clipboard (requires voice software)]

**PlayFileShippingMacro (**

[Play text from a file (requires voice software)]

**PlaySelectionShippingMacro (**

[Play selected text (requires voice software)]

**PleadingShippingMacro (**

[Create lines and numbers for a pleading document]

**PosBlockBeg (**

[Move the insertion point to the beginning of the selected text, then deselect the text]

**PosBlockBottom ()**  
[Move the insertion point to the bottom of the selected text]

**PosBlockTop ()**  
[Move the insertion point to the top of the selected text]

**PosCellDown ()**  
[Move the insertion point down one row in a table]

**PosCellNext ()**  
[Move the insertion point one cell to the right in a table]

**PosCellPrevious ()**  
[Move the insertion point one cell to the left in a table]

**PosCellUp ()**  
[Move the insertion point up one row in a table]

**PosCharacter (CharToMoveTo:String)**  
[Move the insertion point forward to a specified character]

**PosCharNext ()**  
[Move the insertion point one character to the right]

**PosCharPrevious ()**  
[Move the insertion point one character to the left]

**PosColBottom ()**  
[Move the insertion point to the bottom of the current column]

**PosColFirst ()**  
[Move the insertion point to the beginning of the line in the first column]

**PosColLast ()**  
[Move the insertion point to the beginning of the line in the last column]

**PosColNext ()**  
[Move the insertion point to the next column]

**PosColPrevious ()**  
[Move the insertion point to the previous column]

**PosColTop ()**  
[Move the insertion point to the beginning of the current column]

**PosDocBottom ()**  
[Move the insertion point to the bottom of the document (after codes)]

**PosDocTop ()**  
[Move the insertion point to the top of the document (after codes)]

**PosDocVeryTop ()**  
[Move the insertion point ahead of all text and codes in the document]

**PosEndnote ([EndnoteNumber:Numeric])**  
[Move to an endnote]

**PosFootnote ([FootnoteNumber:Numeric])**  
[Move to a footnote]

**PosFunction (Function:Enumeration {  
HardSpace!;  
HardHyphen!;  
CancelHyphenation!;  
HardEndOfLine!;  
SoftEndOfLine!;  
Tab!})**  
[Move the insertion point to the right of the next occurrence of the specified code]

**PosGoPrevious ()**  
[Move the insertion point to its previous position]

**PosHeading ([HeadingNumber:Numeric])**  
[Obsolete Command]

**PosIndex ()**  
[Move to the index]

**PosLine ([LineNumber:Numeric])**  
[Move to a line number]

**PosLineBeg ()**  
[Move the insertion point to the beginning of the active line]

**PosLineDown ()**  
[Move the insertion point down one line]

**PosLineEnd ()**  
[Move the insertion point to the end of the active line]

**PosLineUp ()**  
[Move the insertion point up one line]

**PosLineVeryBeg ()**  
[Move the insertion point ahead of all text and codes on the active line]

**PosLineVeryEnd ()**  
[Move the insertion point after all text and codes on the active line]

**PosNextComment ()**  
[Move to the next comment]

**PosNextEndnote ()**  
[Move to the next endnote]

**PosNextFootnote ()**  
[Move to the next footnote]

**PosNextHeading ()**  
[Move to the next heading]

**PosNextIndex ()**  
[Move to the next index]

**PosNextTOA ()**  
[Move to the next T.O.A.]

**PosNextTOC ()**  
[Move to the next T.O.C.]

**PosPrevComment ()**  
[Move to the previous comment]

**PosPrevEndnote ()**  
[Move to the previous endnote]

**PosPrevFootnote ()**  
[Move to the previous footnote]

**PosPrevHeading ()**  
[Move to the previous heading]

**PosPrevIndex ()**  
[Move to the previous index]

**PosPrevTOA ()**  
[Move to the previous T.O.A.]

**PosPrevTOC ()**  
[Move to the previous T.O.C.]

**PosPage (**  
Volume:Enumeration {NoneSpecified!};  
Chapter:Enumeration {NoneSpecified!};  
Page:Enumeration {NoneSpecified!};

**[Direction:Enumeration {Down!; Up!}]**  
[Move the insertion point to a specified page]

**PosPageBottom ()**  
[Move the insertion point to the end of the page]

**PosPageNext ()**  
[Move the insertion point to the top of the next page]

**PosPagePrevious ()**  
[Move the insertion point to the top of the previous page]

**PosPageTop ()**  
[Move the insertion point to the top of the current page]

**PosParagraphNext ()**  
[Move the insertion point to the beginning of the next paragraph]

**PosParagraphPrevious ()**  
[Move insertion point to the start of the paragraph, then up a paragraph]

**PosScreenDown ()**  
[Move the insertion point to the bottom of the screen, then down a screen]

**PosScreenLeft ()**  
[Move the insertion point one screen to the left]

**PosScreenRight ()**  
[Move the insertion point one screen to the right]

**PosScreenUp ()**  
[Move the insertion point to the top of the screen, then up a screen]

**PosSelectBottom ()**  
[Move the insertion point to the end of the selected text]

**PosSelectTop ()**  
[Move the insertion point to the beginning of the selected text]

**PosTableBegin ()**  
[Move the insertion point to the first cell in a table]

**PosTableBlockBeg ()**  
[Move the insertion point to the first cell of a selection in a table and deselect all other selected cells]

**PosTableCell (Cell:String)**  
[Move the insertion point to a specified cell]

**PosTableCellBottom ()**  
[Move the insertion point to the beginning of the last line in a cell]

**PosTableCellDown ()**  
[Move the insertion point down one cell in a table]

**PosTableCellNext ()**  
[Move the insertion point to the next cell]

**PosTableCellPrevious ()**  
[Move the insertion point to the previous cell]

**PosTableCellTop ()**  
[Move the insertion point to the beginning of the cell]

**PosTableCellUp ()**  
[Move the insertion point up one cell]

**PosTableColBottom ()**  
[Move the insertion point to the end of the table column]

**PosTableColumnBottom ()**  
[Move insertion point to bottom cell in current column of table]

**PosTableColumnTop ()**  
[Move insertion point to top cell in current column of table]

**PosTableEnd ()**  
[Move the insertion point to the last cell in a table]

**PosTableGoPrevious ()**  
[Return the insertion point to its previous position in a table]

**PosTableRowBegin ()**  
[Move the insertion point to the beginning of the row in a table]

**PosTableRowEnd ()**  
[Move the insertion point to the end of the row in a table]

**PosTableScreenDown ()**  
[Move the insertion point to the bottom edge of the screen in the current table]

**PosTableScreenLeft ()**  
[Move the insertion point to the left edge of the screen in a table]

**PosTableScreenRight ()**  
[Move the insertion point to the right edge of the screen in a table]

**PosTableScreenUp ()**  
[Move the insertion point to the top of the screen in a table]

**PosTableTopCellNext ()**  
[Move the insertion point to the top of the next cell]

**PosTableTopCellPrevious ()**  
[Move the insertion point to the top of the previous cell]

**PosTOA ()**  
[Move to the T.O.A.]

**PosTOC ()**  
[Move to the T.O.C.]

**PosWordNext ()**  
[Move the insertion point to the next word in the document]

**PosWordPrevious ()**  
[Move the insertion point to the previous word in the document]

**PowerBarEditDlg ()**  
[Display the Property Bar Editor dialog box]

**PowerBarShow ([State:Enumeration {Off!;On!}])**  
[Show the Property Bar]

**PowerTableNumericFormat ()**  
[Change the number format of a cell]

**PowerTableSpeedFormat ()**  
[Reset table properties to the speedformat defaults]

**PrefApplicationBarEdit ([State:Enumeration {Off!;On!}])**  
[Add or remove items, and change the appearance of the application bar]

**PrefBackupPath (**  
    **[BackupDir:String];**  
    **[UpdateQuickList:Enumeration {No!; Yes!}])**  
[Specify the document backup folder and whether to update the QuickList]

**PrefBeep (**  
    **[Error:Enumeration {Off!; On!}];**  
    **[Hyphenation:Enumeration {Off!; On!}];**  
    **[SearchFailure:Enumeration {Off!; On!}])**  
[Specify when to turn Beep on or off]

**PrefButtonBar ()**

[Create, select, and edit toolbars]

**PrefDisplayDlg ()**

[Specify how to display documents and WordPerfect features]

**PrefDisplaySet ()**

```
[DefaultView:Enumeration {
  ViewDraft!;
  ViewPage!;
  ViewTwoPage!};]
[ShowTableGridLines:Enumeration {No!; Yes!};]
[ShowComments:Enumeration {No!; Yes!};]
[ShowGraphics:Enumeration {No!; Yes!};]
[TextInSystemColors:Enumeration {No!; Yes!};]
[ShowHiddenText:Enumeration {No!; Yes!};]
[SculpturedDialogs:Enumeration {No!; Yes!};]
[VerticalScrollBar:Enumeration {No!; Yes!};]
[HorizontalScrollBar:Enumeration {
  Hide!;
  Show!;
  WhenRequired!};]
[ShowHotSpots:Enumeration {No!; Yes!};]
[ShowShadowCursor:Enumeration {No!; Yes!};]
[ShowTabBarIcons:Enumeration {No!; Yes!};]
[ShowMarginIcons:Enumeration {No!; Yes!};]
[ShowRealWYSBYGI:Enumeration {No!; Yes!};]
[ShowTableFormulaIndicator:Enumeration {No!; Yes!};]
[BrowseBy:Enumeration {
  BottomOnly!;
  TopOnly!;
  BottomTop!};]
```

[Specify how to display features]

**PrefDocSummary ()**

```
[SubjectSearchText:String];
[DefaultDescriptiveType:String];
[CreateOnExit:Enumeration {No!; Yes!};]
[UseDescrOnOpen:Enumeration {No!; Yes!};]
[MakeFilenameDescrOnSaveAs:Enumeration {No!; Yes!};]
```

[Set document summary settings]

**PrefDocSummaryDlg ()**

[Specify document summary defaults]

**PrefEnvironment ()**

```
[UserName:String];
[UserInitials:String];
[UserColorRed:Numeric];
[UserColorGreen:Numeric];
[UserColorBlue:Numeric];
[ConfirmCodeDeletion:Enumeration {No!; Yes!};]
[ConfirmTableFormulaDeletion:Enumeration {No!; Yes!};]
[SetQuickMarkOnSave:Enumeration {No!; Yes!};]
[ActivateHypertext:Enumeration {Off!; On!};]
[FormatForDefaultPrinter:Enumeration {No!; Yes!};]
[SaveWorkspace:Enumeration {
  Always!;
  Never!;
  PromptOnExit!};]
[AutoWordSelect:Enumeration {Off!; On!};]
[GoToQuickMarkOnOpen:Enumeration {No!; Yes!};]
[Use51LetterStyle:Enumeration {No!; Yes!};]
[WP9Selection:Enumeration {Off!; On!};]
```

[Specify environment settings]

**PrefHyphenation ([Prompt:Enumeration {**

```
Always!;
Never!;
WhenRequired!};])
```

[Turn on Hyphenation and specify prompt conditions]

**PrefImportOptions (**

```
[FieldDelimiters:String];
[RecordDelimiters:String];
[StripCharacter:String];
[EncapsulationCharacter:String])
```

[Specify how certain characters are interpreted during document imports]

**PrefLanguageSet (LanguageCode:String)**

[Specify a language]

**PrefLiteCodes (**

```
[Show:Enumeration {No!; Yes!};]
[Space:Enumeration {No!; Yes!};]
[HardReturn:Enumeration {No!; Yes!};]
[Tab:Enumeration {No!; Yes!};]
[Indent:Enumeration {No!; Yes!};]
[SoftHyphen:Enumeration {No!; Yes!};]
[Advance:Enumeration {No!; Yes!};]
[Center:Enumeration {No!; Yes!};]
[CenterPage:Enumeration {No!; Yes!};]
[FlushRight:Enumeration {No!; Yes!};]
```

[Display document symbols in new and active document windows]

**PrefLocationDlg ()**

[Specify default folders for documents, templates, labels, etc.]

**PrefLocationOfDocuments (**

```
[Path:String];
[UseDefExt:Enumeration {No!; Yes!};]
[DefExt:String];
[UpdateQuickList:Enumeration {No!; Yes!};]
[DefFormat:String];
[UseExplorerStyleFileDialogs:Enumeration {No!; Yes!};]
[UseOrigFileFormat:Enumeration {No!; Yes!};]
[UseODMAIntegration:Enumeration {No!; Yes!};]
```

[Specify the location of document and backup files]

**PrefLocationOfGraphicsFiles (**

```
[GraphicsDir:String];
[GraphicsSupplDir:String];
[UpdateQuickList:Enumeration {No!; Yes!};]
[GraphicsFancyBordersDir:String])
```

[Specify the location of graphics files]

**PrefLocationOfMacroFiles (**

```
[MacroDir:String];
[MacroSupplDir:String];
[UpdateQuickList:Enumeration {No!; Yes!};]
```

[Specify the location of macro files]

**PrefLocationOfPrinterFiles (**

```
[PrinterDir:String];
[PrinterSupplDir:String];
[LabelFile:String];
[LabelsDisplay:Enumeration {
  DisplayLaser!;
  DisplayTractorFed!;
  DisplayBoth!};]
[UpdateQuickList:Enumeration {No!; Yes!};]
```

[Specify the location of label files and set related options]

**PrefLocationOfSpreadsheetFiles (**

```
[SpreadsheetDir:String];
[SpreadsheetSupplDir:String];
[DatabaseDir:String];
[DatabaseSupplDir:String];
[UpdateQuickList:Enumeration {No!; Yes!};]
```

[Specify the location of spreadsheet and database files]

**PrefLocationOfTemplates (**

```
[TemplateDir:String];
[TemplateSupplPath:String];
```

[File:String];  
[SupplFile:String];  
[DefExt:String];  
[UpdateQuickList:Enumeration {No!; Yes!}];  
[UpdateDefaultTemplate:Enumeration {No!; Yes!}];  
[Specify the location of template files]

**PrefMenu** (  
[DisplayShortcuts:Enumeration {No!; Yes!}];  
[DisplayLastOpened:Enumeration {No!; Yes!}];  
[DisplayHelpPrompts:Enumeration {No!; Yes!}];  
[Display Quick Tips, shortcut keys, and the last four most recently opened documents])

**PrefMergeCodes** (Show:Enumeration {  
Hide!;  
Show!;  
ShowAsMarkers!})  
[Display or hide merge codes]

**PrefMergeExtensions** (  
[UseDefaultFormExt:Enumeration {No!; Yes!}];  
[FormFileExt:String];  
[UseDefaultDataExt:Enumeration {No!; Yes!}];  
[DataFileExt:String])  
[Specify merge form and data file extension options]

**PrefMetafileOptions** (State:Enumeration {  
ConvertToWPG!;  
RetainWindowsAndWPG!;  
RetainWindowsFormat!})  
[Specify how to import graphics files]

**PrefPowerBar** (  
[Font:String];  
[Size:Numeric];  
[Style:Enumeration {  
Picture!;  
Text!;  
PictureAndText!}];  
[ShowQuickTips:Enumeration {No!; Yes!}])  
[Specify Property Bar display options]

**PrefPowerBarEdit** ([Action:Enumeration {Exit!; Enter!}])  
[Open and close the Property Bar Options dialog box]

**PrefPrintEnvironment** (FormatForDefaultPrinter:Enumeration {No!; Yes!})  
[Format documents for the active printer]

**PrefPrintRedline** (  
[Method:Enumeration {  
PrinterDependent!;  
Left!;  
Alternating!;  
Right!}];  
[Character:String])  
[Specify the default redline method and character]

**PrefPrintSizeRatio** (  
[Fine:Numeric];  
[Small:Numeric];  
[Large:Numeric];  
[VeryLarge:Numeric];  
[ExtraLarge:Numeric];  
[SuperSubscript:Numeric])  
[Specify a size for each relative font size attribute as a percent of the selected font size]

**PrefRevealCodes** (  
[FontFace:String];  
[FontSize:Numeric];  
[WindowSize:Numeric];

[ShowOnNew:Enumeration {No!; Yes!}];  
[ShowSpacesAsBullets:Enumeration {No!; Yes!}];  
[WrapAtWindow:Enumeration {No!; Yes!}];  
[ShowInDetail:Enumeration {No!; Yes!}];  
[Sculptured:Enumeration {No!; Yes!}];  
[ForeColorRed:Numeric];  
[ForeColorGreen:Numeric];  
[ForeColorBlue:Numeric];  
[BackColorRed:Numeric];  
[BackColorGreen:Numeric];  
[BackColorBlue:Numeric];  
[ShowHelpPrompts:Enumeration {No!; Yes!}];  
[UseSystemColors:Enumeration {No!; Yes!}])  
[Specify Reveal Code settings]

**PrefRevealCodesDlg** ()  
[Specify how to display codes in Reveal Codes]

**PrefRuler** (  
[RulerOnNew:Enumeration {No!; Yes!}];  
[TabsToGrid:Enumeration {No!; Yes!}];  
[ShowRulerGuides:Enumeration {No!; Yes!}])  
[Specify default Ruler settings]

**PrefSave** ()  
[Save preference command changes]

**PrefStatusBarEdit** ([State:Enumeration {Off!; On!}])  
[Add or remove items, and change the appearance of the application bar]

**PrefStatusBarOptions** (  
[FontFace:String];  
[FontSize:Numeric];  
[Bold:Enumeration {No!; Yes!}];  
[BoxStyle:Enumeration {  
Flat!;  
Inset!;  
Raised!}];  
[EvenSpaced:Enumeration {No!; Yes!}])  
[Specify the application bar settings]

**PrefStatusBarOptionsDlg** ()  
[Specify the font and appearance of application bar items]

**PrefTaskBar** (  
[Box:Enumeration {  
Font!;  
Select!;  
InsertMode!;  
Date!;  
GeneralStatus!;  
AlignChar!;  
Zoom!;  
CapsLockState!;  
NumLockState!;  
ScrollLockState!;  
Printer!;  
Keyboard!;  
CombinedPosition!;  
Time!;  
CurrentTag!;  
LGCName!;  
CombinedAbsPosition!}];  
[TextOrIcon:Numeric];  
[Width:Numeric])  
[Specify the desired application bar items, their appearance and width]

**PrefTOA** (  
[Underlining:Enumeration {  
NotAllowed!;  
Allowed!}];

**[Runs:Enumeration {  
NotAllowed!;  
Allowed!}]**  
[Specify a table of authorities format]

**PrefUnitsOfMeasure (**  
**[DisplayUnits:Enumeration {**  
**InchesQuote!;**  
**Inches!;**  
**Centimeters!;**  
**Millimeter!;**  
**Point!;**  
**Wpu!}]**;  
**[StatusBarUnits:Enumeration {**  
**InchesQuote!;**  
**Inches!;**  
**Centimeters!;**  
**Millimeter!;**  
**Point!;**  
**Wpu!}]**)  
[Specify unit of measure settings]

**PrefZoom (**  
**Type:Enumeration {**  
**FixedPercent!;**  
**MarginWidth!;**  
**PageWidth!;**  
**FullPage!};**  
**[Percent:Numeric]**)  
[Specify the document window view]

**Print ([Action:Enumeration {**  
**FullDocument!;**  
**CurrentPage!;**  
**MultiplePages!;**  
**AdvancedMultiplePages!;**  
**SelectedText!;**  
**DocumentSummary!;**  
**DocumentOnDisk!}]**)  
[Print according to specified options]

**Numeric := PrintAction ([Action:Enumeration {**  
**FullDocument!;**  
**CurrentPage!;**  
**MultiplePages!;**  
**AdvancedMultiplePages!;**  
**SelectedText!;**  
**DocumentSummary!;**  
**DocumentOnDisk!}]**)  
[Specify a print action or return the current print action]

**PrintAdvancedRangeData (**  
**Pages:String;**  
**SecondaryPages:String;**  
**Chapters:String;**  
**Volumes:String)**  
[Specify pages to print]

**Numeric := PrintBooklet ([State:Enumeration {False!; True!}])**  
[Turn booklet printing on or off]

**Numeric := PrintCopies ([NumberOfCopies:Numeric])**  
[Specify the number of copies to print]

**Any := PrintDestination ([Destination:Any {**  
**DriverPort!;**  
**PromptForFilename!}]**)  
[Specify to print to a printer, file, or printer port]

**PrintDlg ()**  
[Specify how you want to print the document]

**PrintEnvelope ()**  
[Print Current Envelope]

**PrinterCommand (**  
**Flag:Enumeration {**  
**File!;**  
**Command!};**  
**Command:String)**  
[Insert a [Ptr Cmnd] at the insertion point]

**PrinterCommandDlg ()**  
[Insert printer commands into the active document]

**PrinterPause ()**  
[Pause the active print job]

**String := PrinterSelectByName ([PrinterName:Any {Default!}])**  
[Specify a printer to print to]

**PrinterSelectDlg ()**  
[Select a printer and change printer options]

**Boolean := PrintGraphically ([State:Enumeration {False!;  
True!}])**  
[Turn the Print graphically option on or off]

**Boolean := PrintGraphics ([State:Enumeration {False!; True!}])**  
[Turn graphics printing on or off]

**PrintHistoryDlg ()**  
[Cancel a print job, view print job status, or review print job history]

**Numeric := PrintImageBinding ([Edge:Enumeration {**  
**LeftEdge!;**  
**RightEdge!;**  
**TopEdge!;**  
**BottomEdge!;**  
**InsideEdge!;**  
**OutsideEdge!}]**)  
[Specify the binding edge]

**Numeric Measurement := PrintImageOffset ([Offset:Numeric  
Measurement])**  
[Specify the binding edge offset]

**Boolean := PrintInColor ([State:Enumeration {False!; True!}])**  
[Turn color printing on or off]

**Boolean := PrintInReverseOrder ([State:Enumeration {False!;  
True!}])**  
[Turn reverse order printing on or off]

**Numeric := PrintRangeFrom ([Page:Numeric])**  
[Specify the first page in the print range]

**Numeric := PrintRangeTo ([Page:Numeric])**  
[Specify the last page in the print range]

**Numeric := PrintResolution ([Quality:Enumeration {**  
**Low!;**  
**Medium!;**  
**High!}]**)  
[Specify the print resolution]

**Numeric := PrintResolutionDpiX ([Dpi:Numeric])**  
[Specify the horizontal dots per inch (DPI) for a print job]

**Numeric := PrintResolutionDpiY ([Dpi:Numeric])**  
[Specify the vertical dots per inch (DPI) for a print job]

**PrintSettings (**  
**Action:Enumeration {**  
**Add!;**

**Delete!**  
**Update!**  
**Use!;**  
**[SettingName:Any {**  
**Default!**  
**DriverSettings!;}]**  
[Add, delete, update, and use printer named settings]

**Numeric := PrintSortOrder ([Order:Enumeration {**  
**Group!**  
**Collate!;}]**)  
[Specify how to sort the pages of multiple-copy print jobs]

**String := PrintSourceFileName ([Filename:String])**  
[Specify a document on disk to print]

**Boolean := PrintSuppressedBoxes ([State:Enumeration {No!**  
**Yes!;}]**)  
[Specify if suppressed boxes will print]

**PrintTo (**  
**Filename:String;**  
**PrinterName:String;**  
**[PrinterDevice:String];**  
**[PrinterPort:String])**  
[Print full document to a specified printer]

**Numeric := PrintTwoSided ([Option:Enumeration {**  
**None!**  
**LongEdge!**  
**ShortEdge!**  
**OddOnly!**  
**EvenOnly!;}]**)  
[Turn two-sided printing on or off and specify options]

**PrintTwoSidedDocCodesDlg ()**  
[Insert a Two-Sided Document code]

**Boolean := PrintUseTwoSidedDocCodes ([State:Enumeration**  
**{False!; True!;}]**)  
[Turn Use two-sided document settings on or off]

**Boolean := PrintWithDocumentSummary ([State:Enumeration**  
**{False!; True!;}]**)  
[Turn document summary printing on or off]

**ProofReadAsYouGoOff ()**  
[Turn off Spell-As-You-Go and Grammar-As-You-Go]

**ProofReadGrammar ()**  
[Mark and correct grammar errors as you type]

**PublishToFaxDlg ([FileName:String])**  
[Fax the current document]

**PublishToTrellixDlg ()**  
[No note was included in the dump but is an obsolete command]

**PublishToTrellix (**  
**Filename:String;**  
**[PublistToHtmlViaTrellix:Enumeration {No!; Yes!;}]**;  
**[Appearance:Enumeration {**  
**PreserveExactAppearance!**  
**ApplyMatchingTrellixStyles!;}]**;  
**[ImportMethod:Enumeration {**  
**SingleTrellixPage!**  
**BreakByPages!**  
**BreakByStyle!**  
**BreakByFontSize!;}]**;  
**[BreakByStyleOrFontOption:Any])**  
[Obsolete Command]

**QuattroPro ()**  
[Start Quattro Pro 9 within WordPerfect 10]

**QuickCorrect (State:Enumeration {Off!;On!;})**  
[Turn QuickCorrect on or off]

**QuickCorrectAddItem (**  
**Item:String;**  
**Replacement:String;**  
**Type:Enumeration {**  
**QuickCorrect!**  
**QCEXception!**  
**QuickLink!;})**  
[Add a QuickCorrect entry]

**QuickCorrectAutoCreateException (State:Enumeration**  
**{Off!;On!;})**  
[Turn the QuickCorrect Auto Exception List on or off]

**QuickCorrectCapsFixSet (State:Enumeration {Off!;On!;})**  
[Turn the QuickCorrect CapsFix option on or off]

**QuickCorrectDeleteItem (Item:String)**  
[Delete a QuickCorrect entry]

**QuickCorrectDlg ()**  
[Automatically correct typing errors, begin bulleted or numbered lists, etc.]

**QuickCorrectOtherWords (State:Enumeration {Off!;On!;})**  
[Turn QuickCorrect other mis-typed words when possible on or off]

**QuickCorrectQuickBulletsSet (State:Enumeration {Off!;On!;})**  
[Turn the QuickCorrect QuickBullets option on or off]

**QuickCorrectQuickIndentSet (State:Enumeration {Off!;On!;})**  
[Turn the QuickIndent option on or off]

**QuickCorrectQuickLinesSet (State:Enumeration {Off!;On!;})**  
[Replace character lines with graphic lines]

**QuickCorrectQuickOrdinalsSet (State:Enumeration {Off!;On!;})**  
[Superscript ordinal text]

**QuickCorrectQuickSymbolsSet (State:Enumeration {Off!;On!;})**  
[Turn the QuickSymbols option on or off]

**QuickCorrectReplaceWord (String:String)**  
[Replace the current word in the document with specified text]

**QuickFontEx (**  
**FontFace:String;**  
**FontSize:Numeric Measurement;**  
**[Red:Numeric];**  
**[Green:Numeric];**  
**[Blue:Numeric];**  
**[FontType:Enumeration {**  
**Speedo!**  
**Intellifont!**  
**Type1!**  
**TrueType!;}]**;  
**[{Attributes:Enumeration {**  
**Outline!**  
**Italics!**  
**Shadow!**  
**Redline!**  
**DoubleUnderline!**  
**Bold!**  
**Strikeout!**  
**Underline!**  
**SmallCaps!;**

**Every!}}})**  
[Insert a QuickFont into the document]

**QuickFormat (Format:Enumeration {**  
**Off;**  
**Attributes!**  
**Styles!**  
**StylesAndAttributes!**  
**TableCells!**  
**TableStruct!**  
**Graphics!**  
**CharacterStyle!})**

[Format text based on selected fonts and attributes, or paragraph styles, or both]

**QuickFormatDiscontinue (AllParagraphs:Enumeration {False!**  
**True!})**  
[Turn off the QuickFormat link for all paragraphs with the associated heading]

**QuickFormatDlg ()**  
[Display the QuickFormat dialog box]

**QuickFormatToggle ()**  
[Pickup and apply formatting]

**QuickLink (State:Enumeration {Off!;On!})**  
[Turn QuickLink on or off]

**QuickmarkFind ()**  
[Jump to the QuickMark]

**QuickmarkSet ()**  
[Set the QuickMark at the insertion point]

**QuickWords (State:Enumeration {Off!;On!})**  
[Turn QuickWords on or off]

**QuickWordsDlg ()**  
[Create and insert QuickWords]

- R -

**Redisplay ()**  
[Rewrite the display]

**RedlineCharacter (Character:String)**  
[Specify a character to redline text]

**RedlineMethod (RedlineMethod:Enumeration {**  
**PrinterDependent!**  
**LeftMargin!**  
**Alternating!**  
**RightMargin!})**  
[Specify a redline method]

**Redo ([Number:Numeric])**  
[Reverse the last Undo - ^0]

**RepeatDlg ()**  
[Set the number of times to repeat the next action]

**RepeatSet (Value:Numeric)**  
[Specify the Repeat default number]

**RepeatValue ([Value:Numeric])**  
[Repeat the next action the default number of times]

**ReplaceAll ([SearchMode:Enumeration {Regular!; Extended!})**  
[Replace all occurrences of a matched word, code, or phrase]

**ReplaceBackward ([SearchMode:Enumeration {Regular!**  
**Extended!})**  
[Replace specified text or codes from the insertion point to the beginning of the document]

**ReplaceConfirm (State:Enumeration {No!;Yes!})**  
[Specify whether Find and Replace Text prompts the user to confirm replacement]

**ReplaceCurrent ()**  
[Replace a matched word, code, or phrase]

**ReplaceDlg ()**  
[Search for words or codes and replace them with new words or codes]

**ReplaceForward ([SearchMode:Enumeration {Regular!**  
**Extended!})**  
[Replace specified text or codes from the insertion point to the end of the document]

**ReplaceString (RplcStrg:String)**  
[Specify search text or codes]

**ReplaceWithAttributes (**  
**State:Enumeration {No!; Yes!;**  
**[Attribute:Enumeration {**  
**ExtraLarge!**  
**VeryLarge!**  
**Large!**  
**Small!**  
**Fine!**  
**Superscript!**  
**Subscript!**  
**Outline!**  
**Italics!**  
**Shadow!**  
**Redline!**  
**DoubleUnderline!**  
**Bold!**  
**Strikeout!**  
**Underline!**  
**SmallCaps!**  
**Every!})**

[Specify font attributes and relative font size for replacement text]

**ReplaceWithCase (State:Enumeration {No!;Yes!})**  
[Specify the case(s) of replacement text]

**ReplaceWithFont (**  
**State:Enumeration {No!; Yes!;**  
**[Name:String;**  
**[Family:Enumeration {**  
**FamilyTCBenguiat!**  
**FamilyTimesRoman!**  
**FamilyTCGaramond!**  
**FamilyCenturySchoolbook!**  
**FamilyBodoni!**  
**FamilyTCLubalinGraph!**  
**FamilyAlbertus!**  
**FamilyCourier!**  
**FamilyHelvetica!**  
**FamilyTCRonda!**  
**FamilyMicrostyle!**  
**FamilyOptima!**  
**FamilyLetterGothic!**  
**FamilyBrush!**  
**FamilyParkAvenue!**  
**FamilySignetRoundhand!**  
**FamilyTekton!**  
**FamilyUncial!**  
**FamilyOldEnglish!**  
**FamilyTCZapfDingbats!**



**FamilyBroadway!;**  
**FamilyComputer!;**  
**FamilyPTBarnum!;**  
**FamilyPonderosa!;**  
**FamilyMadrone!;**  
**FamilyHobo!;**  
**FamilyRevue!;**  
**FamilyAestheticOrnamented!}}**;  
**[Attributes:Enumeration {**  
**FontMatchNormal!;**  
**FontMatchItalic!;**  
**FontMatchOutline!;**  
**FontMatchShadow!;**  
**FontMatchSmallCaps!}}];**  
**[Weight:Enumeration {**  
**WeightUnknown!;**  
**WeightUltraThin!;**  
**WeightUltraLight!;**  
**WeightExtraThin!;**  
**WeightThin!;**  
**WeightExtraLight!;**  
**WeightLight!;**  
**WeightDemiLight!;**  
**WeightSemiLight!;**  
**WeightBook!;**  
**WeightRegular!;**  
**WeightRoman!;**  
**WeightMedium!;**  
**WeightDemiBold!;**  
**WeightSemiBold!;**  
**WeightBold!;**  
**WeightExtraBold!;**  
**WeightHeavy!;**  
**WeightExtraHeavy!;**  
**WeightBlack!;**  
**WeightExtraBlack!;**  
**WeightUltraHeavy!;**  
**WeightUltraBlack!;**  
**WeightMaximum!}}];**  
**[Width:Enumeration {**  
**WidthUnknown!;**  
**WidthUltraCondensed!;**  
**WidthUltraCompressed!;**  
**WidthCompressed!;**  
**WidthExtraCondensed!;**  
**WidthCondensed!;**  
**WidthSemiCondensed!;**  
**WidthNormal!;**  
**WidthSemiExpanded!;**  
**WidthExpanded!;**  
**WidthExtraExpanded!;**  
**WidthUltraExpanded!;**  
**WidthDoubleWide!;**  
**WidthTripleWide!}}];**  
**[Source:Enumeration {**  
**PRSFile!;**  
**DRSFile!}}];**  
**[Type:Enumeration {**  
**Speedo!;**  
**Intellifont!;**  
**Type1!;**  
**TrueType!}}];**  
**[CharacterSet:Enumeration {**  
**FontMatchASCII!;**  
**FontMatchMultinational1!;**  
**FontMatchMultinational2!;**  
**FontMatchBoxDrawing!;**  
**FontMatchTypographicSymbols!;**  
**FontMatchIconicSymbols!;**  
**FontMatchMath!;**  
**FontMatchMathExtension!;**  
**FontMatchGreek!;**

**FontMatchHebrew!;**  
**FontMatchCyrillic!;**  
**FontMatchJapaneseKana!;**  
**FontMatchUserDefined!;**  
**FontMatchArabic!;**  
**FontMatchArabicScript!}}];**  
 [Specify the font and font style of replacement text]

**ReplaceWithFontSize (**  
**State:Enumeration {No!; Yes!};**  
**[FontSize:Numeric Measurement])**  
 [Specify font size of replacement text]

**ReselectLastBlock ( )**  
 [Select the last selected text]

**ReselectLastSelection ( )**  
 [Select the last selected text]

**ResetKeyboardMenu ( )**  
 [Reset the keyboard and menus to WordPerfect 10 defaults]

**RevealCodes ([State:Enumeration {Off!;On!}])**  
 [Display all codes used in the document]

**ReverseShippingMacro ( )**  
 [Select the text and background colors, and the fill style]

**ReviewDocAnnotationAccept ( )**  
 [Incorporate the active annotation into the document]

**ReviewDocAnnotationAcceptAll ( )**  
 [Incorporate all annotations into the active document]

**ReviewDocAnnotationFirst ( )**  
 [Select the first annotation in the active document]

**ReviewDocAnnotationNext ( )**  
 [Select the next annotation in the active document]

**ReviewDocAnnotationPrev ( )**  
 [Select the previous annotation in the active document]

**ReviewDocAnnotationReject ( )**  
 [Delete the selected annotation]

**ReviewDocAnnotationRejectAll ( )**  
 [Delete all annotations in the active document]

**ReviewDocAuthorDlg ( )**  
 [Display the Review Document Author Mode dialog box and activate Author mode]

**ReviewDocDisplayModeToggle ( )**  
 [Turn annotation display on or off]

**ReviewDocMarginMarksToggle ( )**  
 [Turn annotation margin marks on or off]

**ReviewDocDlg ( )**  
 [Annotate the current document or view other reviewers' comments]

**ReviewDocManualModeToggle ([Mode:Enumeration {**  
**Off!;**  
**Reviewer!;**  
**Author!}])**  
 [Enter or exit the Author or Reviewer annotation modes]

**ReviewDocReviewerDlg ( )**  
 [Display the Review Document Reviewer Mode dialog box and activate Reviewer Mode]

**ReviewDocSetCurrentUserColor (****Red:Numeric;**  
**Green:Numeric;**  
**Blue:Numeric)**

[Set the annotation color for the current reviewer]

**RotateShapeDlg (**

[Rotate a shape object]

**RulerBarShow ([State:Enumeration {Off!;On!}])**

[Show the Ruler]

- S -**SaveAll (**

[Save all documents]

**SaveAllShippingMacro (**

[Select the open documents to save]

**SaveToDriveAMacro (**

[Save the document and copy it to drive A]

**SaveUndoInformation (State:Enumeration {No!;Yes!})**

[Save Undo and Redo information with the current document]

**ScrollScreenLeft (**

[View the left part of the document window]

**ScrollScreenRight (**

[View the right part of the document window]

**SearchCaseSensitive (State:Enumeration {No!;Yes!})**

[Make Find case-sensitive]

**SearchFindWholeWordsOnly (State:Enumeration {No!;Yes!})**

[Specify whether Find matches only whole words or also matches partial words]

**SearchInSelection (State:Enumeration {No!;Yes!})**

[Restrict the search to the selected text]

**SearchNext ([SearchMode:Enumeration {Regular!;Extended!}])**

[Find the next occurrence of the specified text or codes]

**SearchNextCurrentWord ([SearchMode:Enumeration {Regular!; Extended!}])**

[Find the next occurrence of the current word or selection]

**SearchPrevious ([SearchMode:Enumeration {Regular!; Extended!}])**

[Find the previous occurrence of the specified text or codes]

**SearchPreviousCurrentWord ([SearchMode:Enumeration {Regular!; Extended!}])**

[Find the previous occurrence of the current word or selection]

**SearchString (StrgToLookFor:String)**

[Specify a search string]

**SearchWordForms (State:Enumeration {No!;Yes!})**

[Find and/or replace word forms]

**SearchWrap (State:Enumeration {No!;Yes!})**

[Specify whether Find searches from the beginning of a document after it reaches the end of a document]

**SecondaryPageNumber (Page:Numeric)**

[Specify a secondary page number and insert the code into a document]

**SecondaryPageNumberDecrement (**

[Decrease the secondary page number by one]

**SecondaryPageNumberDisplay (**

[Insert the current secondary page number code at the insertion point]

**SecondaryPageNumberIncrement (**

[Increase the secondary page number by one]

**SecondaryPageNumberMethod (NumberingMethod:Enumeration {****Numbers!;**  
**LowerLetters!;**  
**UpperLetters!;**  
**LowerRoman!;**  
**UpperRoman!})**

[Specify a numbering method for secondary page numbers]

**SelectAll (**

[Select everything in the document]

**SelectCell (**

[Select the current cell in a table]

**SelectCellDown (**

[Select text from the insertion point to the cell below]

**SelectCellDownArrow (**

[Select text from the insertion point to the next row]

**SelectCellLeft (**

[Select text from the insertion point to the cell in the previous column]

**SelectCellRight (**

[Select text from the insertion point to the cell in the next column]

**SelectCellUp (**

[Select text from the insertion point to the cell above]

**SelectCellUpArrow (**

[Select the text from the insertion point to the previous row]

**SelectCharNext (**

[Select the text from the insertion point to the next character]

**SelectCharPrevious (**

[Select the text from the insertion point to the previous character]

**SelectColumnBottom (**

[Select text from the insertion point to the bottom of the column]

**SelectColumnNext (**

[Select the text from the insertion point to the same line in the next column]

**SelectColumnPrevious (**

[Select the text from the insertion point to the same line in the previous column]

**SelectColumnTop (**

[Select text from the insertion point to the top of the column]

**SelectDelete (**

[Delete the selected text]

**SelectDocBottom (**

[Select text from insertion point to bottom of document]

**SelectDocTop (**

[Select text from insertion point to top of document]

**SelectDocVeryTop (**

[Select text from insertion point to top of document before codes]

**SelectedReplaceString (**

**RplcString:String;**  
**[State:Enumeration {**  
**UseAlways!}]**)

[Replace a search string with the specified text]

**SelectLineBegin (**

[Select the text from the insertion point to the beginning of the line]

**SelectLineDown (**

[Select the text from the insertion point to the next line]

**SelectLineEnd (**

[Select the text from the insertion point to the end of the line]

**SelectLineUp (**

[Select the text from the insertion point to the previous line]

**SelectLineVeryBegin (**

[Select the text from the insertion point to the beginning of the line, before the codes]

**SelectLineVeryEnd (**

[Select the text from the insertion point to the end of the line, after all the text and codes]

**SelectMode ([State:Enumeration {Off!;On!}])**

[Turn Select on or off]

**SelectOff (**

[Turn off Select]

**SelectOn (SelectMode:Enumeration {**

**CharMode!;**  
**WordMode!;**  
**SentenceMode!;**  
**ParagraphMode!;**  
**PageMode!;**  
**RectangleMode!;**  
**TabularColumnsMode!;**  
**DocMode!;**  
**CharNoCellMode!})**

[Turn on Select and specify the mode]

**SelectOnEmptyDoc (SelectMode:Enumeration {**

**CharMode!;**  
**WordMode!;**  
**SentenceMode!;**  
**ParagraphMode!;**  
**PageMode!;**  
**RectangleMode!;**  
**TabularColumnsMode!;**  
**DocMode!;**  
**CharNoCellMode!})**

[Activate Select and specify a Select mode in an empty document]

**SelectPage (**

[Select everything on the page]

**SelectPageNext (**

[Select the text from the insertion point to the next page]

**SelectPagePrevious (**

[Select the text from the insertion point to the previous page]

**SelectParagraph (**

[Select the current paragraph]

**SelectParagraphNext (**

[Select the text from the insertion point to the next paragraph]

**SelectParagraphPrevious (**

[Select the text from the insertion point to the previous paragraph]

**SelectRectangle (**

[Select a rectangular area based on the current selection]

**SelectSave (Filename:String)**

[Save the selected text as a new file]

**SelectScreenDown (**

[Select the text from the insertion point to the bottom of the screen]

**SelectScreenLeft (**

[Select the text from the insertion point to the left side of the screen]

**SelectScreenRight (**

[Select the text from the insertion point to the right side of the screen]

**SelectScreenUp (**

[Select the text from the insertion point to the top of the screen]

**SelectSentence (**

[Select the current sentence]

**SelectSentenceNext (**

[Select the text from the insertion point to the next sentence]

**SelectSentencePrevious (**

[Select the text from the insertion point to the previous sentence]

**SelectTable (**

[Select all cells in the current table]

**SelectTableColumn (**

[Select all cells in the current column of a table]

**SelectTableColumnExtendLeft (**

[Extend the selection to the beginning of the row in a table]

**SelectTableColumnExtendRight (**

[Extend the selection to the end of the row in a table]

**SelectTableRow (**

[Select all cells in the current row of a table]

**SelectTabularColumn (**

[Select tabular columns within the current selection]

**SelectToggle (**

[Turn on/off Select]

**SelectWord (**

[Select the current word]

**SelectWordNext (**

[Select the text from the insertion point to the next word]

**SelectWordPrevious (**

[Select the text from the insertion point to the previous word]

**SetDocumentMapping (**

**MapName:String;**  
**Location:Enumeration {**  
**MapInternal!;**  
**MapExternal!};**  
**[Filename:String])**

[Specify a character map for one-byte characters]

**SetLineSpacing1 (**

[Set single line spacing]

**SetLineSpacing1\_5 (**

[Set 1.5 line spacing]

**SetLineSpacing2 (**

[Set double line spacing]

**SettingsDlg ()**

[Change WordPerfect settings]

**SetSelEx (Start:Numeric;End:Numeric)**

[Set up a selection]

**SetUndoLevels (Levels:Numeric)**

[Specify the number of Undo/Redo items to maintain in the document]

**SGMLAddToAttrList (Name:String;Value:String)**

[Add a value to an existing attribute value]

**SGMLBoxAddToAttrList (****Element:String;****Name:String;****Value:String)**

[Add a value to an existing box element attribute value]

**SGMLBoxAttribute (****Element:String;****Name:String;****Value:String)**

[Set the attribute value of a box element attribute]

**SGMLBoxDeleteAttribute (****Element:String;****Name:String)**

[Delete a box element attribute]

**SGMLBoxDeleteMarkup (Type:Enumeration {****Box!;****Content!;****Caption!})**

[Delete an element for a box]

**SGMLBoxElementDlg ()**

[Select XML Elements for the box]

**SGMLBoxMarkup (****Type:Enumeration {****Box!;****Content!;****Caption!};****Name:String)**

[Set the element for a box]

**SGMLCDATASection ()**

[Insert an XML CDATA section]

**SGMLCharRef (****Character:String;****[Type:Enumeration {****Decimal!;****Hexadecimal!})**

[Insert a character reference]

**SGMLContentModelDlg ([Element:String])**

[View an element's content model]

**SGMLCreateDocType (****DocTypeName:String;****LogicFile:String;****[LayoutFile:String];****[AliasFile:String];****[OptFile1:String];****[OptFile2:String];****[OptFile3:String];****[OptFile4:String];****[OptFile5:String];****[OptFile6:String];****[OptFile7:String];****[OptFile8:String];****[OptFile9:String];****[OptFile10:String])**

[Create Document Type Description: Logic (DTD), Layout, and Alias files]

**SGMLDeleteAttribute (Name:String)**

[Delete an attribute]

**SGMLDeleteDocType (DocTypeName:String)**

[Delete Document Type Description from the list of Document Types]

**SGMLDocTypeDlg ()**

[Create a new document based on an XML Project]

**SGMLEditAttributeDlg (Name:String)**

[Edit an attribute of the current element]

**SGMLEditLayout (****[Name:String];****[CreateGenericLayout:Enumeration {No!; Yes!}])**

[Edit Layout from XML Project Template]

**SGMLEditAttributesDlg ()**

[Edit the attributes of the current element]

**SGMLEditBoxAttributeDlg (****Element:String;****Name:String)**

[Edit an attribute of a box element]

**SGMLEditBoxAttributesDlg (Element:String)**

[Edit the attributes of a box element]

**SGMLElementDlg ()**

[Insert an XML element]

**SGMLElementList ()**

[Display valid elements]

**SGMLElementEditDlg ()**

[Edit an XML element]

**SGMLElementsDlg ()**

[Insert XML elements]

**SGMLEndElement ()**

[End the current XML Element]

**SGMLEntityRef (Name:String)**

[Insert an entity reference]

**SGMLExportImport (****Name:String;****[DocType:String])**

[Export or Import XML files]

**SGMLFileRefDataType (****FileRefName:String;****DataType:Enumeration {****None!;****CDATA!;****NDATA!;****SDATA!;****SUBDOC!})**

[Set the data type of a file reference]

**SGMLFileRefDelAttribute (****FileRefName:String;****Name:String)**

[Delete a file reference notation attribute]

**SGMLFileRefDelete (Name:String)**

[Delete a file reference declaration]

**SGMLFileRefDig ()**  
[Insert a file reference]

**SGMLFileRefFilePath (**  
  **FileRefName:String;**  
  **FilePath:String)**  
[Set the file path of a file reference]

**SGMLFileRefNew (**  
  **Name:String;**  
  [**PublicID:String;**]  
  [**FilePath:String;**]  
  [**DataType:Enumeration {**  
    **None!;**  
    **CDATA!;**  
    **NDATA!;**  
    **SDATA!;**  
    **SUBDOC!};**  
  [**NotationName:String]**]  
[Create a file reference declaration]

**SGMLFileRefNewAttribute (**  
  **FileRefName:String;**  
  **Name:String;**  
  **Value:String)**  
[Create a file reference notation attribute]

**SGMLFileRefNotation (**  
  **FileRefName:String;**  
  **Notation:String)**  
[Set the notation of a file reference]

**SGMLFileRefPubID (**  
  **FileRefName:String;**  
  **PublicID:String)**  
[Change the name of a file reference]

**SGMLFileRefRename (**  
  **OldName:String;**  
  **NewName:String)**  
[Change the name of a file reference]

**SGMLFindElementEnd ()**  
[Find the end of the current element]

**SGMLFindElementStart ([CursorPos:Enumeration {**  
  **After!;**  
  **Before!};])**  
[Find the start of the current element]

**SGMLFindNextTable ()**  
[Find the next table]

**SGMLFindStartTag (**  
  [**Name:String;**]  
  [**Direction:Enumeration {**  
    **Either!;**  
    **Next!;**  
    **Previous!};**]  
  [**AttributeEditFind:Enumeration {No!; Yes!};**]  
[Find the start of an element]

**SGMLGetAttributeData (**  
  **Variable:Variable;**  
  **Item:Enumeration {**  
    **Count!;**  
    **Name!;**  
    **Type!;**  
    **Value!;**  
    **DefaultValue!;**  
    **Status!;**  
    **ErrorType!};**  
  **Type:Enumeration {**

**All!;**  
    **InTag!;**  
    **NotInTag!;**  
    **Invalid!};**  
  **Element:Enumeration {**  
    **Open!;**  
    **Previous!};**  
  [**ItemNum:Numeric]**  
[Get information related to the attributes of an element]

**SGMLGetElementData (**  
  **Variable:Variable;**  
  **Item:Enumeration {**  
    **Count!;**  
    **Name!;**  
    **LSIType!};**  
  **Type:Enumeration {**  
    **All!;**  
    **Open!;**  
    **Valid!;**  
    **Previous!};**  
  [**ItemNum:Numeric]**  
[Get information related to elements]

**SGMLGetFileRefData (**  
  **Variable:Variable;**  
  **Item:Enumeration {**  
    **Count!;**  
    **Name!;**  
    **FilePath!;**  
    **PublicID!;**  
    **DataType!;**  
    **Notation!;**  
  **Type:Enumeration {**  
    **User!;**  
    **DTD!};**  
  [**ItemNum:Numeric]**  
[Get information related to file references]

**SGMLIDIDREFListDig ()**  
[Display the list of ID and IDREF attributes]

**SGMLIgnore ()**  
[Insert XML WP DATA codes]

**SGMLInsertElement (**  
  **Name:String;**  
  [**Options:Enumeration {**  
    **None!;**  
    **Spawn!;**  
    **Empty!};**]  
  [**Attribute:String;**]  
  [**Value:String;**]  
[Insert an element]

**SGMLLoadDocType (DocTypeName:String)**  
[Load the logic, layout, and alias files]

**SGMLLogicErrorDig ()**  
[Display XML logic errors]

**SGMLMarkedSecParamEntity (**  
  **Name:String;**  
  [**Keyword:Enumeration {**  
    **None!;**  
    **CDATA!;**  
    **RCDATA!;**  
    **IGNORE!;**  
    **INCLUDE!;**  
    **TEMP!};**]  
[Create/Replace a marked section parameter entity declaration]

**SGMLMarkedSection** (  
[Keyword:Enumeration {  
None!;  
CDATA!;  
RCDATA!;  
IGNORE!;  
INCLUDE!;  
TEMP!}];  
{[Name:String]})  
[Insert a marked section]

**SGMLMarkedSectionDig** ()  
[Insert a marked section]

**SGMLMrkSecPEDelete** (Name:String)  
[Delete a marked section parameter entity declaration]

**SGMLMrkSecPERename** (  
Name:String;  
NewName:String)  
[Rename a marked section parameter entity]

**SGMLNewAttribute** (  
Name:String;  
Value:String)  
[Create an attribute]

**SGMLPreferences** (  
[DocTypeDir:String];  
[Show:Enumeration {  
Hide!;  
Show!;  
ShowAsMarkers!}];  
[RevealCode:Enumeration {  
ShowLayout!;  
ShowAttributes!}];  
[Red:Numeric];  
[Green:Numeric];  
[Blue:Numeric];  
[FontSize:Numeric Measurement];  
[HTMLImport:Enumeration {No!; Yes!}];  
[ImportPrompt:Enumeration {No!; Yes!}];  
[OmitTagsImport:Enumeration {No!; Yes!}];  
[DefaultSaveAsXML:Enumeration {No!; Yes!}];  
[TemplateFolder:String];  
[ProjBaseTemplate:String];  
[GraphicsFolder:String];  
{[CatalogFile:String]})  
[Set display of XML codes; specify default document type folder]

**SGMLProcessingInstruction** ([Name:String])  
[Insert an XML processing instruction]

**SGMLProcessingInstructionDig** ()  
[Insert an XML processing instruction]

**SGMLProcessTarget** (  
Name:String;  
[PublicID:String];  
[FilePath:String])  
[Create a process target declaration]

**SGMLProcessTargetDelete** (Name:String)  
[Delete a process target]

**SGMLProcessTargetFilePath** (  
TargetName:String;  
FilePath:String)  
[Set the file path of a process target]

**SGMLProcessTargetPubID** (  
TargetName:String;  
PublicID:String)  
[Change the pubid id of a process target]

**SGMLProcessTargetRename** (  
OldName:String;  
NewName:String)  
[Change the name of a process target]

**SGMLRenameElement** (Name:String)  
[Rename an element without changing the attributes]

**SGMLReplaceAttribute** (  
Name:String;  
Value:String)  
[Replace an existing attribute value]

**SGMLRootElement** (Element:String)  
[Change the XML root element]

**SGMLRootElementDig** ()  
[Select new XML root element]

**SGMLSaveAsOptions** (  
DocTypeDeclaration:Enumeration {No!; Yes!};  
[SGMLDeclaration:Enumeration {No!; Yes!}];  
[WrapText:Enumeration {No!; Yes!}])  
[Set options for saving a document in XML format]

**SGMLSelectElement** ()  
[Select the open XML element at the insertion point]

**SGMLSelectLayout** ([LayoutFile:String])  
[Change the layout of the XML elements]

**SGMLSelectLayoutDig** ()  
[Select XML layouts]

**SGMLSetColors** (  
{[ColorID:Enumeration {  
ELEMBKG!;  
ELEMENTXT!;  
PIBKG!;  
PITXT!;  
MSBKG!;  
MSTXT!;  
WPBKG!;  
WPTXT!;  
ERBKG!;  
ERTXT!}];  
[Red:Numeric];  
[Green:Numeric];  
[Blue:Numeric]})  
[Set XML Colors]

**SGMLSetValidation** ({[Options:Enumeration {  
CharMap!;  
Entity!;  
EntityDecl!;  
Idldref!;  
Logic!;  
TextSeq!}])  
[Select settings for validation]

**SGMLSplitelement** ()  
[Split the current XML Element]

**SGMLTagAllTables** ()  
[Automatically tag all the tables]

**SGMLTagBox** ([Element:String])  
[Tag the selected box]

**SGMLTagCurrentTable ()**  
[Automatically tag the current table]

**SGMLTagTablesDlg ()**  
[Automatically tag tables in the document]

**SGMLTextRefDlg ()**  
[Insert a text reference]

**SGMLUnloadDocType ()**  
[Unload the logic, layout, and alias files]

**SGMLValidate ()**  
[Find the next validation error]

**SGMLValidationDlg ()**  
[Validate the XML document]

**ShadowCursorOptions (**  
  **[State:Enumeration {**  
    **Text!;**  
    **WhiteSpace!;**  
    **TextAndWhiteSpace!});**  
  **[Snap:Enumeration {**  
    **Margins!;**  
    **Tabs!;**  
    **Indents!;**  
    **Spaces!});**  
  **[Shape:Enumeration {**  
    **Caret!;**  
    **IBeam!;**  
    **BrokenCaret!;**  
    **Arrows!;**  
    **DottedCaret!;**  
    **BottomArrow!});**  
  **[ColorRed:Numeric];**  
  **[ColorGreen:Numeric];**  
  **[ColorBlue:Numeric]**  
**]**  
[Set the shape, type and color of the Shadow Cursor]

**Boolean := ShadowCursorShow ([State:Enumeration {Off!;On!}])**  
[Turn the Shadow Cursor on or off and return the current state]

**ShiftTabKey ()**  
[Insert a back tab, go to the previous table cell, or go to the previous outline level]

**ShowBoxes (State:Enumeration {Off!;On!})**  
[Display graphics boxes]

**ShowGraphics ()**  
[Show graphics]

**ShowHiddenText ()**  
[Show hidden text]

**Boolean := ShowMarginIcons ([State:Enumeration {Off!;On!}])**  
[Toggle visibility of all margin icons on or off]

**ShowSGMLTree ([State:Enumeration {Off!;On!}])**  
[Display or hide the XML tree]

**ShowSymbols ()**  
[Display or hide symbols for space, hard return, tab, etc.]

**SignatureDlg ()**  
[View information about this documents signature]

**SigningDlg ()**  
[Specify the Certificate to be used to Digitally Sign this document]

**SingleSmartQuote (**  
  **[State:Enumeration {Off!; On!}];**  
  **[OpenSingleQuote:Numeric];**  
  **[CloseSingleQuote:Numeric]**  
**)**  
[Turn Single SmartQuotes on or off and specify a quote character]

**SingleSpaceInSentence (State:Enumeration {Off!;On!})**  
[Change double spaces in a sentence to single spaces]

**SoftHyphen ()**  
[Insert a soft hyphen to break a word, if necessary, at the insertion point]

**SoftSpace ()**  
[Insert a space]

**Sort (**  
  **[InFile:String];**  
  **[OutFile:String];**  
  **[State:Enumeration {NoUndo!; Undo!}])**  
[Specify a file to sort and a file to which the changes are written]

**SortAction (SortAction:Enumeration {Sort!;SelectOnly!})**  
[Sort a file or selected records]

**SortCaseOrder (Case:Enumeration {**  
  **LowercaseFirst!;**  
  **UppercaseFirst!})**  
[Specify case priority]

**SortDlg ()**  
[Order text alphabetically or numerically]

**SortKeys (**  
  **{[Field:Numeric];**  
  **[Line:Numeric];**  
  **[Word:Numeric];**  
  **[SortType:Enumeration {**  
    **Alphanumeric!;**  
    **Numeric!;**  
    **DateSort!});**  
  **[SortOrder:Enumeration {**  
    **Ascending!;**  
    **Descending!}])**  
[Specify sort key options]

**SortSelectString (SelectString:String)**  
[Insert a character string into the Select Records text box of the Sort dialog box]

**SortType (SortType:Enumeration {**  
  **ParagraphSort!;**  
  **LineSort!;**  
  **ColumnSort!;**  
  **TableSort!;**  
  **MergeSort!})**  
[Specify a sort format]

**SortUndo (State:Enumeration {NoUndo!;Undo!})**  
[Enable or disable Undo in Sort]

**SoundClipAddInstance (Description:String)**  
[Add another reference to a sound clip already in the document]

**SoundClipCreate (**  
  **Description:String;**  
  **Filename:String;**  
  **SoundFileType:Enumeration {**  
    **OnDisk!;**  
    **InDoc!};**  
  **ClipType:Enumeration {**  
    **MIDI!;**  
    **DigitalAudio!};**  
**)**

**LeftVolume:Numeric;**  
**RightVolume:Numeric)**  
[Insert a sound clip from a file into Sound Clips in the Document dialog box]

**SoundClipDelete (Description:String)**  
[Delete a sound clip]

**SoundClipDlg ()**  
[Create and edit sound clips]

**SoundClipPlayNext (**  
  **[Description:String];**  
  **[StartPos:Numeric];**  
  **[EndPos:Numeric])**  
[Play the next sound clip in the document]

**SoundClipSetAttributes (**  
  **Description:String;**  
  **Type:Enumeration {**  
    **MIDI!**  
    **DigitalAudio!};**  
  **[LeftVolume:Numeric];**  
  **[RightVolume:Numeric])**  
[Specify sound clip attributes]

**SoundClipSetDescription (**  
  **OldDescription:String;**  
  **NewDescription:String)**  
[Assign a new description to a sound clip]

**SoundClipSetFilename (**  
  **Description:String;**  
  **Filename:String;**  
  **Location:Enumeration {OnDisk!; InDoc!})**  
[Specify a sound clip description, filename, and location]

**SpecificDeleteStringBegin ()**  
[Delete specific codes]

**SpecificReplaceStringBegin ()**  
[Specify a specific code replacement]

**SpecificSearchStringBegin ()**  
[Find a specific code]

**SpecificStringEnd ()**  
[End a specific code search]

**Boolean := SpellAsYouGo ([State:Enumeration {Off!;On!}])**  
[Mark and correct spelling and typing errors as you type]

**SpreadsheetShowLinkCode (State:Enumeration {No!;Yes!})**  
[Display or hide spreadsheet link codes]

**SpreadsheetUpdateOnRetrieve (State:Enumeration {No!;Yes!})**  
[Turn the Update on retrieve option on or off for spreadsheet/database links]

**StatusBarShow ([State:Enumeration {Off!;On!}])**  
[Show the status bar]

**StyleCodes (**  
  **[State:Enumeration {WithoutOffCodes!; WithOffCodes!}];**  
  **[Library:Enumeration {**  
    **CurrentDoc!**  
    **PersonalLibrary!**  
    **SharedLibrary!}])**  
[Edit style codes]

**StyleCopy (**  
  **Style:Any {**  
    **FootnoteNumberInDocumentStyle!**

**EndnoteNumberInDocumentStyle!**  
**FigureNumberStyle!**  
**TableBoxNumberStyle!**  
**TextBoxNumberStyle!**  
**UserBoxNumberStyle!**  
**EquationNumberStyle!**  
**HypertextStyle!**  
**ToC1Style!**  
**ToC2Style!**  
**ToC3Style!**  
**ToC4Style!**  
**ToC5Style!**  
**Index1Style!**  
**Index2Style!**  
**ListStyle!**  
**ToAStyle!**  
**DocStyle!**  
**BoxTextStyle!**  
**CaptionStyle!**  
**FootnoteStyle!**  
**EndnoteStyle!**  
**CommentStyle!**  
**HeaderAStyle!**  
**HeaderBStyle!**  
**FooterAStyle!**  
**FooterBStyle!**  
**WatermarkAStyle!**  
**WatermarkBStyle!**  
**ParagraphOutInStyle!**  
**OutlineStyle!**  
**LegalOutInStyle!**  
**BulletsOutInStyle!**  
**HeadingsOutInStyle!**  
**Legal\_2OutInStyle!**  
**NumbersOutInStyle!**  
**Level1Style!**  
**Level2Style!**  
**Level3Style!**  
**Level4Style!**  
**Level5Style!**  
**Level6Style!**  
**Level7Style!**  
**Level8Style!**  
**Legal1Style!**  
**Legal2Style!**  
**Legal3Style!**  
**Legal4Style!**  
**Legal5Style!**  
**Legal6Style!**  
**Legal7Style!**  
**Legal8Style!**  
**Heading1Style!**  
**Heading2Style!**  
**Heading3Style!**  
**Heading4Style!**  
**Heading5Style!**  
**Heading6Style!**  
**Heading7Style!**  
**Heading8Style!**  
**MonospacedStyle!**  
**CustomHTMLStyle!**  
**PreformattedStyle!**  
**AddressStyle!**  
**QuotationsOutInStyle!**  
**DefinitionsOutInStyle!**  
**Quotation1Style!**  
**Quotation2Style!**  
**Quotation3Style!**  
**Quotation4Style!**  
**Quotation5Style!**  
**Quotation6Style!**  
**Quotation7Style!**



Quotation8Style!  
 Definition1Style!  
 Definition2Style!  
 Definition3Style!  
 Definition4Style!  
 Definition5Style!  
 Definition6Style!  
 Definition7Style!  
 Definition8Style!  
 Level9Style!  
 Legal9Style!  
 Heading9Style!  
 Quotation9Style!  
 Definition9Style!);

[FromLibrary:Enumeration {  
   CurrentDoc!  
   PersonalLibrary!  
   SharedLibrary!});

[ToLibrary:Enumeration {  
   CurrentDoc!  
   PersonalLibrary!  
   SharedLibrary!});

[NewName:String])

[Copy a style between the current document, the personal library, or the shared library]

### StyleCreate (

Name:String;  
 Type:Enumeration {  
   OpenStyle!  
   CharacterStyle!  
   ParagraphStyle!  
   AutoParagraphStyle!  
   NewBulletStyle!  
   AutoCharacterStyle!  
   VariableType!};  
 [CreateFlag:Enumeration {  
   CreateFromCurrent!  
   CreateBulletStyle!  
   CreateNumberStyle!  
   CreateOutlineStyle!  
   CreateFromCurrentHS!});  
 [Library:Enumeration {  
   CurrentDoc!  
   PersonalLibrary!  
   SharedLibrary!});

[Create a style]

### StyleDelete (

Style:Any {  
   FootnoteNumberInDocumentStyle!  
   EndnoteNumberInDocumentStyle!  
   FigureNumberStyle!  
   TableBoxNumberStyle!  
   TextBoxNumberStyle!  
   UserBoxNumberStyle!  
   EquationNumberStyle!  
   HypertextStyle!  
   ToC1Style!  
   ToC2Style!  
   ToC3Style!  
   ToC4Style!  
   ToC5Style!  
   Index1Style!  
   Index2Style!  
   ListStyle!  
   ToAStyle!  
   DocStyle!  
   BoxTextStyle!  
   CaptionStyle!  
   FootnoteStyle!  
   EndnoteStyle!;

CommentStyle!  
 HeaderAStyle!  
 HeaderBStyle!  
 FooterAStyle!  
 FooterBStyle!  
 WatermarkAStyle!  
 WatermarkBStyle!  
 ParagraphOutInStyle!  
 OutlineStyle!  
 LegalOutInStyle!  
 BulletsOutInStyle!  
 HeadingsOutInStyle!  
 Legal\_2OutInStyle!  
 NumbersOutInStyle!  
 Level1Style!  
 Level2Style!  
 Level3Style!  
 Level4Style!  
 Level5Style!  
 Level6Style!  
 Level7Style!  
 Level8Style!  
 Legal1Style!  
 Legal2Style!  
 Legal3Style!  
 Legal4Style!  
 Legal5Style!  
 Legal6Style!  
 Legal7Style!  
 Legal8Style!  
 Heading1Style!  
 Heading2Style!  
 Heading3Style!  
 Heading4Style!  
 Heading5Style!  
 Heading6Style!  
 Heading7Style!  
 Heading8Style!  
 MonospacedStyle!  
 CustomHTMLStyle!  
 PreformattedStyle!  
 AddressStyle!  
 QuotationsOutInStyle!  
 DefinitionsOutInStyle!  
 Quotation1Style!  
 Quotation2Style!  
 Quotation3Style!  
 Quotation4Style!  
 Quotation5Style!  
 Quotation6Style!  
 Quotation7Style!  
 Quotation8Style!  
 Definition1Style!  
 Definition2Style!  
 Definition3Style!  
 Definition4Style!  
 Definition5Style!  
 Definition6Style!  
 Definition7Style!  
 Definition8Style!  
 Level9Style!  
 Legal9Style!  
 Heading9Style!  
 Quotation9Style!  
 Definition9Style!};  
 Codes:Enumeration {  
   IncludingCodes!  
   LeavingCodes!};  
 [Library:Enumeration {  
   CurrentDoc!  
   PersonalLibrary!;

**SharedLibrary!}}**  
[Delete a specified style]

**StyleDescription (Description:String)**  
[Change a style description]

**StyleEditBegin (**

**Style:Any {**  
FootnoteNumberInDocumentStyle!  
EndnoteNumberInDocumentStyle!  
FigureNumberStyle!  
TableBoxNumberStyle!  
TextBoxNumberStyle!  
UserBoxNumberStyle!  
EquationNumberStyle!  
HypertextStyle!  
ToC1Style!  
ToC2Style!  
ToC3Style!  
ToC4Style!  
ToC5Style!  
Index1Style!  
Index2Style!  
ListStyle!  
ToAStyle!  
DocStyle!  
BoxTextStyle!  
CaptionStyle!  
FootnoteStyle!  
EndnoteStyle!  
CommentStyle!  
HeaderAStyle!  
HeaderBStyle!  
FooterAStyle!  
FooterBStyle!  
WatermarkAStyle!  
WatermarkBStyle!  
ParagraphOutInStyle!  
OutlineStyle!  
LegalOutInStyle!  
BulletsOutInStyle!  
HeadingsOutInStyle!  
Legal\_2OutInStyle!  
NumbersOutInStyle!  
Level1Style!  
Level2Style!  
Level3Style!  
Level4Style!  
Level5Style!  
Level6Style!  
Level7Style!  
Level8Style!  
Legal1Style!  
Legal2Style!  
Legal3Style!  
Legal4Style!  
Legal5Style!  
Legal6Style!  
Legal7Style!  
Legal8Style!  
Heading1Style!  
Heading2Style!  
Heading3Style!  
Heading4Style!  
Heading5Style!  
Heading6Style!  
Heading7Style!  
Heading8Style!  
MonospacedStyle!  
CustomHTMLStyle!  
PreformattedStyle!  
AddressStyle!

QuotationsOutInStyle!  
DefinitionsOutInStyle!  
Quotation1Style!  
Quotation2Style!  
Quotation3Style!  
Quotation4Style!  
Quotation5Style!  
Quotation6Style!  
Quotation7Style!  
Quotation8Style!  
Definition1Style!  
Definition2Style!  
Definition3Style!  
Definition4Style!  
Definition5Style!  
Definition6Style!  
Definition7Style!  
Definition8Style!  
Level9Style!  
Legal9Style!  
Heading9Style!  
Quotation9Style!  
Definition9Style!};  
[Library:Enumeration {  
CurrentDoc!  
PersonalLibrary!  
SharedLibrary!}]}

[Begin editing a style]

**StyleEditEnd (State:Enumeration {Cancel!;Save!})**  
[Save or cancel style changes]

**StyleEnterKeySetting (Action:Enumeration {  
HardReturn!  
StyleOff!  
StyleOffOn!})**

[Specify the effect of pressing Enter when a style is on]

**StyleLinkTo (Style:Any {**  
FootnoteNumberInDocumentStyle!  
EndnoteNumberInDocumentStyle!  
FigureNumberStyle!  
TableBoxNumberStyle!  
TextBoxNumberStyle!  
UserBoxNumberStyle!  
EquationNumberStyle!  
HypertextStyle!  
ToC1Style!  
ToC2Style!  
ToC3Style!  
ToC4Style!  
ToC5Style!  
Index1Style!  
Index2Style!  
ListStyle!  
ToAStyle!  
DocStyle!  
BoxTextStyle!  
CaptionStyle!  
FootnoteStyle!  
EndnoteStyle!  
CommentStyle!  
HeaderAStyle!  
HeaderBStyle!  
FooterAStyle!  
FooterBStyle!  
WatermarkAStyle!  
WatermarkBStyle!  
ParagraphOutInStyle!  
OutlineStyle!  
LegalOutInStyle!  
BulletsOutInStyle!

HeadingsOutInStyle!  
 Legal\_2OutInStyle!  
 NumbersOutInStyle!  
 Level1Style!  
 Level2Style!  
 Level3Style!  
 Level4Style!  
 Level5Style!  
 Level6Style!  
 Level7Style!  
 Level8Style!  
 Legal1Style!  
 Legal2Style!  
 Legal3Style!  
 Legal4Style!  
 Legal5Style!  
 Legal6Style!  
 Legal7Style!  
 Legal8Style!  
 Heading1Style!  
 Heading2Style!  
 Heading3Style!  
 Heading4Style!  
 Heading5Style!  
 Heading6Style!  
 Heading7Style!  
 Heading8Style!  
 MonospacedStyle!  
 CustomHTMLStyle!  
 PreformattedStyle!  
 AddressStyle!  
 QuotationsOutInStyle!  
 DefinitionsOutInStyle!  
 Quotation1Style!  
 Quotation2Style!  
 Quotation3Style!  
 Quotation4Style!  
 Quotation5Style!  
 Quotation6Style!  
 Quotation7Style!  
 Quotation8Style!  
 Definition1Style!  
 Definition2Style!  
 Definition3Style!  
 Definition4Style!  
 Definition5Style!  
 Definition6Style!  
 Definition7Style!  
 Definition8Style!  
 Level9Style!  
 Legal9Style!  
 Heading9Style!  
 Quotation9Style!  
 Definition9Style!)

[Specify a style to turn on when Enter is pressed]

**StyleOff (**

[Style:Any {  
 FootnoteNumberInDocumentStyle!  
 EndnoteNumberInDocumentStyle!  
 FigureNumberStyle!  
 TableBoxNumberStyle!  
 TextBoxNumberStyle!  
 UserBoxNumberStyle!  
 EquationNumberStyle!  
 HypertextStyle!  
 ToC1Style!  
 ToC2Style!  
 ToC3Style!  
 ToC4Style!  
 ToC5Style!  
 Index1Style!

Index2Style!  
 ListStyle!  
 ToAStyle!  
 DocStyle!  
 BoxTextStyle!  
 CaptionStyle!  
 FootnoteStyle!  
 EndnoteStyle!  
 CommentStyle!  
 HeaderAStyle!  
 HeaderBStyle!  
 FooterAStyle!  
 FooterBStyle!  
 WatermarkAStyle!  
 WatermarkBStyle!  
 ParagraphOutInStyle!  
 OutlineStyle!  
 LegalOutInStyle!  
 BulletsOutInStyle!  
 HeadingsOutInStyle!  
 Legal\_2OutInStyle!  
 NumbersOutInStyle!  
 Level1Style!  
 Level2Style!  
 Level3Style!  
 Level4Style!  
 Level5Style!  
 Level6Style!  
 Level7Style!  
 Level8Style!  
 Legal1Style!  
 Legal2Style!  
 Legal3Style!  
 Legal4Style!  
 Legal5Style!  
 Legal6Style!  
 Legal7Style!  
 Legal8Style!  
 Heading1Style!  
 Heading2Style!  
 Heading3Style!  
 Heading4Style!  
 Heading5Style!  
 Heading6Style!  
 Heading7Style!  
 Heading8Style!  
 MonospacedStyle!  
 CustomHTMLStyle!  
 PreformattedStyle!  
 AddressStyle!  
 QuotationsOutInStyle!  
 DefinitionsOutInStyle!  
 Quotation1Style!  
 Quotation2Style!  
 Quotation3Style!  
 Quotation4Style!  
 Quotation5Style!  
 Quotation6Style!  
 Quotation7Style!  
 Quotation8Style!  
 Definition1Style!  
 Definition2Style!  
 Definition3Style!  
 Definition4Style!  
 Definition5Style!  
 Definition6Style!  
 Definition7Style!  
 Definition8Style!  
 Level9Style!  
 Legal9Style!  
 Heading9Style!  
 Quotation9Style!

```
Definition9Style!});  
[Type:Enumeration {  
  IsVariable!})]  
[Turn off a style]
```

```
StyleOn (  
  Style:Any {  
    NormalStyle!;  
    FootnoteNumberInDocumentStyle!;  
    EndnoteNumberInDocumentStyle!;  
    FigureNumberStyle!;  
    TableBoxNumberStyle!;  
    TextBoxNumberStyle!;  
    UserBoxNumberStyle!;  
    EquationNumberStyle!;  
    HypertextStyle!;  
    ToC1Style!;  
    ToC2Style!;  
    ToC3Style!;  
    ToC4Style!;  
    ToC5Style!;  
    Index1Style!;  
    Index2Style!;  
    ListStyle!;  
    ToAStyle!;  
    DocStyle!;  
    BoxTextStyle!;  
    CaptionStyle!;  
    FootnoteStyle!;  
    EndnoteStyle!;  
    CommentStyle!;  
    HeaderAStyle!;  
    HeaderBStyle!;  
    FooterAStyle!;  
    FooterBStyle!;  
    WatermarkAStyle!;  
    WatermarkBStyle!;  
    ParagraphOutInStyle!;  
    OutlineStyle!;  
    LegalOutInStyle!;  
    BulletsOutInStyle!;  
    Legal_2OutInStyle!;  
    NumbersOutInStyle!;  
    Level1Style!;  
    Level2Style!;  
    Level3Style!;  
    Level4Style!;  
    Level5Style!;  
    Level6Style!;  
    Level7Style!;  
    Level8Style!;  
    Legal1Style!;  
    Legal2Style!;  
    Legal3Style!;  
    Legal4Style!;  
    Legal5Style!;  
    Legal6Style!;  
    Legal7Style!;  
    Legal8Style!;  
    Heading1Style!;  
    Heading2Style!;  
    Heading3Style!;  
    Heading4Style!;  
    Heading5Style!;  
    Heading6Style!;  
    Heading7Style!;  
    Heading8Style!;  
    MonospacedStyle!;  
    CustomHTMLStyle!;  
    PreformattedStyle!;  
    AddressStyle!;  
    QuotationsOutInStyle!
```

```
DefinitionsOutInStyle!;  
Quotation1Style!;  
Quotation2Style!;  
Quotation3Style!;  
Quotation4Style!;  
Quotation5Style!;  
Quotation6Style!;  
Quotation7Style!;  
Quotation8Style!;  
Definition1Style!;  
Definition2Style!;  
Definition3Style!;  
Definition4Style!;  
Definition5Style!;  
Definition6Style!;  
Definition7Style!;  
Definition8Style!;  
Level9Style!;  
Legal9Style!;  
Heading9Style!;  
Quotation9Style!;  
Definition9Style!};  
[Action:Enumeration {  
  SkipEmptyParagraphs!})]  
[Turn on a style]
```

**StylePersonalLibrary** (Filename:String)  
[Specify a DOS Personal Library style file for the current document]

**StyleRename** (Name:String)  
[Rename a style]

```
StyleRetrieve (  
  File:String;  
  Styles:Enumeration {  
    UserStyles!;  
    SystemStyles!;  
    AllStyles!};  
  [Library:Enumeration {  
    CurrentDoc!;  
    PersonalLibrary!;  
    SharedLibrary!})]  
[Retrieve styles in a style file]
```

```
StyleSave (  
  Filename:String;  
  StyleType:Enumeration {  
    UserStyles!;  
    SystemStyles!;  
    AllStyles!};  
  [Library:Enumeration {  
    CurrentDoc!;  
    PersonalLibrary!;  
    SharedLibrary!})]  
[Save styles from a library to a style file]
```

**StyleSharedLibrary** (Filename:String)  
[Specify a DOS shared library style file for the current document]

**StylesListDlg** ()  
[Create, edit, and select styles]

```
StyleSystemOn (  
  Style:Any {  
    NormalStyle!;  
    FootnoteNumberInDocumentStyle!;  
    EndnoteNumberInDocumentStyle!;  
    FigureNumberStyle!;  
    TableBoxNumberStyle!;  
    TextBoxNumberStyle!;  
    UserBoxNumberStyle!;  
    EquationNumberStyle!;  
    HypertextStyle!
```

ToC1Style!;  
 ToC2Style!;  
 ToC3Style!;  
 ToC4Style!;  
 ToC5Style!;  
 Index1Style!;  
 Index2Style!;  
 ListStyle!;  
 ToAStyle!;  
 DocStyle!;  
 BoxTextStyle!;  
 CaptionStyle!;  
 FootnoteStyle!;  
 EndnoteStyle!;  
 CommentStyle!;  
 HeaderAStyle!;  
 HeaderBStyle!;  
 FooterAStyle!;  
 FooterBStyle!;  
 WatermarkAStyle!;  
 WatermarkBStyle!;  
 ParagraphOutInStyle!;  
 OutlineStyle!;  
 LegalOutInStyle!;  
 BulletsOutInStyle!;  
 HeadingsOutInStyle!;  
 Legal\_2OutInStyle!;  
 NumbersOutInStyle!;  
 Level1Style!;  
 Level2Style!;  
 Level3Style!;  
 Level4Style!;  
 Level5Style!;  
 Level6Style!;  
 Level7Style!;  
 Level8Style!;  
 Legal1Style!;  
 Legal2Style!;  
 Legal3Style!;  
 Legal4Style!;  
 Legal5Style!;  
 Legal6Style!;  
 Legal7Style!;  
 Legal8Style!;  
 Heading1Style!;  
 Heading2Style!;  
 Heading3Style!;  
 Heading4Style!;  
 Heading5Style!;  
 Heading6Style!;  
 Heading7Style!;  
 Heading8Style!;  
 MonospacedStyle!;  
 CustomHTMLStyle!;  
 PreformattedStyle!;  
 AddressStyle!;  
 QuotationsOutInStyle!;  
 DefinitionsOutInStyle!;  
 Quotation1Style!;  
 Quotation2Style!;  
 Quotation3Style!;  
 Quotation4Style!;  
 Quotation5Style!;  
 Quotation6Style!;  
 Quotation7Style!;  
 Quotation8Style!;  
 Definition1Style!;  
 Definition2Style!;  
 Definition3Style!;  
 Definition4Style!;  
 Definition5Style!;  
 Definition6Style!;

Definition7Style!;  
 Definition8Style!;  
 Level9Style!;  
 Legal9Style!;  
 Heading9Style!;  
 Quotation9Style!;  
 Definition9Style!);  
 [Action:Enumeration {  
   SkipEmptyParagraphs!});  
 [Type:Enumeration {  
   IsVariable!})  
 [Turn on a WordPerfect 10 or user-defined style]

**StyleType** (Type:Enumeration {  
   OpenStyle!;  
   CharacterStyle!;  
   ParagraphStyle!;  
   AutoParagraphStyle!;  
   NewBulletStyle!;  
   AutoCharacterStyle!;  
   VariableType!})  
 [Specify a style type]

**StyleUpdate** ()  
 [Update all styles in the current document]

**SubdividePage** (  
   Columns:Numeric;  
   Rows:Numeric)  
 [Divide a physical page into logical pages]

**SubdividePageDlg** ()  
 [Divide page into multiple pages]

**SubdividePageOff** ()  
 [End logical page breaks]

**SubdocInclude** (DocumentName:String)  
 [Insert a subdocument into the current document]

**SubstructureExit** ([Action:Enumeration {  
   Previous!;  
   Next!}))  
 [Close the current substructure]

**Suppress** ([Item:Enumeration {  
   PageNumbering!;  
   PageNumberBottomCenter!;  
   HeaderA!;  
   HeaderB!;  
   FooterA!;  
   FooterB!;  
   WatermarkA!;  
   WatermarkB!}))  
 [Turn off the specified page formatting items for the current page]

**SwitchDoc** (DocNum:Numeric)  
 [Specify a document to switch to]

- T -

**Tab** ()  
 [Insert a tab]

**TabAdd** (  
   Position:Numeric Measurement;  
   Type:Enumeration {  
     TabLeft!;  
     TabCenter!;  
     TabRight!;  
     TabDecimal!;  
     TabLeftDot!;  
     TabCenterDot!;

**TabRightDot!**;  
**TabDecimalDot!;**  
**[SpaceBetweenTabs:Numeric Measurement]**  
 [Add one or more tab sets to the current tab setting]

**Boolean := TabBarIconShow (State:Enumeration {Off!;On!})**  
 [Turn the Tab Bar icons on or off and return the current state]

**TabCenter ([Type:Enumeration {Normal!;DotLeader!}])**  
 [Insert a hard centered tab]

**TabCenterDot ()**  
 [Insert a hard centered tab with dot leader]

**TabClearAll ()**  
 [Clear all tab settings]

**TabDecimal ([Type:Enumeration {Normal!;DotLeader!}])**  
 [Insert a hard decimal-aligned tab]

**TabDecimalDot ()**  
 [Insert a hard decimal-aligned tab with dot leader]

**TabDelete (Position:Numeric Measurement)**  
 [Delete one tab setting without affecting any other tab settings]

**TabForced ()**  
 [Insert a forced tab]

**TabInsert ()**  
 [Insert a tab into the document]

**TabKey ()**  
 [Insert a tab, go to the next table cell, or go to the next outline level]

**TableAdjustBegin ()**  
 [Begin table adjustments from the Ruler]

**TableAdjustColumnWidth ( Column:Numeric; Width:Numeric Measurement)**  
 [Specify a column width in WordPerfect units when adjusting columns from the Ruler]

**TableAdjustEnd ()**  
 [End table adjustments from the Ruler]

**TableAdjustPosition (Position:Numeric Measurement)**  
 [Adjust the position of a table relative to the left margin]

**TableAlternateFillSet ( State:Enumeration { Off!; CheckerBoard!; Row!; Column!}; [Span:Numeric]; [AlternateSpan:Numeric])**  
 [Specify the type of alternating table fill and the number of cells to which the fill is applied]

**TableAppendRow ()**  
 [Insert a new row below the current row]

**TableApplyStyle ( Style:Any { TableStyle1!; TableStyle2!; TableStyle3!; TableStyle4!; TableStyle5!; TableStyle6!; TableStyle7!;**

**TableStyle8!;**  
**TableStyle9!;**  
**TableStyle10!;**  
**TableStyle11!;**  
**TableStyle12!;**  
**TableStyle13!;**  
**TableStyle14!;**  
**TableStyle15!;**  
**TableStyle16!;**  
**TableStyle17!;**  
**TableStyle18!;**  
**TableStyle19!;**  
**TableStyle20!;**  
**TableStyle21!;**  
**TableStyle22!;**  
**TableStyle23!;**  
**TableStyle24!;**  
**TableStyle25!;**  
**TableStyle26!;**  
**TableStyle27!;**  
**TableStyle28!;**  
**TableStyle29!;**  
**TableStyle30!;**  
**TableStyle31!;**  
**TableStyle32!;**  
**TableStyle33!;**  
**TableStyle34!;**  
**TableStyle35!;**  
**TableStyle36!;**  
**TableStyle37!;**  
**TableStyle38!;**  
**TableStyle39!;**  
**TableStyle40!;**  
**TableStyle41!;**  
**TableStyle42!;**  
**TableStyle43!;**  
**TableStyle44!;**  
**NoTableStyle!;**  
**[ResetAttributes:Enumeration {No!; Yes!};**  
**[ShowIn60:Enumeration {No!; Yes!}])**  
 [Apply a table style using Table SpeedFormat]

**TableAttributeOff ({[Attribute:Enumeration { ExtraLarge!; VeryLarge!; Large!; Small!; Fine!; Superscript!; Subscript!; Outline!; Italics!; Shadow!; Redline!; DoubleUnderline!; Bold!; Strikeout!; Underline!; SmallCaps!; Every!}])**  
 [Turn off specified font attributes and relative font size attributes in a table]

**TableAttributeOn ({[Attribute:Enumeration { ExtraLarge!; VeryLarge!; Large!; Small!; Fine!; Superscript!; Subscript!; Outline!;**

**Italics!**;  
**Shadow!**;  
**Redline!**;  
**DoubleUnderline!**;  
**Bold!**;  
**Strikeout!**;  
**Underline!**;  
**SmallCaps!}}**)

[Turn on specified character attributes in a table]

**TableAutoCalc** (State:Enumeration {  
**Off!**;  
**AutoTbl!**;  
**AutoDoc!}}**)

[Specify automatic calculation for tables]

**TableAutoFill** ()

[Increment values across a row or down a column]

**TableBlockOff** ()

[Turn off Select in a table]

**TableBlockOn** ()

[Turn on Select in a table]

**TableBorder** (Style:Any {

**SpacingOnly!**;  
**SingleBorder!**;  
**DoubleBorder!**;  
**DashedBorder!**;  
**DottedBorder!**;  
**ThickBorder!**;  
**ExtraThickBorder!**;  
**ThinThickBorder!**;  
**ThickThinBorder!**;  
**ThickTopBottomBorder!**;  
**ButtonBorder!**;  
**ColumnBorderBetween!**;  
**ColumnBorderAll!**;  
**HeavySingleBorder!**;  
**HeavyDoubleBorder!**;  
**NoBorder!}}**)

[Specify a table border type]

**TableBorderEditBegin** ()

[Begin table border editing]

**TableBorderEditEnd** (State:Enumeration {Cancel!;Save!})

[Save table border editing changes]

**TableCalculate** ()

[Calculate formulas in the current table or selected cells]

**TableCalculateAll** ()

[Change the automatic calculation mode]

**TableCalculateDig** ()

[Calculate the formulas in all tables of the current document]

**TableCellAttributeOff** ({Attribute:Enumeration {

**ExtraLarge!**;  
**VeryLarge!**;  
**Large!**;  
**Small!**;  
**Fine!**;  
**Superscript!**;  
**Subscript!**;  
**Outline!**;  
**Italics!**;  
**Shadow!**;  
**Redline!**;  
**DoubleUnderline!**;  
**Bold!**;

**Strikeout!**;  
**Underline!**;  
**SmallCaps!**;  
**Every!}}**)

[Turn off the specified font appearance, position, and relative size attributes in the selected cells]

**TableCellAttributeOn** ({Attribute:Enumeration {

**ExtraLarge!**;  
**VeryLarge!**;  
**Large!**;  
**Small!**;  
**Fine!**;  
**Superscript!**;  
**Subscript!**;  
**Outline!**;  
**Italics!**;  
**Shadow!**;  
**Redline!**;  
**DoubleUnderline!**;  
**Bold!**;  
**Strikeout!**;  
**Underline!**;  
**SmallCaps!}}**)

[Apply the specified font appearance, position, and relative size attributes to the selected cells]

**TableCellCopy** (

**CopyType:Enumeration** {  
**Down!**;  
**Right!**;  
**Cell!**};  
**[Count:Numeric]**)

[Copy the cell content to another cell]

**TableCellDiagonalLine** (Diagonal:Enumeration {

**None!**;  
**TopLeftToBotRight!**;  
**BotLeftToTopRight!**;  
**TwoDiagonals!}}**)

[Add diagonal lines to a table cell]

**TableCellFillColors** (

**[ForeColorName:String]**;  
**[ForeColorRed:Numeric]**;  
**[ForeColorGreen:Numeric]**;  
**[ForeColorBlue:Numeric]**;  
**[ForeColorShade:Numeric]**;  
**[BackColorName:String]**;  
**[BackColorRed:Numeric]**;  
**[BackColorGreen:Numeric]**;  
**[BackColorBlue:Numeric]**;  
**[BackColorShade:Numeric]**)

[Specify the foreground and background colors for the current cell or selected cells]

**TableCellFillStyle** (FillPattern:Any {

**Fill10!**;  
**Fill20!**;  
**Fill30!**;  
**Fill40!**;  
**Fill50!**;  
**Fill60!**;  
**Fill70!**;  
**Fill80!**;  
**Fill90!**;  
**Fill100!**;  
**FillButton!**;  
**Fill5!**;  
**NoFill!}}**)

[Specify a shading style for the active cell or selected cells]

**TableCellIgnoreCalculation (State:Enumeration {No!;Yes!})**

[Specify one or more selected cells to ignore when calculating formulas]

**TableCellJustification (Mode:Enumeration {**

**Left!;**  
**Full!;**  
**Center!;**  
**Right!;**  
**FullAll!;**  
**DecAlign!})**

[Specify justification for the current cell or selected cells]

**TableCellLine (**

**Position:Enumeration {**

**Left!;**  
**Top!;**  
**Inside!;**  
**Right!;**  
**Bottom!;**  
**Outside!};**

**Style:Any {**

**SingleLine!;**  
**DoubleLine!;**  
**DashedLine!;**  
**Dotte dLine!;**  
**ThickLine!;**  
**ExtraThickLine!;**  
**ThinThickLine!;**  
**ThickThinLine!;**  
**ButtonTopLeftLine!;**  
**ButtonBottomRightLine!;**  
**HeavySingleLine!;**  
**HeavyDoubleLine!;**  
**TableDefaultLine!;**  
**NoLine!})**

[Change the specified lines around a cell or selected cells, or between selected cells]

**TableCellLineColor (**

**[ColorName:String];**  
**[Red:Numeric];**  
**[Green:Numeric];**  
**[Blue:Numeric];**  
**[Shade:Numeric])**

[Specify the line color of the current cell or selected cells]

**TableCellLock (State:Enumeration {No!;Yes!})**

[Lock the current cell or selected cells]

**TableCellLockToggle ()**

[Lock/unlock the current cell or selected cells]

**TableCellNumberAlignCurrency (State:Enumeration {No!;Yes!})**

[Align the currency symbols in the current cell or selected cells]

**TableCellNumberDateFormat (DateFormatNumber:Numeric)**

[Specify a date format for one or more selected cells]

**TableCellNumberDecimalDigits (NumDigits:Numeric)**

[Specify how many digits appear after the decimal]

**TableCellNumberFormat (FormatType:Enumeration {**

**General!;**  
**Integer!;**  
**Fixed2!;**  
**Percent!;**  
**Currency!;**  
**Accounting!;**  
**Commas!;**  
**Scientific!;**  
**DateFormat!;**

**TextOnly!})**

[Specify a number format for the current cell or selected cells]

**TableCellNumberNegativeNumber (NegDisplayType:Enumeration {**

**Minus!;**  
**Parentheses!;**  
**CRDR!})**

[Specify how to display negative numbers in the current cell or selected cells]

**TableCellNumberRound (State:Enumeration {No!;Yes!})**

[Round numbers entered as decimals to convert to integers]

**TableCellNumberSelectCurrency (IntlCurrency:Enumeration {**

**DefaultCurrency!;**  
**Dollar!;**  
**Peso!;**  
**FrenchCanada!;**  
**Canada!;**  
**Brazil!;**  
**Denmark!;**  
**Peseta!;**  
**Croatia!;**  
**CzechRepublic!;**  
**SlovakRepublic!;**  
**Finland!;**  
**FinlandIntl!;**  
**France!;**  
**Germany!;**  
**Greece!;**  
**Hungary!;**  
**Iceland!;**  
**Italy!;**  
**Japan!;**  
**Netherlands!;**  
**Norway!;**  
**NorwayIntl!;**  
**Ruble!;**  
**RussianRuble!;**  
**SouthAfrica!;**  
**Sweden!;**  
**Turkey!;**  
**UnitedKingdom!;**  
**Switzerland!;**  
**Austria!;**  
**Belgium!;**  
**Ecuador!;**  
**ElSalvador!;**  
**Guatemala!;**  
**Honduras!;**  
**Paraguay!;**  
**Venezuela!;**  
**Poland!;**  
**Slovenia!;**  
**Israel!;**  
**Korea!;**  
**China!;**  
**Taiwan!;**  
**HongKong!;**  
**Singapore!;**  
**Portugal!;**  
**BelgiumFrench!;**  
**Euro!})**

[Specify an international currency type for the current cell or selected cells]

**TableCellNumberUseCommas (State:Enumeration {No!;Yes!})**

[Display commas in numbers with more than three digits to the left of the decimal]

**TableCellNumberUseCurrency (State:Enumeration {No!;Yes!})**

[Display a currency symbol with numbers]



**TableCellRotateContents** (Angle:Enumeration {

None!;  
Degrees90!;  
Degrees180!;  
Degrees270!}}

[Rotate the contents of a table cell]

**TableCellsJoin** ()

[Join the selected table cells to form one cell]

**TableCellUseColumnAttribute** (State:Enumeration {No!;Yes!})

[Apply a specified column attribute to the current cell or selected cells]

**TableCellUseColumnJust** (State:Enumeration {No!;Yes!})

[Turn column justification on or off in the current cell]

**TableCellUseColumnNumberType** (State:Enumeration {No!;Yes!})

[Turn the current column number type off or on in the current cell or selected cells]

**TableCellUseLineColor** ()

[Turn on table line color for the current cell or selected cells]

**TableCellVerticalAlignment** (Alignment:Enumeration {

Top!;  
Center!;  
Bottom!})

[Align the current cell or selected cells vertically]

**TableColumnAttributeOff** ({Attribute:Enumeration {

ExtraLarge!;  
VeryLarge!;  
Large!;  
Small!;  
Fine!;  
Superscript!;  
Subscript!;  
Outline!;  
Italics!;  
Shadow!;  
Redline!;  
DoubleUnderline!;  
Bold!;  
Strikeout!;  
Underline!;  
SmallCaps!;  
Every!}}})

[Turn off specified character attributes in a table column]

**TableColumnAttributeOn** ({Attribute:Enumeration {

ExtraLarge!;  
VeryLarge!;  
Large!;  
Small!;  
Fine!;  
Superscript!;  
Subscript!;  
Outline!;  
Italics!;  
Shadow!;  
Redline!;  
DoubleUnderline!;  
Bold!;  
Strikeout!;  
Underline!;  
SmallCaps!}}})

[Turn on specified character attributes in a table column]

**TableColumnDecimalAlignDigits** (Digits:Numeric)

[Specify the number of digits to display between the decimal point and the right edge of cells]

**TableColumnDecimalAlignDist** (Distance:Numeric Measurement)

[Specify the distance between the decimal and the right edge of cells]

**TableColumnFixedWidth** (State:Enumeration {No!;Yes!})

[Specify a fixed column width]

**TableColumnJustification** (Mode:Enumeration {

Left!;  
Full!;  
Center!;  
Right!;  
FullAll!;  
DecAlign!})

[Specify column justification]

**TableColumnMarginLeft** (Margin:Numeric Measurement)

[Specify the distance between the left edge of a column and the text in the column]

**TableColumnMarginRight** (Margin:Numeric Measurement)

[Specify the distance between the right edge of a column and the text in the column]

**TableColumnNumberAlignCurrency** (State:Enumeration {No!;Yes!})

[Align currency symbols in a column]

**TableColumnNumberDateFormat** (DateFmtNum:Numeric)

[Specify a date format for the current column]

**TableColumnNumberDecimalDigits** (NumDigits:Numeric)

[Specify the number of digits after the decimal]

**TableColumnNumberFormat** (FormatType:Enumeration {

General!;  
Integer!;  
Fixed2!;  
Percent!;  
Currency!;  
Accounting!;  
Commas!;  
Scientific!;  
DateFormat!;  
TextOnly!})

[Specify a number format for the current column]

**TableColumnNumberNegNumber** (NegDisplayType:Enumeration {

Minus!;  
Parentheses!;  
CRDR!})

[Specify how to display negative numbers in the current column]

**TableColumnNumberRound** (State:Enumeration {No!;Yes!})

[Round numbers entered as decimals to integers in the current column]

**TableColumnNumberSelCurrency** (IntlCurrency:Enumeration {

DefaultCurrency!;  
Dollar!;  
Peso!;  
FrenchCanada!;  
Canada!;  
Brazil!;  
Denmark!;  
Peseta!;  
Croatia!;  
CzechRepublic!;  
SlovakRepublic!;  
Finland!;  
FinlandIntl!;  
France!;  
Germany!;

Greece!;  
 Hungary!;  
 Iceland!;  
 Italy!;  
 Japan!;  
 Netherlands!;  
 Norway!;  
 NorwayIntl!;  
 Ruble!;  
 RussianRuble!;  
 SouthAfrica!;  
 Sweden!;  
 Turkey!;  
 UnitedKingdom!;  
 Switzerland!;  
 Austria!;  
 Belgium!;  
 Ecuador!;  
 ElSalvador!;  
 Guatemala!;  
 Honduras!;  
 Paraguay!;  
 Venezuela!;  
 Poland!;  
 Slovenia!;  
 Israel!;  
 Korea!;  
 China!;  
 Taiwan!;  
 HongKong!;  
 Singapore!;  
 Portugal!;  
 BelgiumFrench!;  
 Euro!))

[Specify a currency type for the current column]

**TableColumnNumberUseCommas** (State:Enumeration {No!;Yes!})

[Display commas in numbers with three or more digits to the left of the decimal]

**TableColumnNumberUseCurrency** (State:Enumeration {No!;Yes!})

[Display a currency symbol with numbers]

**TableColumnsFixedWidth** (State:Enumeration {No!;Yes!})

[Turn on the Fixed width option for all columns in the current table]

**TableColumnSize2Fit** ()

[Adjust the column width to fit the data]

**TableColumnWidth** (ColumnWidth:Numeric Measurement)

[Specify the width of the current column]

**TableColumnWidthEqualize** ()

[Make the selected columns have the same widths]

**TableColumnWidthGrow** ()

[Increase the size of a table column]

**TableColumnWidthShrink** ()

[Decrease the size of a table column]

**TableConvertDig** ()

[Convert text to tables or tables to text]

**TableConvertFrom** (

Type:Enumeration {  
 TabularColumns!;  
 ParallelColumns!;  
 MergeDataFile!;  
 DelimitedColumns!};

[DelimiterChar:String]

[Convert the selected text to a table]

**TableCopy** (Copy:Enumeration {  
 Block!;  
 Row!;  
 Column!})

[Copy the contents of the specified cells to the current column or the current row]

**TableCopyFormula** (  
 CellIDCntToCell:Enumeration {  
 Down!;  
 Up!;  
 CopyLeft!;  
 Right!;  
 Cell!};  
 [Where:Any])

[Copy a formula from the current cell to another cell]

**TableCreate** (  
 Columns:Numeric;  
 Rows:Numeric)

[Click and drag to create a table]

**TableCreateDig** ()

[Create a table or floating cell]

**TableCreateNewDig** ()

[Create rows and columns of text or data]

**TableDataFill** ()

[Extend the pattern in the current selection]

**TableDecimalAlignmentDigits** ([Number:Numeric])

[Specify the number of digits between the decimal point and the right edge of cells]

**TableDecimalAlignmentDistance** ([Position:Numeric Measurement])

[Specify the distance between the decimal and the right edge of cells]

**TableDefaultLine** (Style:Any {

SingleLine!;  
 DoubleLine!;  
 DashedLine!;  
 DottedLine!;  
 ThickLine!;  
 ExtraThickLine!;  
 ThinThickLine!;  
 ThickThinLine!;  
 ButtonTopLeftLine!;  
 ButtonBottomRightLine!;  
 HeavySingleLine!;  
 HeavyDoubleLine!;  
 TableDefaultLine!;  
 NoLine!})

[Specify the default line style of a table]

**TableDefaultLineColor** (

[ColorName:String];  
 [Red:Numeric];  
 [Green:Numeric];  
 [Blue:Numeric];  
 [Shade:Numeric])

[Specify the default line color for a table]

**TableDeleteBlock** ()

[Delete the contents of selected cells]

**TableDeleteCellFormula** ()

[Delete the formulas in the current cell]

**TableDeleteColumn (Number:Numeric)**

[Delete table column(s)]

**TableDeleteDlg ()**

[Delete rows, columns, or data]

**TableDeleteRow ([Number:Numeric])**

[Delete table row(s)]

**TableDeleteTable (**

**DelWhat:Enumeration {**  
**EntireTable!;**  
**ConvertOnly!;**  
**ConvertToTabs!;**  
**ConvertToMerge!;**  
**ConvertToMergeNames!;**  
**ConvertToDelimitedText!;**

[Delimiter:String])

[Delete a table and/or its contents, or convert table contents to tabbed text or merge fields]

**TableDeleteTableDlg ()**

[Delete a table and/or its contents, or convert table contents to tabbed text or merge fields]

**TableDisableCellLocks (State:Enumeration {No!;Yes!})**

[Turn off cell locks]

**TableDragToCreate ([State:Enumeration {Off!;On!}])**

[Turn the Drag to Create Tables option on or off]

**TableEdit ()**

[Begin a table editing command sequence]

**TableExpertDlg ()**

[Change the style of the current table, or create a new style]

**TableFill (Pattern:Any {**

**Fill10!;**  
**Fill20!;**  
**Fill30!;**  
**Fill40!;**  
**Fill50!;**  
**Fill60!;**  
**Fill70!;**  
**Fill80!;**  
**Fill90!;**  
**Fill100!;**  
**FillButton!;**  
**Fill5!;**  
**NoFill!})**

[Fill a table with a specified fill style]

**TableFormatDlg ()**

[Format cells, rows, columns, or tables]

**TableFormatJustification (Justification:Enumeration {**

**Left!;**  
**Full!;**  
**Center!;**  
**Right!;**  
**FullAll!;**  
**DecAlign!})**

[Specify table justification]

**TableFormatWidths (Width:Numeric Measurement)**

[Set table column width]

**TableFormula (Formula:String)**

[Specify a formula for the current cell]

**TableFormulaCBar ()**

[View and edit formulas in a table]

**TableLeft ([Type:Enumeration {Normal!;DotLeader!}])**

[Insert a hard left-aligned tab]

**TableLeftDot ()**

[Insert a hard left-aligned tab with dot leader]

**TableGridLinesToggle ()**

[Toggle table gridlines]

**TableGuides (State:Enumeration {Off!;On!}])**

[Turn Table Guides on or off]

**TableHardColumnBreak ()**

[Insert a column break and continue the table in the next column]

**TableHardRow ()**

[Insert a page break and continue the table on the next page]

**TableHeader (State:Enumeration {No!;Yes!}])**

[Specify the current row as a header row]

**TableInsertAutoRow (State:Enumeration {No!;Yes!}])**

[Turn Auto Row Insert on or off]

**TableInsertColumn (**

**Number:Numeric;**  
**InsertPos:Enumeration {Before!; After!};**  
**[PreserveWidths:Enumeration {No!; Yes!}])**

[Insert table columns before or after the insertion point]

**TableInsertDlg ()**

[Insert new rows or columns]

**TableInsertOneRow ()**

[Insert a new row above the current row]

**TableInsertRow (**

**NumberOfRows:Numeric;**  
**InsertPosition:Enumeration {Before!; After!}])**

[Insert rows before or after the insertion point]

**TableJoin ()**

[Combine two adjacent tables into one]

**TableJoinCells ()**

[Combine the selected cells into one]

**TableJoinCellTool ()**

[Combine cells into one by dragging]

**TableLineFillDlg ()**

[Change borders or shading for a cell or table]

**TableMarginLeft (Margin:Numeric Measurement)**

[Specify the distance from the left edge of all cells in a table to cell text]

**TableMarginRight (Margin:Numeric Measurement)**

[Specify the distance from the right edge of all cells in a table to cell text]

**TableMove (Move:Enumeration {**

**Block!;**  
**Row!;**  
**Column!;**  
**Cell!})**

[Move the contents of the current cell, selected cells, current column, or current row]

**TableMoveCopyDlg ()**

[Copy the current formula to specified cells]

**TableMoveModeCancel ()**

[Cancel TableMove or TableCopy]

**TableMoveModeEnd ()**

[Paste contents moved or copied with TableMove or TableCopy into a table at the insertion point]

**TableNameAdd (**

**RefType:Enumeration {**  
**Block!;**  
**Row!;**  
**Column!;**  
**Cell!;**  
**Table!;**  
**FloatingCell!;**  
**CellsDown!;**  
**CellsRight!};**  
**[Name:String]**

[Name a cell, block of cells, row, column, or table]

**TableNameChange (**

**OldName:String;**  
**[NewName:String];**  
**[NewRef:String]**

[Rename a table]

**TableNameCreateDlg ()**

[Name a column, row, or range of cells]

**TableNameDelete (TableName:String)**

[Delete a table name]

**TableNameListDlg ()**

[Name cells, rows, or columns; insert names into formulas]

**TableNumberAlignCurrency (State:Enumeration {No!;Yes!})**

[Align currency symbols in a table]

**TableNumberDateFormat (DateFmtNum:Numeric)**

[Specify a date format for the current table]

**TableNumberDecimalDigits (NumDigits:Numeric)**

[Specify the number of digits after the decimal]

**TableNumberFormat (FormatType:Enumeration {**

**General!;**  
**Integer!;**  
**Fixed2!;**  
**Percent!;**  
**Currency!;**  
**Accounting!;**  
**Commas!;**  
**Scientific!;**  
**DateFormat!;**  
**TextOnly!})**

[Specify a numbering type for the current table]

**TableNumberNegativeNumber (NegDisplay:Enumeration {**

**Minus!;**  
**Parentheses!;**  
**CRDR!})**

[Specify the display of negative numbers in the current table]

**TableNumberRound (State:Enumeration {No!;Yes!})**

[Round decimal numbers to integers]

**TableNumberSelectCurrency (IntlCurrency:Enumeration {**

**DefaultCurrency!;**  
**Dollar!;**  
**Peso!;**  
**FrenchCanada!;**  
**Canada!;**  
**Brazil!;**  
**Denmark!;**  
**Peseta!;**  
**Croatia!;**

**CzechRepublic!;**

**SlovakRepublic!;**

**Finland!;**

**FinlandIntl!;**

**France!;**

**Germany!;**

**Greece!;**

**Hungary!;**

**Iceland!;**

**Italy!;**

**Japan!;**

**Netherlands!;**

**Norway!;**

**NorwayIntl!;**

**Ruble!;**

**RussianRuble!;**

**SouthAfrica!;**

**Sweden!;**

**Turkey!;**

**UnitedKingdom!;**

**Switzerland!;**

**Austria!;**

**Belgium!;**

**Ecuador!;**

**ElSalvador!;**

**Guatemala!;**

**Honduras!;**

**Paraguay!;**

**Venezuela!;**

**Poland!;**

**Slovenia!;**

**Israel!;**

**Korea!;**

**China!;**

**Taiwan!;**

**HongKong!;**

**Singapore!;**

**Portugal!;**

**BelgiumFrench!;**

**Euro!})**

[Specify a currency type for the current table]

**TableNumberTypeDlg ()**

[Specify the number format for cells, columns, or tables]

**TableNumberUseCommas (State:Enumeration {No!;Yes!})**

[Display commas in numbers with more than three digits to the left of the decimal]

**TableNumberUseCurrency (State:Enumeration {No!;Yes!})**

[Display a currency symbol with numbers]

**TablePointMode (State:Enumeration {Off!;On!})**

[Toggle Formula Edit in a table]

**TablePosition (**

**PositionVal:Enumeration {**  
**AlignLeft!;**  
**AlignRight!;**  
**AlignCenter!;**  
**AlignFull!;**  
**AlignAbsolute!};**  
**[SetAbsPos:Numeric Measurement]**

[Specify table position]

**TablePosNextTable ()**

[Move the insertion point to the beginning of the next table]

**TablePosPreviousTable ()**

[Move the insertion point to the beginning of the previous table]

**TableQuickSum ()**

[Calculate the total of the cells above or to the left of the current cell]

**TableReformatAll ()**

[Update the currency and date cells to match the new language or date code]

**TableRetrieve (Reference:Enumeration {**

**Block!;**  
**Row!;**  
**Column!;**  
**Cell!})**

[Retrieve the last item moved or copied in a table]

**TableRotateCell ()**

[Rotate the contents of the current cell by 90 degrees clockwise]

**TableRowCanSpanPages (State:Enumeration {No!;Yes!})**

[Allow a table row to span two or more pages]

**TableRowColIndicators ()**

[Turn row/column indicators on or off]

**TableRowHeight (**

**RowHeight:Enumeration {**  
**Auto!;**  
**Fixed!};**  
**[Amount:Numeric Measurement]**

[Specify whether row height is automatic or fixed, and if fixed, specify the height]

**TableRowMarginBottom (Margin:Numeric Measurement)**

[Specify the distance between the last line of text and the bottom of the cell]

**TableRowMarginTop (Margin:Numeric Measurement)**

[Specify the distance between the top of a cell and first line of text in the cell]

**TableRowNumberOfLines (RowLines:Enumeration {Multiple!;Single!})**

[Specify single or multiple lines in the cells of a row]

**TableSecondFill (FillStyle:Any {**

**Fill10!;**  
**Fill20!;**  
**Fill30!;**  
**Fill40!;**  
**Fill50!;**  
**Fill60!;**  
**Fill70!;**  
**Fill80!;**  
**Fill90!;**  
**Fill100!;**  
**FillButton!;**  
**Fill5!;**  
**NoFill!})**

[Specify an alternating fill style for a table]

**TableSecondFillColors (**

**[ForeColor:String];**  
**[ForeRed:Numeric];**  
**[ForeGreen:Numeric];**  
**[ForeBlue:Numeric];**  
**[ForeShade:Numeric];**  
**[BackColor:String];**  
**[BackRed:Numeric];**  
**[BackGreen:Numeric];**  
**[BackBlue:Numeric];**  
**[BackShade:Numeric])**

[Specify a color for the alternating fill style of a table]

**TableSelectOff ()**

[Turn off Select]

**TableSelectOn (SelectionMode:Enumeration {**

**Row!;**  
**Column!;**  
**Cell!;**  
**Table!})**

[Select a row, column, cell, or table]

**TableSize ([Rows:Numeric]; [Columns:Numeric])**

[Change the number of rows or columns in an existing table]

**TableSkewColumn (**

**Column:Enumeration {**  
**FirstColumn!;**  
**LastColumn!};**  
**SkewValue:Numeric)**

[Set a skew value for the first or last column of the current table]

**TableSkewColumnText (State:Enumeration {No!;Yes!})**

[Skew/unskew text in a skewed table column]

**TableSkewJoinCorners (State:Enumeration {No!;Yes!})**

[Join corners in a skewed table (adjusts row height or column width)]

**TableSkewRow (SkewValue:Numeric)**

[Set a skew value for the top row of the current table]

**TableSkewRowText (State:Enumeration {No!;Yes!})**

[Skew/unskew text in a skewed table row]

**TableSortTable (**

**{[SortType:Enumeration {**  
**Alphanumeric!;**  
**Numeric!;**  
**DateSort!};**  
**[SortOrder:Enumeration {**  
**Ascending!;**  
**Descending!};**  
**})**

[Sort the table based on the cells selected]

**TableSplit ()**

[Divide a table at the current row]

**TableSplitCellsDlg ()**

[Divide the selected cells into multiple rows or columns]

**TableSplitColumn (**

**ColumnCount:Numeric;**  
**[ColumnPos:Numeric Measurement];**  
**[CurrentCol:Numeric])**

[Split the current cell into multiple columns]

**TableSplitColumnTool ([State:Enumeration {Off!;On!}])**

[Split or create new columns by clicking or dragging]

**TableSplitColumnsAndRowsTool ([State:Enumeration {Off!;On!}])**

[Split a cell into rows and columns]

**TableSplitRow (**

**RowCount:Numeric;**  
**[RowPos:Numeric])**

[Split the current cell into multiple rows]

**TableSplitRowCol (**

**RowCount:Numeric;**  
**ColumnCount:Numeric;**  
**[ColumnPos:Numeric Measurement];**  
**[RowPos:Numeric])**

[Split a cell into rows and columns]

**TableSquareSkewedRow (**

**Column:Enumeration {**  
**FirstColumn!;**

**LastColumn!;**  
**State:Enumeration {No!; Yes!}**  
[Square the first or last column of a skewed table row]

**TableSplitRowTool** (**State:Enumeration {Off!;On!}**)  
[Split or create new rows by clicking or dragging]

**TableStyleCreate** (  
**Style:String;**  
**[Library:Enumeration {**  
**CurrentDoc!;**  
**PersonalLibrary!;**  
**SharedLibrary!}]**)  
[Create a table style]

**TableStyleDelete** (  
**Style:String;**  
**[Library:Enumeration {**  
**CurrentDoc!;**  
**PersonalLibrary!;**  
**SharedLibrary!}]**)  
[Delete a table style]

**TableStyleEdit** (  
**Style:Any {**  
**TableStyle1!;**  
**TableStyle2!;**  
**TableStyle3!;**  
**TableStyle4!;**  
**TableStyle5!;**  
**TableStyle6!;**  
**TableStyle7!;**  
**TableStyle8!;**  
**TableStyle9!;**  
**TableStyle10!;**  
**TableStyle11!;**  
**TableStyle12!;**  
**TableStyle13!;**  
**TableStyle14!;**  
**TableStyle15!;**  
**TableStyle16!;**  
**TableStyle17!;**  
**TableStyle18!;**  
**TableStyle19!;**  
**TableStyle20!;**  
**TableStyle21!;**  
**TableStyle22!;**  
**TableStyle23!;**  
**TableStyle24!;**  
**TableStyle25!;**  
**TableStyle26!;**  
**TableStyle27!;**  
**TableStyle28!;**  
**TableStyle29!;**  
**TableStyle30!;**  
**TableStyle31!;**  
**TableStyle32!;**  
**TableStyle33!;**  
**TableStyle34!;**  
**TableStyle35!;**  
**TableStyle36!;**  
**TableStyle37!;**  
**TableStyle38!;**  
**TableStyle39!;**  
**TableStyle40!;**  
**TableStyle41!;**  
**TableStyle42!;**  
**TableStyle43!;**  
**TableStyle44!;**  
**NoTableStyle!;**  
**[Library:Enumeration {**  
**CurrentDoc!;**

**PersonalLibrary!;**  
**SharedLibrary!}]**)  
[Edit a table style]

**TableStyleEnd** (**State:Enumeration {Cancel!;Save!}**)  
[Save or cancel table style settings]

**TableStyleName** (**Style:String**)  
[Specify the name of a new table style]

**TableStyleReplace** (  
**Style:String;**  
**[Library:Enumeration {**  
**CurrentDoc!;**  
**PersonalLibrary!;**  
**SharedLibrary!}]**)  
[Specify a user-defined table style to replace the current table style]

**TableStyleRetrieve** (  
**Filename:String;**  
**Library:Enumeration {**  
**CurrentDoc!;**  
**PersonalLibrary!;**  
**SharedLibrary!}]**)  
[Retrieve a table style from the specified style library]

**TableStyleSave** (  
**Filename:String;**  
**Library:Enumeration {**  
**CurrentDoc!;**  
**PersonalLibrary!;**  
**SharedLibrary!}]**)  
[Save a table style to the specified style library]

**TableTextBoxCreate** (**Columns:Numeric;Rows:Numeric**)  
[Create a table in a text box]

**TableTextToFormula** ()  
[Convert the contents of the current cell to a formula]

**TableToolsDlg** ()  
[Display a dialog for quickly editing a table]

**TableUndelete** ()  
[Insert the last deleted item in a table]

**TableUpdateAssociatedCharts** (**State:Enumeration {Off!;On!}**)  
[Turn the Update associated charts option on or off]

**TableUseDefaultLinesColor** ()  
[Specify the Line Style line color as the default line color for a table]

**TabRight** (**[Type:Enumeration {Normal!;DotLeader!}]**)  
[Insert a hard right-aligned tab]

**TabRightDot** ()  
[Insert a hard right-aligned tab with dot leader]

**TabSet** (  
**Origin:Enumeration {**  
**Absolute!;**  
**Relative!;**  
**[Position:Numeric Measurement];**  
**[TabType:Enumeration {**  
**TabLeft!;**  
**TabCenter!;**  
**TabRight!;**  
**TabDecimal!;**  
**TabLeftDot!;**  
**TabCenterDot!;**  
**TabRightDot!;**  
**TabDecimalDot!}]**)  
[Specify tab settings]

### TabSetDlg ()

[Set or clear tabs and specify dot leaders]

### TemplateCopyObject (

```
CopyFrom:String;
ObjectType:Enumeration {
    QuickWords!;
    Abbreviation!;
    ButtonBar!;
    Keyboard!;
    Macro!;
    Menu!};
ObjectName:String)
```

[Copy a template object, such as a style or a macro, from one template to another]

### TemplateCopyObjectDlg ()

[Display the Copy/Remove Template Objects dialog box]

### TemplateCreate (Filename:String;[BasedOn:String])

[Create a new template based on an existing template]

### TemplateDescriptionDlg ()

[Create and edit a description of the template]

### TemplateEdit (Filename:String)

[Edit a template]

### TemplateFill ()

[Fill in the template with new information]

### String := TemplateGetAssociation (

```
FeatureTrigger:Enumeration {
    Main!;
    EquationEditor!;
    Tables!;
    BoxSelected!;
    Headers!;
    Footers!;
    Footnote!;
    Endnote!;
    Outline!;
    Watermark!;
    Comment!;
    SGML!;
    PostStartup!;
    PreTables!;
    PostTables!;
    PreNew!;
    PostNew!;
    PreOpen!;
    PostOpen!;
    PreClose!;
    PostClose!;
    PrePrint!;
    PostPrint!;
    PreDocument!;
    PostDocument!};
Object:Enumeration {
    ButtonBar!;
    Keyboard!;
    Macro!;
    Menu!})
```

[Return the association of the feature or trigger, and the object specified for the current template]

### TemplateMacroEditControlBar ([State:Enumeration {Off!;On!}])

[Display the macro toolbar to edit template macros]

### TemplateSelect (Filename:String)

[Select a document template]

### TemplateSetAssociation (

```
FeatureTrigger:Enumeration {
    Main!;
    EquationEditor!;
    Tables!;
    BoxSelected!;
    Headers!;
    Footers!;
    Footnote!;
    Endnote!;
    Outline!;
    Watermark!;
    Comment!;
    SGML!;
    PostStartup!;
    PreTables!;
    PostTables!;
    PreNew!;
    PostNew!;
    PreOpen!;
    PostOpen!;
    PreClose!;
    PostClose!;
    PrePrint!;
    PostPrint!;
    PreDocument!;
    PostDocument!};
Object:Enumeration {
    ButtonBar!;
    Keyboard!;
    Macro!;
    Menu!};
ObjectName:String)
```

[Assign menus, keyboards, and toolbars to selected WordPerfect 10 features or triggers in the current template]

### TextBorderCreate (

```
Style:Any {
    SpacingOnly!;
    SingleBorder!;
    DoubleBorder!;
    DashedBorder!;
    DottedBorder!;
    ThickBorder!;
    ExtraThickBorder!;
    ThinThickBorder!;
    ThickThinBorder!;
    ThickTopBottomBorder!;
    ButtonBorder!;
    ColumnBorderBetween!;
    ColumnBorderAll!;
    HeavySingleBorder!;
    HeavyDoubleBorder!;
    NoBorder!};
FillName:Any {
    Fill10!;
    Fill20!;
    Fill30!;
    Fill40!;
    Fill50!;
    Fill60!;
    Fill70!;
    Fill80!;
    Fill90!;
    Fill100!;
    FillButton!;
    Fill5!;
    NoFill!};
BorderFunction:Enumeration {
    PageBorder!;
    ParagraphBorder!;
```

**ColumnBorder!}}**  
[Create a paragraph, column, or page border]

**TextBorderEdit** (BorderType:Enumeration {  
page!;  
paragraph!;  
column!})  
[Begin editing changes to the current border]

**TextBorderEnd** (State:Enumeration {Cancel!;Save!})  
[End text border creation or editing and save or cancel changes]

**TextBorderFillStyle** (Fillname:Any {  
Fill10!;  
Fill20!;  
Fill30!;  
Fill40!;  
Fill50!;  
Fill60!;  
Fill70!;  
Fill80!;  
Fill90!;  
Fill100!;  
FillButton!;  
Fill5!;  
NoFill!})  
[Specify a fill style for the current border]

**TextBorderStyle** (Style:Any {  
SpacingOnly!;  
SingleBorder!;  
DoubleBorder!;  
DashedBorder!;  
DottedBorder!;  
ThickBorder!;  
ExtraThickBorder!;  
ThinThickBorder!;  
ThickThinBorder!;  
ThickTopBottomBorder!;  
ButtonBorder!;  
ColumnBorderBetween!;  
ColumnBorderAll!;  
HeavySingleBorder!;  
HeavyDoubleBorder!;  
NoBorder!})  
[Specify a border style]

**TextBoxCreate** ()  
[Create a text box]

**TextColor** (  
[ColorName:String];  
[Red:Numeric];  
[Green:Numeric];  
[Blue:Numeric])  
[Change the color of printed characters]

**TextShade** (ShadingValue:Numeric)  
[Specify the intensity of a text color]

**ThirdPartyCreate** (  
Id:Numeric;  
Data:Raw Binary Data)  
[Insert a third party function into the document]

**ThirdPartyDelete** (  
Id:Numeric;  
From:Enumeration {  
BegOfFile!;  
CurrentPos!};  
Direction:Enumeration {  
Forward!;

Backward!})  
[Delete a third party function from the document]

**Numeric := ThirdPartyFind** (  
From:Enumeration {  
BegOfFile!;  
CurrentPos!};  
Direction:Enumeration {  
Forward!;  
Backward!})  
[Find the third party id of the next or previous third party function]

**Raw Binary Data := ThirdPartyFindByID** (  
Id:Numeric;  
From:Enumeration {  
BegOfFile!;  
CurrentPos!};  
Direction:Enumeration {  
Forward!;  
Backward!})  
[Get the data from a third party function]

**String := ThousandsSeparator** ([Character:String])  
[Specify a character to separate every three digits]

**TOACreateFullFormDig** ()  
[Create the full form of a table of authorities citation]

**TOADefineDig** ()  
[Define a table of authorities]

**TOADefinition** (  
SectionName:String;  
[Style:Any {  
NormalStyle!;  
FootnoteNumberInDocumentStyle!;  
EndnoteNumberInDocumentStyle!;  
FigureNumberStyle!;  
TableBoxNumberStyle!;  
TextBoxNumberStyle!;  
UserBoxNumberStyle!;  
EquationNumberStyle!;  
HypertextStyle!;  
ToC1Style!;  
ToC2Style!;  
ToC3Style!;  
ToC4Style!;  
ToC5Style!;  
Index1Style!;  
Index2Style!;  
ListStyle!;  
ToAStyle!;  
DocStyle!;  
BoxTextStyle!;  
CaptionStyle!;  
FootnoteStyle!;  
EndnoteStyle!;  
CommentStyle!;  
HeaderAStyle!;  
HeaderBStyle!;  
FooterAStyle!;  
FooterBStyle!;  
WatermarkAStyle!;  
WatermarkBStyle!;  
ParagraphOutInStyle!;  
OutlineStyle!;  
LegalOutInStyle!;  
BulletsOutInStyle!;  
HeadingsOutInStyle!;  
Legal\_2OutInStyle!;  
NumbersOutInStyle!;  
Level1Style!;  
Level2Style!;



Level3Style!;  
 Level4Style!;  
 Level5Style!;  
 Level6Style!;  
 Level7Style!;  
 Level8Style!;  
 Legal1Style!;  
 Legal2Style!;  
 Legal3Style!;  
 Legal4Style!;  
 Legal5Style!;  
 Legal6Style!;  
 Legal7Style!;  
 Legal8Style!;  
 Heading1Style!;  
 Heading2Style!;  
 Heading3Style!;  
 Heading4Style!;  
 Heading5Style!;  
 Heading6Style!;  
 Heading7Style!;  
 Heading8Style!;  
 MonospacedStyle!;  
 CustomHTMLStyle!;  
 PreformattedStyle!;  
 AddressStyle!;  
 QuotationsOutInStyle!;  
 DefinitionsOutInStyle!;  
 Quotation1Style!;  
 Quotation2Style!;  
 Quotation3Style!;  
 Quotation4Style!;  
 Quotation5Style!;  
 Quotation6Style!;  
 Quotation7Style!;  
 Quotation8Style!;  
 Definition1Style!;  
 Definition2Style!;  
 Definition3Style!;  
 Definition4Style!;  
 Definition5Style!;  
 Definition6Style!;  
 Definition7Style!;  
 Definition8Style!;  
 Level9Style!;  
 Legal9Style!;  
 Heading9Style!;  
 Quotation9Style!;  
 Definition9Style!;

[Numbering:Enumeration {

DefNone!;  
 DefFollows!;  
 DefParentheses!;  
 DefFlushRight!;  
 DefLeader!;

[Underline:Enumeration {

NoToAUnderline!;  
 ToAUnderline!;

[PageCombine:Enumeration {

NoCombine!;  
 Combine!;

[NumberingFormat:String])

[Specify a section, define format options, and insert a table of authorities definition in the document]

**TOADefinitionCombine (**

SectionName:String;

[Combine:Enumeration {

NoCombine!;  
 Combine!;

[Combine sequential page numbers in a table of authorities section]

**TOADefinitionCreate (**

SectionName:String;

[Style:Any {

NormalStyle!;  
 FootnoteNumberInDocumentStyle!;  
 EndnoteNumberInDocumentStyle!;  
 FigureNumberStyle!;  
 TableBoxNumberStyle!;  
 TextBoxNumberStyle!;  
 UserBoxNumberStyle!;  
 EquationNumberStyle!;  
 HypertextStyle!;  
 ToC1Style!;  
 ToC2Style!;  
 ToC3Style!;  
 ToC4Style!;  
 ToC5Style!;  
 Index1Style!;  
 Index2Style!;  
 ListStyle!;  
 ToAStyle!;  
 DocStyle!;  
 BoxTextStyle!;  
 CaptionStyle!;  
 FootnoteStyle!;  
 EndnoteStyle!;  
 CommentStyle!;  
 HeaderAStyle!;  
 HeaderBStyle!;  
 FooterAStyle!;  
 FooterBStyle!;  
 WatermarkAStyle!;  
 WatermarkBStyle!;  
 ParagraphOutInStyle!;  
 OutlineStyle!;  
 LegalOutInStyle!;  
 BulletsOutInStyle!;  
 HeadingsOutInStyle!;  
 Legal\_2OutInStyle!;  
 NumbersOutInStyle!;  
 Level1Style!;  
 Level2Style!;  
 Level3Style!;  
 Level4Style!;  
 Level5Style!;  
 Level6Style!;  
 Level7Style!;  
 Level8Style!;  
 Legal1Style!;  
 Legal2Style!;  
 Legal3Style!;  
 Legal4Style!;  
 Legal5Style!;  
 Legal6Style!;  
 Legal7Style!;  
 Legal8Style!;  
 Heading1Style!;  
 Heading2Style!;  
 Heading3Style!;  
 Heading4Style!;  
 Heading5Style!;  
 Heading6Style!;  
 Heading7Style!;  
 Heading8Style!;  
 MonospacedStyle!;  
 CustomHTMLStyle!;  
 PreformattedStyle!;  
 AddressStyle!;  
 QuotationsOutInStyle!;  
 DefinitionsOutInStyle!;  
 Quotation1Style!;  
 Quotation2Style!;

Quotation3Style!  
 Quotation4Style!  
 Quotation5Style!  
 Quotation6Style!  
 Quotation7Style!  
 Quotation8Style!  
 Definition1Style!  
 Definition2Style!  
 Definition3Style!  
 Definition4Style!  
 Definition5Style!  
 Definition6Style!  
 Definition7Style!  
 Definition8Style!  
 Level9Style!  
 Legal9Style!  
 Heading9Style!  
 Quotation9Style!  
 Definition9Style!});  
 [Numbering:Enumeration {  
   DefNone!  
   DefFollows!  
   DefParentheses!  
   DefFlushRight!  
   DefLeader!});  
 [Underline:Enumeration {  
   NoToAUnderline!  
   ToAUnderline!});  
 [PageCombine:Enumeration {  
   NoCombine!  
   Combine!});  
 [NumberingFormat:String])  
 [Create a table of authorities definition]

**TOADefinitionDelete** (SectionName:String)  
 [Delete a table of authorities section definition]

**TOADefinitionMode** (  
   SectionName:String;  
   [Numbering:Enumeration {  
     DefNone!  
     DefFollows!  
     DefFollowsParenthesis!  
     DefFlushRight!  
     DefLeaders!});  
 [Specify a page numbering mode in a section]

**TOADefinitionRename** (OldSection:String; NewSection:String)  
 [Rename a section]

**TOADefinitionRetrieve** (Filename:String; SectionName:String)  
 [Retrieve a section from another document into the current document]

**TOADefinitionStyle** (  
   SectionName:String;  
   Style:Any {  
     NormalStyle!  
     FootnoteNumberInDocumentStyle!  
     EndnoteNumberInDocumentStyle!  
     FigureNumberStyle!  
     TableBoxNumberStyle!  
     TextBoxNumberStyle!  
     UserBoxNumberStyle!  
     EquationNumberStyle!  
     HypertextStyle!  
     ToC1Style!  
     ToC2Style!  
     ToC3Style!  
     ToC4Style!  
     ToC5Style!  
     Index1Style!  
     Index2Style!  
     ListStyle!  
     ToAStyle!  
     DocStyle!  
     BoxTextStyle!  
     CaptionStyle!  
     FootnoteStyle!  
     EndnoteStyle!  
     CommentStyle!  
     HeaderAStyle!  
     HeaderBStyle!  
     FooterAStyle!  
     FooterBStyle!  
     WatermarkAStyle!  
     WatermarkBStyle!  
     ParagraphOutInStyle!  
     OutlineStyle!  
     LegalOutInStyle!  
     BulletsOutInStyle!  
     HeadingsOutInStyle!  
     Legal\_2OutInStyle!  
     NumbersOutInStyle!  
     Level1Style!  
     Level2Style!  
     Level3Style!  
     Level4Style!  
     Level5Style!  
     Level6Style!  
     Level7Style!  
     Level8Style!  
     Legal1Style!  
     Legal2Style!  
     Legal3Style!  
     Legal4Style!  
     Legal5Style!  
     Legal6Style!  
     Legal7Style!  
     Legal8Style!  
     Heading1Style!  
     Heading2Style!  
     Heading3Style!  
     Heading4Style!  
     Heading5Style!  
     Heading6Style!  
     Heading7Style!  
     Heading8Style!  
     MonospacedStyle!  
     CustomHTMLStyle!  
     PreformattedStyle!  
     AddressStyle!  
     QuotationsOutInStyle!  
     DefinitionsOutInStyle!  
     Quotation1Style!  
     Quotation2Style!  
     Quotation3Style!  
     Quotation4Style!  
     Quotation5Style!  
     Quotation6Style!  
     Quotation7Style!  
     Quotation8Style!  
     Definition1Style!  
     Definition2Style!  
     Definition3Style!  
     Definition4Style!  
     Definition5Style!  
     Definition6Style!  
     Definition7Style!  
     Definition8Style!  
     Level9Style!  
     Legal9Style!  
     Heading9Style!  
     Quotation9Style!

**Definition9Style!)**  
[Specify a table of authorities section style]

**TOADefinitionUnderline (**  
  **Section:String;**  
  **[Action:Enumeration {**  
    **NoToAUnderline!;**  
    **ToAUnderline!}]])**

[Turn Underlining on or off in a table of authorities section]

**TOAEditFullForm (ShortForm:String)**  
[Edit a full form citation]

**TOAEditFullFormCBar ()**  
[Display the Table of Authorities feature bar]

**TOAEditFullFormDlg ()**  
[Edit the full form of a table of authorities citation]

**TOAEditPageFormat (**  
  **Listname:String;**  
  **NumberFormat:Any {**  
    **DocFormat!})**  
[Specify the text for page numbering in a table of authorities]

**TOAEditSection (**  
  **OldName:String;**  
  **NewName:String)**  
[Rename a section name in the full form of a table of authorities]

**TOAEditShortForm (**  
  **OldName:String;**  
  **NewName:String)**  
[Rename a short form name in the full form of a table of authorities]

**TOAMark (**  
  **[SectionName:String];**  
  **[ShortForm:String])**  
[Make a full form citation at the insertion point]

**TOAMarkShortForm (ShortForm:String)**  
[Mark the current short form citation at the insertion point]

**TOCDefine (**  
  **[WrapLastLevel:Enumeration {No!; Yes!}];**  
  **[NumberingFormat:String];**  
  **{{Style:Any {**  
    **NormalStyle!;**  
    **FootnoteNumberInDocumentStyle!;**  
    **EndnoteNumberInDocumentStyle!;**  
    **FigureNumberStyle!;**  
    **TableBoxNumberStyle!;**  
    **TextBoxNumberStyle!;**  
    **UserBoxNumberStyle!;**  
    **EquationNumberStyle!;**  
    **HypertextStyle!;**  
    **ToC1Style!;**  
    **ToC2Style!;**  
    **ToC3Style!;**  
    **ToC4Style!;**  
    **ToC5Style!;**  
    **Index1Style!;**  
    **Index2Style!;**  
    **ListStyle!;**  
    **ToAStyle!;**  
    **DocStyle!;**  
    **BoxTextStyle!;**  
    **CaptionStyle!;**  
    **FootnoteStyle!;**  
    **EndnoteStyle!;**  
    **CommentStyle!;**  
    **HeaderAStyle!;**  
    **HeaderBStyle!;**

**FooterAStyle!;**  
**FooterBStyle!;**  
**WatermarkAStyle!;**  
**WatermarkBStyle!;**  
**ParagraphOutInStyle!;**  
**OutlineStyle!;**  
**LegalOutInStyle!;**  
**BulletsOutInStyle!;**  
**HeadingsOutInStyle!;**  
**Legal\_2OutInStyle!;**  
**NumbersOutInStyle!;**  
**Level1Style!;**  
**Level2Style!;**  
**Level3Style!;**  
**Level4Style!;**  
**Level5Style!;**  
**Level6Style!;**  
**Level7Style!;**  
**Level8Style!;**  
**Legal1Style!;**  
**Legal2Style!;**  
**Legal3Style!;**  
**Legal4Style!;**  
**Legal5Style!;**  
**Legal6Style!;**  
**Legal7Style!;**  
**Legal8Style!;**  
**Heading1Style!;**  
**Heading2Style!;**  
**Heading3Style!;**  
**Heading4Style!;**  
**Heading5Style!;**  
**Heading6Style!;**  
**Heading7Style!;**  
**Heading8Style!;**  
**MonospacedStyle!;**  
**CustomHTMLStyle!;**  
**PreformattedStyle!;**  
**AddressStyle!;**  
**QuotationsOutInStyle!;**  
**DefinitionsOutInStyle!;**  
**Quotation1Style!;**  
**Quotation2Style!;**  
**Quotation3Style!;**  
**Quotation4Style!;**  
**Quotation5Style!;**  
**Quotation6Style!;**  
**Quotation7Style!;**  
**Quotation8Style!;**  
**Definition1Style!;**  
**Definition2Style!;**  
**Definition3Style!;**  
**Definition4Style!;**  
**Definition5Style!;**  
**Definition6Style!;**  
**Definition7Style!;**  
**Definition8Style!;**  
**Level9Style!;**  
**Legal9Style!;**  
**Heading9Style!;**  
**Quotation9Style!;**  
**Definition9Style!}}];**  
  **[Numbering:Enumeration {**  
    **DefNone!;**  
    **DefFollows!;**  
    **DefParentheses!;**  
    **DefFlushRight!;**  
    **DefLeader!}]])**  
[Define a table of contents and insert the definition]

**TOCDefineDlg ()**  
[Define a table of contents]

**TOCMark (Level:Numeric)**

[Mark the selected text for the table of contents]

**ToggleMergeView ()**

[Toggle between the merge form and data view]

**TotalPagesDisplay ()**

[Insert a code in the text to display the total pages]

**TotalPagesMethod ([Method:Enumeration {  
 Numbers!;  
 LowerLetters!;  
 UpperLetters!;  
 LowerRoman!;  
 UpperRoman!}])**

[Specify the page numbering method to use for the total number of pages]

**TwainAcquire ()**

[Scan a document or an image]

**TwainSelectSourceDlg ()**

[Select scanning software]

**Type ([Text:String])**

[Insert text at the insertion point]

**TypeChar (**

**CharacterSet:Numeric;  
 Offset:Numeric)**

[Insert a character from the WordPerfect 10 character sets]

**Typeover (State:Enumeration {Off!;On!})**

[Turn Typeover on or off]

**TypeoverKey ()**

[Turn on Typeover]

**TypesetManualKerningDlg ()**

[Increase or decrease the space between two characters]

- U -**Undelete (Which:Numeric)**

[Restore one of the last three deletions]

**UndeleteDlg ()**

[Restore one of the last three deletions]

**UnderlineColor (**

**[ColorName:String];  
 [Red:Numeric];  
 [Green:Numeric];  
 [Blue:Numeric];  
 [Shade:Numeric])**

[Underline color]

**UnderlineSpaces (State:Enumeration {No!;Yes!})**

[Underline spaces]

**UnderlineStyle (Style:Enumeration {**

**Solid!;  
 Dotted!;  
 Dashed!;  
 DashDot!;  
 DashDotDot!;  
 Wave!;  
 Thick!})**

[Underline style]

**UnderlineTabs (State:Enumeration {No!;Yes!})**

[Underline tabs]

**Undo ([Number:Numeric])**

[Reverse the last change - ^0]

**UndoRedoDlg ()**

[Cancel or restore recent changes]

**UseRegQuotesWithNumbers (State:Enumeration {Off!;On!})**

[Turn Use Regular Quotes with Numbers on or off]

**UserFunction (Action:String; {[Data:Any]})**

[Execute a third-party execution command]

- V -**VariableEditBegin ()**

[Begin editing a variable]

**VariableEditEnd (State:Enumeration {Cancel!;Save!})**

[Save or cancel variable changes]

**VariableCodes (**

**[State:Enumeration {WithoutOffCodes!; WithOffCodes!};  
 [EditType:Numeric])**

[Edit variable codes]

**VariableCreate (**

**Name:String;  
 [CreateFlag:Enumeration {  
 CreateFromScratch!;  
 CreateFromCurrentVar!}])**

[Create a variable]

**VariableDelete (**

**VarName:String;  
 Codes:Enumeration {  
 IncludingCodes!;  
 LeavingCodes!};  
 [Library:Enumeration {  
 CurrentDoc!;  
 PersonalLibrary!;  
 SharedLibrary!}])**

[Delete a specified variable]

**VariableRename (Name:String)**

[Rename a variable]

**VariableRetrieve (**

**File:String;  
 [Library:Enumeration {  
 CurrentDoc!;  
 PersonalLibrary!;  
 SharedLibrary!}])**

[Retrieve variables from a style file]

**VariableSave (**

**Filename:String;  
 [Library:Enumeration {  
 CurrentDoc!;  
 PersonalLibrary!;  
 SharedLibrary!}])**

[Save variables from a library to a style file]

**VariablesListDlg ()**

[Insert variable into your document]

**ViewDraft ()**

[Display document without headers, footnotes, watermarks, etc.]

**ViewOutline ()**

[Create or edit an outline, or change an outline definition]

**ViewPage ()**

[Display document as it would look printed, including headers, footnotes, watermarks, etc.]

**ViewToggle ()**

[No note was included in the dump]

**ViewToolbarsDlg ()**

[Select the bars to display or hide]

**ViewTwoPage ()**

[Display two pages]

**VLineCreate ()**

[Put a line in your document from the top margin to the bottom margin]

**VolumeNumber (Volume:Numeric)**

[Specify the current volume number]

**VolumeNumberDecrement ()**

[Decrease the volume number by one]

**VolumeNumberDisplay ()**

[Insert the current volume number at the insertion point]

**VolumeNumberIncrement ()**

[Increase the volume number by one]

**VolumeNumberMethod (NumberingMethod:Enumeration {  
Numbers!;  
LowerLetters!;  
UpperLetters!;  
LowerRoman!;  
UpperRoman!})**  
[Specify a numbering type to display volume numbers]

- W -

**WatermarkA (**

**Action:Enumeration {  
Off!;  
Create!;  
Edit!};**  
**[Occurrence:Enumeration {  
NoPages!;  
OddPages!;  
EvenPages!;  
AllPages!}];**  
**[State:Enumeration {Overlay!; NoOverlay!}])**  
[Create, edit, or turn off Watermark A on the specified pages]

**WatermarkB (**

**Action:Enumeration {  
Off!;  
Create!;  
Edit!};**  
**[Occurrence:Enumeration {  
NoPages!;  
OddPages!;  
EvenPages!;  
AllPages!}];**  
**[State:Enumeration {Overlay!; NoOverlay!}])**  
[Create, edit, or turn off Watermark B on the specified pages]

**WatermarkDlg ()**

[Create a watermark]

**WatermarkNext ()**

[Go to the next watermark]

**WatermarkPlacementDlg ()**

[Select whether a watermark will appear on odd-numbered pages, even-numbered pages, or on every page]

**WatermarkPrev ()**

[Go to the previous watermark]

**WatermarkShadingDlg ()**

[Change the watermark text or the image shading]

**WebActiveLinkColor (**

**Red:Numeric;  
Green:Numeric;  
Blue:Numeric)**  
[Obsolete Command]

**WebBackgroundColor (**

**Red:Numeric;  
Green:Numeric;  
Blue:Numeric)**  
[Obsolete Command]

**WebBackgroundImage ([Filename:String])**

[Obsolete Command]

**WebBaseURL ([URL:String])**

[Obsolete Command]

**WebBoxHTMLProperties (Properties:String)**

[Specify the HTML properties of the current box]

**WebBoxHTMLPropertiesDlg ()**

[Edit the HTML properties of the current box]

**WebColorDlg ()**

[Obsolete Command]

**WebColumnDefineDlg ()**

[No note was included in the dump]

**WebCustomHTMLToggle ()**

[No note was included in the dump]

**WebFormsFormPropertiesDlg ()**

[Obsolete Command]

**WebFormsHiddenDlg ()**

[Obsolete Command]

**WebFormsListPropertiesDlg ()**

[Obsolete Command]

**WebFormsProperties ()**

[No note was included in the dump]

**WebFormsRadioPropertiesDlg ()**

[Obsolete Command]

**WebFormsTextAreaPropertiesDlg ()**

[Obsolete Command]

**WebFormsTextPropertiesDlg ()**

[Obsolete Command]

**WebInsertCheckBox ()**

[No note was included in the dump]

**WebInsertForm ()**

[No note was included in the dump]

**WebInsertHiddenField ()**

[No note was included in the dump]

**WebInsertPassword ()**  
[No note was included in the dump]

**WebInsertRadioButton ()**  
[No note was included in the dump]

**WebInsertResetButton ()**  
[No note was included in the dump]

**WebInsertSelectList ()**  
[No note was included in the dump]

**WebInsertSubmitButton ()**  
[No note was included in the dump]

**WebInsertSubmitImage ()**  
[No note was included in the dump]

**WebInsertTextArea ()**  
[No note was included in the dump]

**WebInsertTextLine ()**  
[No note was included in the dump]

**WebJavaCreateApplet ()**  
[No note was included in the dump]

**WebJavaModifyApplet ()**  
[No note was included in the dump]

**WebJavaRunApplets ()**  
[No note was included in the dump]

**WebLaunchWebBrowser ([Location:String])**  
[Launch your internet browser]

**WebLinkColor (**  
    **Red:Numeric;**  
    **Green:Numeric;**  
    **Blue:Numeric)**  
[Obsolete Command]

**WebMetaTags (Meta:String)**  
[Obsolete Command]

**WebMonoToggle ()**  
[No note was included in the dump]

**WebNewDlg ()**  
[Use the WebExpert or a blank web document template to create a new web document]

**WebPublish (**  
    **Filename:String;**  
    **Server:String;**  
    **SeperateFrames:Enumeration {Off!;On!})**  
[Save the current document in HTML format for publication on the Internet]

**WebPublishDlg ()**  
[Save a copy of the current document in HTML format]

**WebTableProperties (**  
    **[TableBordWidth:Numeric];**  
    **[CellBordWidth:Numeric];**  
    **[CellMarginWidth:Numeric];**  
    **[TableWidthType:Enumeration {**  
        **NotSet!;**  
        **WPU!;**  
        **Pixels!;**  
        **Percent!});**  
    **[TableWidth:Numeric];**  
    **[TableHeightType:Enumeration {**

**NotSet!;**  
    **WPU!;**  
    **Pixels!;**  
    **Percent!});**  
    **[TableHeight:Numeric];**  
    **[TablePosition:Enumeration {**  
        **Top!;**  
        **Center!;**  
        **Bottom!});**  
[Edit the properties of a table in HTML]

**WebTextColor (**  
    **Red:Numeric;**  
    **Green:Numeric;**  
    **Blue:Numeric)**  
[Obsolete Command]

**WebTitle (Title:Any {**  
    **Auto!})**  
[Obsolete Command]

**WebTitleDlg ()**  
[No note was included in the dump]

**WebViewAsHTML ()**  
[No note was included in the dump]

**WebViewInWebBrowser ()**  
[Create a temporary copy of the current document in HTML and display it in a browser]

**WebVisitedLinkColor (**  
    **Red:Numeric;**  
    **Green:Numeric;**  
    **Blue:Numeric)**  
[Obsolete Command]

**WebWallpaper ()**  
[No note was included in the dump]

**WidowOrphan (State:Enumeration {Off!;On!})**  
[Turn Widow/Orphan on or off]

**WindowCascade ()**  
[Arrange windows so they overlap]

**WindowTile ()**  
[Arrange windows one above the other without overlapping]

**WindowTileVertical ()**  
[Arrange windows next to each other without overlapping]

**WordHelpTopics ()**  
[Display a list of Word Help topics]

**WordLetterspaceDlg ()**  
[Adjust word/letter spacing, line height, etc.]

**WordLetterSpacing (**  
    **CharacterWidth:Enumeration {**  
        **Normal!;**  
        **Optimal!;**  
        **Maximum!};**  
    **SpaceWidth:Enumeration {**  
        **Normal!;**  
        **Optimal!;**  
        **Maximum!})**  
[Specify the display spacing between words and letters]

**WordSpacingJustificationLimits (**  
    **Compression:Numeric;**  
    **Expansion:Numeric)**  
[Adjust the spacing between words when Full justification is on]

**WP51CursorMovement (State:Enumeration {Off!;On!})**  
[Turn WordPerfect 5.1 for DOS cursor movement on or off]

**WPActivate ()**  
[Activate WordPerfect 10]

**WPCharactersDlg ()**  
[Insert WordPerfect Symbols]

**WPDraw ()**  
[Create a picture]

**WPGrammatik ()**  
[Check and correct grammar and spelling]

**WPSpeller ()**  
[Check and correct spelling]

**WPThesaurus ()**  
[Get synonyms, antonyms, definitions, and usage examples for a word]

**WPDictionary ()**  
[Look up words in the dictionary]

**WritingToolsDisable ()**  
[Toggle grammar and spell checking on or off in the current document]

- Z -

**ZoomDlg ()**  
[Change the magnification of the active window]

**ZoomIn ()**  
[Increase the magnification of the active window]

**ZoomOut ()**  
[Decrease the magnification of the active window]

**ZoomToFullPage ()**  
[Display the entire page]

**ZoomToMarginWidth ()**  
[Display the document at the current margin width]

**ZoomToPageWidth ()**  
[Display the document at the current page width]

## WORDPERFECT PERFECTSCRIPT COMMANDS FOR WORDPERFECT 10

Misc \* / - + <  
> % = etc

A B C D E F G I L M N O P Q R S T U V W X

- ~ \* / % + - ^ = < > != & ; : . . . -

**Numeric := ~ Numeric**

[Returns the bitwise complement of a number; Precedence level 1]

**Any := Any \*\* Any**

[Returns the left operand raised to the power of the right operand; Precedence level 2]

**Any := Any \* Any**

[Returns the product of two numbers; Precedence level 3]

**Numeric := Any / Any**

[Returns the floating point quotient of two numbers; Precedence level 3]

**Numeric := Any % Any**

[Returns the floating point remainder of the quotient of two numbers; Precedence level 3]

**Any := Any + Any**

[Returns the sum of two numbers; Precedence level 4]

**String := String + String**

[Returns the concatenation of two strings; Precedence level 4]

**Any := Any - Any**

[Returns the difference between 2 numbers; Precedence level 4]

**String := String - String**

[Returns the left string with the first occurrence of the right string removed; Precedence level 4]

**Numeric := Numeric << Numeric**

[Returns the left operand shifted left by the right operand; Precedence level 5]

**Numeric := Numeric >> Numeric**

[Returns the left operand shifted right by the right operand; Precedence level 5]

**Numeric := Numeric <<< Numeric**

[Returns the left operand rotated left by the right operand; Precedence level 5]

**Numeric := Numeric >>> Numeric**

[Returns the left operand rotated right by the right operand; Precedence level 5]

**Boolean := Any = Any**

[Returns TRUE if left is equal to right, otherwise FALSE; Precedence level 6]

**Boolean := Any <> Any**

[Returns TRUE if left is not equal to right, otherwise FALSE; Precedence level 6]

**Boolean := Any != Any**

[Returns TRUE if left is not equal to right, otherwise FALSE; Precedence level 6]

**Boolean := Any < Any**

[Returns TRUE if left is less than right, otherwise FALSE; Precedence level 6]

**Boolean := Any <= Any**

[Returns TRUE if left is less than or equal to right, otherwise FALSE; Precedence level 6]

**Boolean := Any > Any**

[Returns TRUE if left is greater than right, otherwise FALSE; Precedence level 6]

**Boolean := Any >= Any**

[Returns TRUE if left is greater than or equal to right, otherwise FALSE; Precedence level 6]

**Numeric := Numeric & Numeric**

[Returns bitwise AND of 2 numbers; Precedence level 7]

**Variable := & Variable**

[Use only in DLLCall or OLE Automation call in-line parameter function; pass address of variable]

**Numeric := Numeric | Numeric**

[Returns bitwise OR of 2 numbers; Precedence level 7]

**Numeric := Numeric ^ Numeric**

[Returns bitwise XOR of 2 numbers; Precedence level 7]

**Variable := Any**

[Assigns a value to a variable; Precedence level 10]

**Any := Variable = Any**

[Assigns a value to a variable; Precedence level 10]

;

[Parameter separator in a call]

**Variable . Command Name**

[Separates a command prefix and command name]

**.. Command Name**

[Specifies that the default application/object is to be used]

**Array := { } Any**

[Specifies an array literal value]

{ } Any

[Specifies a repeating group parameter]

( ) Any

[Specifies a parameter list]

**Any := ( ) Any**

[Specifies an expression]

**Numeric := ' String**

[Specifies a character value]

!

[Specifies that the previous identifier is an enumeration]

@

[Specifies that the previous identifier is a label]

:

[Mark a spot in the macro with the specified label name]

:

[Terminate a parameter name in a command call]

**Numeric := b**

[Specifies that the previous numeric value is a binary numeric value]

**Numeric := o**

[Specifies that the previous numeric value is an octal numeric value]



**Numeric := x**

[Specifies that the previous numeric value is a hexadecimal numeric value]

**Numeric := h**

[Specifies that the previous numeric value is a hexadecimal numeric value]

**Numeric Measurement := w**

[Specifies that the previous numeric value is a WP units measurement value]

**Numeric Measurement := c**

[Specifies that the previous numeric value is a centimeters measurement value]

**Numeric Measurement := m**

[Specifies that the previous numeric value is a millimeters measurement value]

**Numeric Measurement := i**

[Specifies that the previous numeric value is an inches measurement value]

**Numeric Measurement := p**

[Specifies that the previous numeric value is a points measurement value]

// ([<CommentText>Any])

[Start a comment]

/\* \*/([<CommentText>Any])

[Block comments]

- A -

**Array := Any .. Any**

[Specifies the range of elements in an array slice]

**Any := Array [ ] Numeric**

[Returns the specified element of an array]

**Numeric := [0]<Array>Array**

[Returns the number of elements in an array variable]

**Numeric := AbsVal (Value:Numeric)**

[Return the absolute value of a value]

**Numeric := acos (Value:Numeric)**

[Return the angle in radians for a specified cosine (inverse (arc) cosine)]

**Numeric := acosh (Value:Numeric)**

[Return the angle in radians for a specified hyperbolic cosine (inverse (arc) hyperbolic cosine)]

**Variable := Address (<VariableName>Variable)**

[Use only in DLLCall or OLE Automation call in-line parameter function; pass address of variable]

**Numeric := Amortize (**

**[Principle:Any {**  
**Compute!};**

**[InterestRate:Any {**  
**Compute!};**

**[Duration:Any {**  
**Compute!};**

**[Payment:Any {**  
**Compute!};**

[Return Principle, Monthly payment, Duration in months or Monthly interest rate of a loan, from the other 3 values]

**Boolean := Boolean AND Boolean**

[Returns BOOLEAN AND of two BOOLEANS; Precedence level 8]

**Animate (**

**Type:Enumeration {**

**Mouse!;**

**Keyboard!};**

**Specification:String)**

[Start animation of mouse or keyboard]

**String := AnsiString (<Value>String)**

[Use only in DLLCall or OLE Automation call in-line parameter function; pass value as ANSI string]

**Numeric := AppActivate (**

**Window:String;**

**[State:Enumeration {**

**Hide!;**

**Normal!;**

**Maximize!;**

**Minimize!;**

**MinimizeNoActivate!;**

**MinimizeActivateTopLevel!;**

**CurrentState!;**

**CurrentStateNoActivate!;**

**RecentStateNoActivate!;**

**Restore!;**

**Default!};**

[Activate a running application by title or window handle]

**Boolean := AppClose (Window:String)**

[Close a running application by title or window handle]

**Enumeration {**

**Success!;**

**AccessDenied!;**

**BadFileFormat!;**

**DDEBusy!;**

**DDEFailed!;**

**DDETimeout!;**

**DLLNotFound!;**

**FileNotFound!;**

**InvalidFileAssociation!;**

**NoFileAssociation!;**

**OutOfMemory!;**

**OutOfResources!;**

**PathNotFound!;**

**SharingError!}**

**:= AppExecute (**

**CommandLine:String;**

**[State:Enumeration {**

**Hide!;**

**Normal!;**

**Maximize!;**

**Minimize!;**

**MinimizeNoActivate!;**

**MinimizeActivateTopLevel!;**

**CurrentState!;**

**CurrentStateNoActivate!;**

**RecentStateNoActivate!;**

**Restore!;**

**Default!};**

[Execute an application]

**Application (**

**<ProductPrefix>String;**

**<ApplicationName>String;**

**[<Default>Enumeration {**

**Default!};**

**[<Language>String])**

[Specify an application that is used in the macro]

**Numeric := AppLocate (WindowTitle:String)**

[Return handle of application window]

**Numeric := AppShow** (  
 Window:String;  
 [State:Enumeration {  
 Hide!;  
 Normal!;  
 Maximize!;  
 Minimize!;  
 MinimizeNoActivate!;  
 MinimizeActivateTopLevel!;  
 CurrentState!;  
 CurrentStateNoActivate!;  
 RecentStateNoActivate!;  
 Restore!;  
 Default!}])

[Change display characteristics of a running application]

**Numeric := asin** (Value:Numeric)

[Return the angle in radians for a specified sine (inverse (arc) sine)]

**Numeric := asinh** (Value:Numeric)

[Return the angle in radians for a specified hyperbolic sine (inverse (arc) hyperbolic sine)]

**Assert** (Condition:Enumeration {

CancelCondition!;  
 ErrorCondition!;  
 NotFoundCondition!;  
 ExitCondition!;  
 VarErrChkCondition!;  
 UserDefinedCondition!)

[Assert a condition]

**Numeric := atan** (Value:Numeric)

[Return the angle in radians for a specified tangent (inverse (arc) tangent)]

**Numeric := atan2** (

Value:Numeric;  
 Value2:Numeric)

[Return the angle in radians for a tangent that is Value / Value2 (inverse (arc) tangent)]

**Numeric := atanh** (Value:Numeric)

[Return the angle in radians for a specified hyperbolic tangent (inverse (arc) hyperbolic tangent)]

**Numeric := Average** ({Value:Numeric})

[Return the average value of a list of values]

- B -

**Beep** ([BeepType:Enumeration {

Default!;  
 Question!;  
 Asterisk!;  
 Exclamation!;  
 CriticalStop!;  
 Standard!}])

[Cause the computer to produce a beep]

**Raw Binary Data := BinaryPack** (

Value:Any;  
 [TotalSize:Enumeration {  
 Auto!}];  
 {[Offset:Enumeration {  
 Auto!}];  
 [Type:Enumeration {  
 Auto!;  
 Detect!;  
 Byte!;  
 Word!;  
 DWord!;  
 Real!;  
 Integer!;

Boolean!;  
 WPString!;  
 AnsiString!;  
 HexString!}];  
 [Size:Enumeration {  
 Auto!}])

[Pack data into a raw binary data value]

**Any := BinaryUnPack** (

Value:Raw Binary Data;  
 {Offset:Enumeration {  
 Auto!};  
 Type:Enumeration {  
 Byte!;  
 Word!;  
 DWord!;  
 Real!;  
 Integer!;  
 Boolean!;  
 WPString!;  
 AnsiString!;  
 HexString!};  
 Size:Enumeration {  
 Auto!}])

[UnPack data from a raw binary data value]

**Boolean := Bool** (<Value>Boolean)

[Use only in OLE Automation call in-line parameter function; pass value as BOOL]

**Boolean := Boolean** (<Value>Boolean)

[Use only in OLE Automation call in-line parameter function; pass value as BOOL]

**Break**

[Jump to the code following the current compound statement]

**Numeric := Byte** (<Value>Numeric)

[Use only in OLE Automation call in-line parameter function; pass value as unsigned BYTE]

- C -

**Call** (<Label>Label)

[Call macro code associated with a label]

**Call** <Label>Label ({[<Parameter>Any]})

[Call macro code associated with a label]

**CallbackResume** ()

[Resume execution after callback]

**CallbackWait** ()

[Pause execution for callback]

**Enumeration {Off!;On!} := Cancel** ([State:Enumeration {Off!;On!}])

[Ignore or accept cancel conditions]

**Case** (

<Test>Any;  
 {<Case>Any;  
 <Label>Label};  
 [<DefaultLabel>Label])

[Jump to a label depending on a value]

**Case Call** (

<Test>Any;  
 {<Case>Any;  
 <Label>Label};  
 [<DefaultLabel>Label])

[Call a label depending on a value]

**CaseOf: (<<Selector>Any)**

[Specify the code to be executed for a particular value in a SWITCH statement]

**Numeric := Ceiling (Value:Numeric)**

[Return the ceiling of a number (smallest integer >= Value)]

**Chain (**

**MacroFilename:String;**  
**{[Parameter:Any]}**)

[Run a macro on current macro termination]

**Numeric := CharLen (String:String)**

[Return number of characters in a string]

**Numeric := CharPos (**

**String:String;**  
**SubString:String;**  
**[Beginning:Numeric]**)

[Return the beginning character position where a substring begins in a string]

**Numeric := CheckSum (**

**Data:Any;**  
**[Options:Enumeration {**  
**Checksum16!;**  
**Checksum32!;**  
**CRC\_16!;**  
**CCITT!;**  
**AUTODIN\_III!});**  
**[Seed:Numeric]**)

[Compute a checksum value]

**Boolean := CloseFile ([FileID:Numeric])**

[Close a file or all files]

**Enumeration {Off!;On!} := Condition (**

**Condition:Enumeration {**  
**CancelCondition!;**  
**ErrorCondition!;**  
**NotFoundCondition!;**  
**ExitCondition!;**  
**VarErrChkCondition!;**  
**UserDefinedCondition!};**  
**[State:Enumeration {**  
**Off!;**  
**On!});**)

[Ignore or accept specific condition]

**Constant (**

**<Name>Variable;**  
**<Value>Any)**

[Define a name to have a compile time constant value]

**Numeric := Constants (Constant:Enumeration {**

**e!;**  
**pi!;**  
**EulersConstant!;**  
**GoldenRatio!})**

[Return various mathematical and physical constants]

**Continue**

[Execute the code in the next CASEOF without evaluating its expression]

**Any := ConvertType (**

**Value:Any;**  
**Type:Enumeration {**  
**Boolean!;**  
**String!;**  
**WPString!;**  
**AnsiString!;**  
**OemString!;**

**Number!;**

**Float!;**

**Integer!;**

**DateTime!;**

**Measurement!;**

**Centimeters!;**

**Inches!;**

**Millimeters!;**

**Points!;**

**WPUnts!;**

**RawBinary!};**

**[DefaultUnitsType:Enumeration {**

**None!;**

**DefaultUnits!;**

**Centimeters!;**

**Inches!;**

**Millimeters!;**

**Points!;**

**WPUnts!});**

[Convert the value to the data type specified]

**Boolean := CopyFile (**

**SourceFilename:String;**  
**DestinationFilename:String;**  
**[Prompts:Enumeration {**  
**NoPrompts!;**  
**Prompts!});**)

[Copy a file]

**Numeric := cos (Angle:Numeric)**

[Return the cosine of an angle in radians]

**Numeric := cosh (Angle:Numeric)**

[Return the hyperbolic cosine of an angle in radians]

**Boolean := CreateDirectory (**

**DirectoryName:String;**  
**[Prompts:Enumeration {**  
**NoPrompts!;**  
**Prompts!});**)

[Create a directory (folder)]

**Object := CreateObject (ClassName:String)**

[Create a new instance of the specified OLE Automation object]

**Numeric := CToN (Character:String)**

[Convert a character to its numeric equivalent]

**Numeric := CubeRoot (Value:Numeric)**

[Return the cube root of a value]

**Numeric := Currency (<Value>Numeric)**

[Use only in OLE Automation call in-line parameter function; pass value as CURRENCY]

**- D -**

**Numeric := Date (<Value>Numeric)**

[Use only in OLE Automation call in-line parameter function; pass value as DATE]

**Date/Time := DateAddDays (**

**[Date:Date/Time];**  
**Days:Numeric)**

[Add a number of days to a date]

**Date/Time := DateAddMonths (**

**[Date:Date/Time];**  
**Months:Numeric;**  
**[EndOfMonth:Enumeration {**  
**SameDay!;**  
**EndOfMonth!});**)

[Add a number of months to a date]

**Date/Time := DateAddWeeks** (  
 [Date:Date/Time];  
 Weeks:Numeric;  
 [FirstDayOfWeek:Enumeration {  
 Current!;  
 Sunday!;  
 Monday!;  
 Tuesday!;  
 Wednesday!;  
 Thursday!;  
 Friday!;  
 Saturday!}])

[Add a number of weeks to a date]

**Date/Time := DateAddYears** (  
 [Date:Date/Time];  
 Years:Numeric)

[Add a number of years to a date]

**Any {Error!} := DateAndTime** (  
 [Day:Numeric];  
 [Month:Enumeration {

January!;  
 February!;  
 March!;  
 April!;  
 May!;  
 June!;  
 July!;  
 August!;  
 September!;  
 October!;  
 November!;  
 December!});

[Year:Numeric];  
 [Hour:Numeric];  
 [Minutes:Numeric];  
 [Seconds:Numeric];  
 [Hundredths:Numeric])

[A value representing the date and time]

**Numeric := DateDay** ([Date:Date/Time])

[A value representing the day of the month]

**Numeric := DateDayOfYear** ([Date:Date/Time])

[A value representing the number of the day in the year (starting with 1)]

**Numeric := DateDaysInMonth** (  
 [Month:Enumeration {

January!;  
 February!;  
 March!;  
 April!;  
 May!;  
 June!;  
 July!;  
 August!;  
 September!;  
 October!;  
 November!;  
 December!});

[Year:Numeric])

[The number of days in the specified month]

**Numeric := DateDaysInYear** ([Year:Numeric])

[The number of days in the specified year]

**Boolean := DateIsLeapYear** ([Year:Numeric])

[Determine if the specified year is a leap year]

**Enumeration {**  
 Error!;

January!;  
 February!;  
 March!;  
 April!;  
 May!;  
 June!;  
 July!;  
 August!;  
 September!;  
 October!;  
 November!;  
 December!)

**:= DateMonth** ([Date:Date/Time])

[A value representing the month of the year]

**String := DateMonthName** (  
 [Date:Date/Time];

[Type:Enumeration {  
 Short!;  
 Long!}])

[The name of the month]

**Date/Time := DateOfMonthEnd** (  
 [Month:Enumeration {

January!;  
 February!;  
 March!;  
 April!;  
 May!;  
 June!;  
 July!;  
 August!;  
 September!;  
 October!;  
 November!;  
 December!});

[Year:Numeric])

[A value representing the end of the specified month]

**Date/Time := DateOfNthDay** (  
 Nth:Numeric;

[Year:Numeric])

[A value representing the date of the Nth day of a year]

**Date/Time := DateOfNthWeek** (  
 Nth:Numeric;

[Year:Numeric];  
 [FirstDayOfWeek:Enumeration {  
 Sunday!;  
 Monday!;  
 Tuesday!;  
 Wednesday!;  
 Thursday!;  
 Friday!;  
 Saturday!}])

[A value representing the start date of the Nth week of a year]

**Date/Time := DateOfNthWeekday** (  
 Nth:Enumeration {

First!;  
 Last!};  
 DayOfWeek:Enumeration {  
 Sunday!;  
 Monday!;  
 Tuesday!;  
 Wednesday!;  
 Thursday!;  
 Friday!;  
 Saturday!};

[Month:Enumeration {  
 January!;  
 February!;

**March!**;  
**April!**;  
**May!**;  
**June!**;  
**July!**;  
**August!**;  
**September!**;  
**October!**;  
**November!**;  
**December!}}**;  
**[Year:Numeric]**  
 [A value representing the date of the Nth weekday of a month]

**Date/Time := DatePart ([Date:Date/Time])**  
 [Return the date part of a date and time value]

**String := DateString (**  
**[Date:Date/Time];**  
**[Type:Enumeration {**  
**Short!**;  
**Long!});**  
**[Format:String])**  
 [A string representing the date]

**Enumeration {**  
**Error!**;  
**Sunday!**;  
**Monday!**;  
**Tuesday!**;  
**Wednesday!**;  
**Thursday!**;  
**Friday!**;  
**Saturday!}**  
**:= DateWeekday ([Date:Date/Time])**  
 [A value representing the day of the week]

**String := DateWeekdayName (**  
**[Date:Date/Time];**  
**[Type:Enumeration {**  
**Short!**;  
**Long!});**  
 [The name of the day of the week]

**Numeric := DateWeekOfYear (**  
**[Date:Date/Time];**  
**[FirstDayOfWeek:Enumeration {**  
**Sunday!**;  
**Monday!**;  
**Tuesday!**;  
**Wednesday!**;  
**Thursday!**;  
**Friday!**;  
**Saturday!});**  
 [A value representing the number of the week in the year (starting with 1)]

**Numeric := DateYear ([Date:Date/Time])**  
 [A value representing the year]

**Any {Error!} := DDEExecute (**  
**ConversationID:Numeric;**  
**Command:String;**  
**[Timeout:Enumeration {**  
**Asynchronous!**;  
**NoTimeout!**;  
**Standard!});**  
**[Label:Label])**  
 [Execute a DDE command]

**Any {Error!} := DDEInitiate (**  
**ServiceName:String;**  
**TopicName:String)**  
 [Initiate a DDE conversation and return conversation handle]

**DDEPoke (**  
**ConversationID:Numeric;**  
**ItemName:String;**  
**ItemData:String;**  
**[Timeout:Enumeration {**  
**NoTimeout!**;  
**Standard!});**  
 [Assign a value to an item in a DDE application]

**String := DDERequest (**  
**ConversationID:Numeric;**  
**ItemName:String;**  
**[Timeout:Enumeration {**  
**NoTimeout!**;  
**Standard!});**  
 [Return the value of an item from a DDE application]

**DDETerminate ([ConversationID:Numeric])**  
 [Terminate a DDE conversation]

**DDETerminateAll ()**  
 [Terminate all DDE conversations]

**Declare (**  
**{<VariableName>Variable;**  
**<Value>Any})**  
 [Declare a local variable or array]

**Default:**  
 [Specify the code to be executed if all CASEOF statements in a SWITCH statement are false]

**Enumeration {**  
**None!**;  
**Centimeters!**;  
**Inches!**;  
**Millimeters!**;  
**Points!**;  
**WPUnits!}**  
**:= DefaultUnits ([Units:Enumeration {**  
**None!**;  
**Centimeters!**;  
**Inches!**;  
**Millimeters!**;  
**Points!**;  
**WPUnits!**;  
**WP1200ths!**;  
**NoChange!});**  
 [Specify the default measurement unit]

**Numeric := DegreesToRadians (Degrees:Numeric)**  
 [Convert an angle in degrees to radians]

**Boolean := DeleteDirectory (**  
**DirectoryName:String;**  
**[Prompts:Enumeration {**  
**NoPrompts!**;  
**Prompts!});**  
 [Delete a directory (folder)]

**Boolean := DeleteFile (**  
**Filename:String;**  
**[Prompts:Enumeration {**  
**NoPrompts!**;  
**Prompts!});**  
 [Delete a file]

**String := DialogAddBitmap (**  
**Dialog:String;**  
**Control:String;**  
**[Left:Numeric];**  
**[Top:Numeric];**  
**[Width:Numeric];**

```

[Height:Numeric];
[Style:Enumeration {
  NoResize!;
  SizeCtlToBmp!;
  SizeBmpToCtl!;
  Transparent!;
  Opaque!});
Filename:String;
[Name:String];
[TransparentColor:Enumeration {
  White!;
  Black!;
  Red!;
  Green!;
  Blue!;
  Cyan!;
  Yellow!;
  Magenta!;
  Orange!;
  Purple!;
  Gray25!;
  Gray50!;
  Gray75!});
[Add a Bitmap to the dialog]

```

```

String := DialogAddCheckBox (
  Dialog:String;
  Control:String;
  [Left:Numeric];
  [Top:Numeric];
  [Width:Numeric];
  [Height:Numeric];
  Text:String;
  MacroVar:Variable;
  [Style:Enumeration {
    CheckboxAuto!;
    Checkbox!;
    Checkbox3State!;
    CheckboxAuto3State!;
    TextOnLeft!;
    TextOnRight!});
[Add a CheckBox control to the dialog]

```

```

String := DialogAddColorWheel (
  Dialog:String;
  Control:String;
  [Left:Numeric];
  [Top:Numeric];
  [Width:Numeric];
  [Height:Numeric];
  MacroVar:Variable)
[Add a ColorWheel control to the dialog]

```

```

String := DialogAddComboBox (
  Dialog:String;
  Control:String;
  [Left:Numeric];
  [Top:Numeric];
  [Width:Numeric];
  [Height:Numeric];
  [Style:Enumeration {
    Simple!;
    Dropdown!;
    Droplist!;
    Unsorted!;
    Sorted!;
    WPChars!;
    AutoHScroll!;
    OEMConvert!});
  MacroVar:Variable;
  [LimitText:Numeric])
[Add a ComboBox control to the dialog]

```

```

String := DialogAddControl (
  Dialog:String;
  Control:String;
  [Left:Numeric];
  [Top:Numeric];
  [Width:Numeric];
  [Height:Numeric];
  Class:String;
  [Style:Numeric];
  WindowName:String;
  [MacroVar:Variable];
  [Instance:Numeric])
[Add a control to the dialog]

```

```

String := DialogAddCounter (
  Dialog:String;
  Control:String;
  [Left:Numeric];
  [Top:Numeric];
  [Width:Numeric];
  [Height:Numeric];
  [Format:Enumeration {
    DisplayNormal!;
    DisplayWPU!;
    DisplayPoints!;
    DisplayCentimeters!;
    DisplayFixedPoint!;
    DisplayPercent!;
    DisplayInches!;
    DisplayInches2!;
    DisplayMillimeters!;
    Display_Inches!;
    AutoHScroll!;
    NoScroll!;
    AutoValidate!});
  MacroVar:Variable;
  [CountMin:Any];
  [CountMax:Any];
  [CountStep:Any])
[Add a Counter control to the dialog]

```

```

String := DialogAddDate (
  Dialog:String;
  Control:String;
  [Left:Numeric];
  [Top:Numeric];
  [Width:Numeric];
  [Height:Numeric];
  [Style:Enumeration {
    NoValidate!;
    Validate!});
  MacroVar:Variable;
  [Year:Numeric];
  [Month:Enumeration {
    January!;
    February!;
    March!;
    April!;
    May!;
    June!;
    July!;
    August!;
    September!;
    October!;
    November!;
    December!});
  [Day:Numeric])
[Add a Date to the dialog]

```

```

String := DialogAddEditBox (
  Dialog:String;
  Control:String;

```

```

[Left:Numeric];
[Top:Numeric];
[Width:Numeric];
[Height:Numeric];
[Style:Enumeration {
  Left!;
  Right!;
  Center!;
  VScroll!;
  HScroll!;
  WPChars!;
  Multiline!;
  Uppercase!;
  Lowercase!;
  Password!;
  WordWrap!;
  SoftRet!;
  Attributes!;
  NoTabs!;
  NoWPChar!;
  AutoVScroll!;
  AutoHScroll!;
  OEMConvert!;
  ReadOnly!;
  WantReturn!});
MacroVar:Variable;
[LimitText:Numeric]
[Add an EditBox control to the dialog]

```

```

String := DialogAddFileNameBox (
  Dialog:String;
  Control:String;
  [Left:Numeric];
  [Top:Numeric];
  [Width:Numeric];
  [Height:Numeric];
  [Style:Enumeration {
    FilesAndDirs!;
    DirOnly!;
    FileDoesntHaveToExist!});
  MacroVar:Variable;
  [DefaultDir:String];
  [Template:String])
[Add a FileName control to the dialog]

```

```

String := DialogAddFrame (
  Dialog:String;
  Control:String;
  [Left:Numeric];
  [Top:Numeric];
  [Width:Numeric];
  [Height:Numeric];
  [Style:Enumeration {
    Frame!;
    Black!;
    Gray!;
    White!;
    Filled!}))
[Add a Frame (rectangle) control to the dialog]

```

```

String := DialogAddGroupBox (
  Dialog:String;
  Control:String;
  [Left:Numeric];
  [Top:Numeric];
  [Width:Numeric];
  [Height:Numeric];
  Title:String)
[Add a GroupBox control to the dialog]

```

```

String := DialogAddHLine (
  Dialog:String;

```

```

  Control:String;
  [Left:Numeric];
  [Top:Numeric];
  [Length:Numeric])
[Add a Horizontal Line to the dialog]

```

```

String := DialogAddHotSpot (
  Dialog:String;
  Control:String;
  [Left:Numeric];
  [Top:Numeric];
  [Width:Numeric];
  [Height:Numeric];
  [Style:Enumeration {
    Click!;
    DbClick!}))
[Add a HotSpot control to the dialog]

```

```

String := DialogAddIcon (
  Dialog:String;
  Control:String;
  [Left:Numeric];
  [Top:Numeric];
  [Width:Numeric];
  [Height:Numeric];
  IconName:String;
  [Instance:Numeric])
[Add an Icon to the dialog]

```

```

String := DialogAddListBox (
  Dialog:String;
  Control:String;
  [Left:Numeric];
  [Top:Numeric];
  [Width:Numeric];
  [Height:Numeric];
  [Style:Enumeration {
    Unsorted!;
    Sorted!;
    NameSearch!;
    MultiColumn!;
    WPChars!;
    ExtendedSelection!;
    MultipleSelection!;
    UseTabStops!;
    Checkboxes!});
  MacroVar:Variable)
[Add a ListBox control to the dialog]

```

```

Numeric := DialogAddListItem (
  Dialog:String;
  Control:String;
  {Item:String})
[Add an item to a listbox of the dialog]

```

```

String := DialogAddProgress (
  Dialog:String;
  Control:String;
  [Left:Numeric];
  [Top:Numeric];
  [Width:Numeric];
  [Height:Numeric])
[Add a Progress Indicator to the dialog]

```

```

String := DialogAddPushButton (
  Dialog:String;
  Control:String;
  [Left:Numeric];
  [Top:Numeric];
  [Width:Numeric];
  [Height:Numeric];
  [Style:Enumeration {
    NonDefaultBbtn!;

```

```

    DefaultBtn!;
    OKBtn!;
    CancelBtn!;
    HelpBtn!});
    ButtonText:String)
[Add a Pushbutton control to the dialog]

```

```

String := DialogAddRadioButton (
    Dialog:String;
    Control:String;
    [Left:Numeric];
    [Top:Numeric];
    [Width:Numeric];
    [Height:Numeric];
    ButtonText:String;
    MacroVar:Variable;
    [Style:Enumeration {
        Radio!;
        RadioAuto!;
        RadioLeft!});
[Add a RadioButton control to the dialog]

```

```

String := DialogAddScrollBar (
    Dialog:String;
    Control:String;
    [Left:Numeric];
    [Top:Numeric];
    [Width:Numeric];
    [Height:Numeric];
    [Style:Enumeration {
        Left!;
        Top!;
        Right!;
        Bottom!;
        VScroll!;
        HScroll!});
    MacroVar:Variable;
    [Minimum:Numeric];
    [Maximum:Numeric];
    [Step:Numeric])
[Add a Scrollbar control to the dialog]

```

```

String := DialogAddText (
    Dialog:String;
    Control:String;
    [Left:Numeric];
    [Top:Numeric];
    [Width:Numeric];
    [Height:Numeric];
    [Style:Enumeration {
        Left!;
        Right!;
        Center!;
        RecessBox!;
        ShadowBox!;
        WPChars!;
        Multiline!;
        Filename!;
        NoPrefix!});
    Text:String)
[Add Text to the dialog]

```

```

String := DialogAddViewer (
    Dialog:String;
    Control:String;
    [Left:Numeric];
    [Top:Numeric];
    [Width:Numeric];
    [Height:Numeric];
    [Filename:String])
[Add a Viewer to the dialog]

```

```

String := DialogAddVLine (
    Dialog:String;
    Control:String;
    [Left:Numeric];
    [Top:Numeric];
    [Length:Numeric])
[Add a Vertical Line to the dialog]

```

```

Boolean := DialogControlQuery (
    Dialog:String;
    [Control:String];
    [MacroVar:Variable])
[Query the value of a control on a user dialog]

```

```

Boolean := DialogControlUpdate (
    Dialog:String;
    [Control:String];
    [Value:Any])
[Update the value of a control on a user dialog]

```

```

String := DialogDefine (
    Dialog:String;
    Left:Numeric;
    Top:Numeric;
    Width:Numeric;
    Height:Numeric;
    Style:Enumeration {
        OK!;
        Cancel!;
        Percent!;
        NoFrame!;
        Sizeable!;
        NoTitle!;
        Modeless!;
        Enter2HRtn!;
        NoCloseBox!};
    Caption:String)
[Create a dialog]

```

```

Boolean := DialogDelete (Dialog:String)
[Delete a dialog from current macro file]

```

```

Boolean := DialogDestroy (Dialog:String)
[Destroy a dialog]

```

```

String := DialogDismiss (
    Dialog:String;
    Control:String)
[Dismiss a dialog]

```

```

Numeric := DialogHandle (
    Dialog:String;
    [Control:String])
[Get a handle to the specified item]

```

```

Numeric := DialogLoad (
    Dialog:String;
    [Parent:String])
[Loads a dialog, but does not show it]

```

```

Boolean := DialogSave (
    Dialog:String;
    [SaveAsName:String])
[Save specified dialog in current macro file]

```

```

Boolean := DialogSetProperties (
    Dialog:String;
    [FontName:String];
    [FontSize:Numeric Measurement];
    [ClassName:String];
    [HelpFile:String];
    [HelpString:String])
[Set properties for a dialog]

```



**Any := DialogShow** (  
  **Dialog: String;**  
  [**Parent: String;**]  
  [**Callback: Label;**]  
  [**Focus: String;**])  
[Display a dialog]

**Numeric := Dimensions** (  
  **VariableName: Variable;**  
  [**IndexOption: Enumeration {**  
    **DimensionCount!;**  
    **ElementCount!;**  
    **IndexLimit1!;**  
    **IndexLimit2!;**  
    **IndexLimit3!;**  
    **IndexLimit4!;**  
    **IndexLimit5!;**  
    **IndexLimit6!;**  
    **IndexLimit7!;**  
    **IndexLimit8!;**  
    **IndexLimit9!;**  
    **IndexLimit10!}}**])  
[Return dimension information for an array variable]

**Discard** ({**VariableName: Variable;**)  
[Discard and undefine a variable]

**DiscardAll** (  
  [**VariableName: Variable;**]  
  [**Pool: Enumeration {**  
    **AllPools!;**  
    **Local!;**  
    **Global!;**  
    **Persistent!;**  
    **ExactMatch!;**  
    **PartialMatch!}}**])  
[Discard and undefine all variables in a specified pool]

**Numeric := Any DIV Any**  
[Returns the integer quotient of two numbers; Precedence level 3]

**DLLCall** (  
  <**ModuleInstance**>**Numeric;**  
  <**FunctionName**>**String;**  
  [<**ReturnVariable**>**Variable;**]  
  <**ReturnType**>**Enumeration {**  
    **AnsiString!;**  
    **Bool!;**  
    **DWord!;**  
    **Integer!;**  
    **OemString!;**  
    **Real!;**  
    **String!;**  
    **Word!;**  
    **WString!;**  
    **Void!;**  
  }{<**Parameter**>**Any}}**)  
[Call a DLL routine]

**DLLCall Prototype**  
  <**Name**>**Label** (<**ModuleFileName**>**String;**  
  [<**FunctionName**>**String;**]  
  <**ReturnType**>**Enumeration {**  
    **AnsiString!;**  
    **Bool!;**  
    **DWord!;**  
    **Integer!;**  
    **OemString!;**  
    **Real!;**  
    **String!;**  
    **Word!;**  
    **WString!;**  
    **Void!;**

  }{<**Parameter**>**Any}}**)  
[Define the calling format of a DLL routine]

**DLLFree** (**ModuleInstance: Numeric**)  
[Free a DLL module]

**Numeric := DLLLoad** (**Filename: String**)  
[Load a DLL module]

**Boolean := DoesDialogExist** (**Dialog: String**)  
[Looks for dialog within context of the macro]

**Boolean := DoesDirectoryExist** (**DirectoryName: String**)  
[See if a directory (folder) exists]

**Boolean := DoesFileExist** (**Filename: String**)  
[See if a file exists]

**Boolean := DoesRegionExist** (**NamedRgn: String**)  
[Determine if a named region exists or not]

**Numeric := Double** (<**Value**>**Numeric**)  
[Use only in OLE Automation call in-line parameter function; pass value as 8 byte float]

**Numeric := DWord** (<**Value**>**Numeric**)  
[Use only in DLLCall or OLE Automation call in-line parameter function; pass value as unsigned DWORD]

**- E -**

**Else**  
[Statements to execute when an IF expression is FALSE]

**ElseifPlatform** ({<**PlatformID**>**Enumeration {**  
  **Win!;**  
  **Win32!;**  
  **Win95!;**  
  **Win98!;**  
  **WinNT!;**  
  **Win2000!;**  
  **\_Version10!;**  
  **\_Version10\_0\_0!}}**)  
[Statements to compile when an IFPLATFORM expression is FALSE]

**EndApp** (<**ProductPrefix**>**String**)  
[Specify that the specified application is no longer used in a macro]

**EndFor**  
[End the statements in a FOR loop]

**EndFunc**  
[End the statements in a FUNCTION]

**EndFunction**  
[End the statements in a FUNCTION]

**EndIf**  
[End the statements in an IF statement]

**EndIfPlatform** ({<**PlatformID**>**Enumeration {**  
  **Win!;**  
  **Win32!;**  
  **Win95!;**  
  **Win98!;**  
  **WinNT!;**  
  **Win2000!;**  
  **\_Version10!;**  
  **\_Version10\_0\_0!}}**)  
[End the conditional compile statements]

**EndProc**  
[End the statements in a PROCEDURE]

**EndProcedure**

[End the statements in a PROCEDURE]

**EndPrompt ()**

[Terminate the current PROMPT]

**EndSwitch**

[End the statements in a SWITCH statement]

**EndWhile**

[End the statements in a WHILE loop]

**EndWith**

[End the statements in a WITH block]

**String := EnvVariableGet (**  
     [EnvVariableName:String];  
     [EnvType:Enumeration {  
         DOS!}]})

[Get the value of an environment variable]

**EnvVariableSet (**  
     EnvVariableName:String;  
     [Value:String];  
     [EnvType:Enumeration {  
         DOS!}]})

[Set the value of, or delete an environment variable]

**Enumeration {Off!;On!} := Error ([State:Enumeration {Off!;On!}])**  
 [Ignore or accept error conditions]

**Enumeration {**  
     Success!;  
     CancelConditionAsserted!;  
     ErrorConditionAsserted!;  
     NotFoundConditionAsserted!;  
     VarErrChkAsserted!;  
     UserDefinedCondition!}

**:= ErrorNumber ()**

[Return last error that occurred]

**Any {**  
     NotFound!;  
     Local!;  
     Global!;  
     Persistent!}

**:= Exists (**  
     [VariableName:Variable];  
     [Pool:Enumeration {  
         Local!;  
         Global!;  
         Persistent!;  
         ExactMatch!;  
         PartialMatch!;  
         Declared!}]})

[See if a variable exists]

**Enumeration {Off!; On!}**  
**:= ExitHandlerState ([State:Enumeration {Off!;On!}])**  
 [Enable or disable Exit handler on macro end]

**Numeric := ExponentPart (Value:Numeric)**  
 [Return the exponent portion of a numeric value (18 from 3.2e18)]

**- F -**

**Numeric := Factorial (Nth:Numeric)**  
 [Return the Nth factorial (product of all previous terms - 1\*2\*3\*4\*...\*n)]

**Boolean := False**  
 [The constant boolean value 'FALSE']

**Numeric := Fibonacci (Nth:Numeric)**  
 [Return the Nth term in the Fibonacci sequence (sum of previous 2 terms - 0,1,1,2,3,5,8,...)]

**Numeric := FileChecksum (**  
     Filename:String;  
     [Options:Enumeration {  
         Checksum16!;  
         Checksum32!;  
         CRC\_16!;  
         CCITT!;  
         AUTODIN\_III!}];  
     [Seed:Numeric])  
 [Compute a checksum on the contents of a file]

**Enumeration {**  
     Success!;  
     InAndOutSame!;  
     NoDriver!;  
     UnknownGraphic!;  
     Incompatible!;  
     UnknownType!;  
     UnknownTypeSpecified!;  
     UnknownTypeDetected!;  
     InvalidSource!;  
     InvalidSourceSpec!;  
     InvalidSourceName!;  
     InvalidDestination!;  
     InvalidDestinationSpec!;  
     InvalidDestinationName!;  
     UnsupportedFormat!;  
     DetectDriverMissing!;  
     ThirdPartyError!;  
     OutOfMemory!;  
     IncorrectSourceFormat!;  
     FileLocked!;  
     DiskFull!;  
     NoFileDescriptors!;  
     NoGraphicConverter!;  
     DocumentCorrupt!;  
     IgnoringDetectedType!;  
     FileNotSpreadsheet!;  
     EntriesNotConverted!;  
     NoConvertibleData!;  
     InvalidSourceData!}

**:= FileConvert (**  
     SourceFilename:String;  
     [SourceFileType:Enumeration {  
         AutoDetect!;  
         WordPerfect42!;  
         WordPerfect50!;  
         WordPerfect51!;  
         WordPerfect6x!;  
         WordPerfect7!;  
         WordPerfect8!;  
         WordPerfect9!;  
         WordPerfect6789!;  
         WordPerfectCompound!;  
         WP6xText!;  
         WPMac31!;  
         AmiPro30!;  
         WindowsWrite!;  
         MSWord55!;  
         Word60!;  
         Word97!;  
         WordStar70!;  
         AsciiText!;  
         AsciiDelimited!;  
         AsciiCRLFtoSRt!;  
         AnsiText!;  
         AnsiDelimited!;  
         AnsiCRLFtoSRt!;

RichTextFormat!;  
 UnicodeText!;  
 WPG1!;  
 WPG2!;  
 TIFF!;  
 WMF!;  
 BMP!;  
 GIF!;  
 JPG!;  
 CoreIPHOTOPaint7!;  
 CoreIDRAW7!;  
 CoreIDRAW8!;  
 CoreIDRAW9!;  
 CoreIDRAW10!;  
 CoreIDRAWClipart!;  
 AdobePhotoshop!;  
 KodakPhotoCD!;  
 PCPaintBrush!;  
 Excel70!;  
 Excel97!;  
 Lotus123v4!;  
 SDIF!;  
 QuattroPro6!;  
 QuattroPro7!;  
 QuattroPro8!;  
 QuattroPro9!;  
 QuattroPro78!}}

UnknownGraphic!;  
 Incompatible!;  
 UnknownType!;  
 UnknownTypeSpecified!;  
 UnknownTypeDetected!;  
 InvalidSource!;  
 InvalidSourceSpec!;  
 InvalidSourceName!;  
 InvalidDestination!;  
 InvalidDestinationSpec!;  
 InvalidDestinationName!;  
 UnsupportedFormat!;  
 DetectDriverMissing!;  
 ThirdPartyError!;  
 OutOfMemory!;  
 IncorrectSourceFormat!;  
 FileLocked!;  
 DiskFull!;  
 NoFileDescriptors!;  
 NoGraphicConverter!;  
 DocumentCorrupt!;  
 IgnoringDetectedType!;  
 FileNotSpreadsheet!;  
 EntriesNotConverted!;  
 NoConvertibleData!;  
 InvalidSourceData!}}

[Get a string representing the error return after file conversion]

DestinationFilename:String;  
 DestinationFileType:Enumeration {  
 WordPerfect42!;  
 WordPerfect50!;  
 WordPerfect51!;  
 WordPerfect6x!;  
 WordPerfect7!;  
 WordPerfect8!;  
 WordPerfect9!;  
 WordPerfect6789!;  
 WordPerfectCompound!;  
 WP6xText!;  
 AmiPro30!;  
 WindowsWrite!;  
 MSWord55!;  
 Word60!;  
 Word97!;  
 WordStar70!;  
 AsciiText!;  
 AsciiDelimited!;  
 AsciiGenericWPI!;  
 AnsiText!;  
 AnsiDelimited!;  
 AnsiGenericWPI!;  
 RichTextFormat!;  
 UnicodeText!;  
 WPG1!;  
 WPG2!;  
 TIFF!;  
 WMF!;  
 BMP!;  
 GIF!;  
 JPG!;  
 CoreIPHOTOPaint7!;  
 PCPaintBrush!;  
 PresentationsShow!;  
 Lotus123v4!;  
 SDIF!;  
 QuattroPro6!}}

[Convert a file to another type]

String := **FileConvertError** (ErrorCode:Enumeration {  
 Success!;  
 InAndOutSame!;  
 NoDriver!;

Enumeration {  
 Success!;  
 UnknownError!;  
 Error!;  
 OutOfMemory!;  
 AllocationError!;  
 BadFileFormat!;  
 FileLocked!;  
 CantLockFile!;  
 AccessDenied!;  
 NoFileHandles!;  
 FileExists!;  
 InvalidDrivePath!;  
 FileNotFound!;  
 DiskFull!;  
 FileReadError!;  
 InvalidFileHandle!;  
 CantOpenFile!}  
 := **FileError** ()  
 [Get last file error]

String := **FileFind** (  
 [Filename:String];  
 [Attributes:Enumeration {  
 Normal!;  
 ReadOnly!;  
 Hidden!;  
 System!;  
 Label!;  
 Directory!;  
 Archive!;  
 SkipDotDirs!;  
 SearchSubDirs!;  
 MatchAnyAttribute!;  
 MinimumAttributes!;  
 ExactAttributes!});  
 [Context:Numeric])  
 [Find a matching file]

Boolean := **FileFlushData** (FileID:Numeric)  
 [Flush file data from internal buffers to disk]

String := **FileGetLongName** (Filename:String)  
 [Get the long form of a full file name]

**String := FileGetShortName** (Filename:String)  
[Get the short 8.3 compatible form of a full file name]

**Boolean := FileIsEOF** (FileID:Numeric)  
[Test file for end-of-file condition]

**String := FileNameDialog** (  
[StyleOptions:Enumeration {  
  OpenDialog!;  
  SaveAsDialog!;  
  DirectoryDialog!;  
  WindowsDialog!;  
  EnhancedDialog!;  
  FileMustExist!;  
  MultipleFiles!;  
  PromptOnReplace!}];  
[CaptionText:String];  
[ButtonText:String];  
[InitialFolder:String];  
[InitialFilename:String];  
[FileTemplate:String];  
[RecentFiles:String];  
[Separator:String];  
[Parent:String])  
[Display file name dialog and return file name]

**Enumeration {Error!} := FilePosition** (  
FileID:Numeric;  
[NewPosition:Numeric];  
[PositionFrom:Enumeration {  
  FromBeginning!;  
  FromCurrentPosition!;  
  FromEnd!}])  
[Get or set file position]

**Enumeration {Error!} := FileRead** (  
FileID:Numeric;  
Data:Variable)  
[Read data from a file]

**Enumeration {Error!} := FileSize** (Filename:String)  
[Get a file's size]

**Enumeration {Error!} := FileTruncate** (FileID:Numeric)  
[Truncate a data file]

**Enumeration {**  
  OpenReadError!;  
  OutOfMemory!;  
  PasswordProtected!;  
  Success!;  
  Unknown!;  
  WordPerfect42!;  
  WordPerfect50!;  
  WordPerfect51!;  
  WordPerfect6789!;  
  WordPerfectCompound!;  
  WPMac31!;  
  AmiPro30!;  
  WindowsWrite!;  
  MSWord55!;  
  Word60!;  
  Word97!;  
  WordStar70!;  
  AsciiText!;  
  AnsiText!;  
  RichTextFormat!;  
  UnicodeText!;  
  HTML!;  
  SGML!;  
  XML!;  
  WPG1!;  
  WPG2!;

  TIFF!;  
  WMF!;  
  BMP!;  
  GIF!;  
  JPG!;  
  CorelPHOTOPaint7!;  
  CorelDRAW7!;  
  CorelDRAW8!;  
  CorelDRAW9!;  
  CorelDRAW10!;  
  AdobePhotoshop!;  
  KodakPhotoCD!;  
  PCPaintBrush!;  
  PresentationsMaster!;  
  PresentationsMaster3!;  
  PresentationsShow!;  
  PresentationsShow3!;  
  Excel70!;  
  Excel97!;  
  Lotus123v4!;  
  SDIF!;  
  QuattroPro6!;  
  QuattroPro78!;  
  QuattroPro9!;  
  Paradox!;  
  Zip!}

**:= FileType** (Filename:String)  
[Get the type of a file]

**String := FileTypeExtension** (FileType:Enumeration {  
  WordPerfect42!;  
  WordPerfect50!;  
  WordPerfect51!;  
  WordPerfect6x!;  
  WordPerfect7!;  
  WordPerfect8!;  
  WordPerfect9!;  
  WordPerfect6789!;  
  WordPerfectCompound!;  
  WP6xText!;  
  WPMac31!;  
  AmiPro30!;  
  WindowsWrite!;  
  MSWord55!;  
  Word60!;  
  Word97!;  
  WordStar70!;  
  AsciiText!;  
  AsciiDelimited!;  
  AsciiCRLFtoSR!;  
  AsciiGenericWP!;  
  AnsiText!;  
  AnsiDelimited!;  
  AnsiCRLFtoSR!;  
  AnsiGenericWP!;  
  RichTextFormat!;  
  UnicodeText!;  
  HTML!;  
  SGML!;  
  XML!;  
  WPG1!;  
  WPG2!;  
  TIFF!;  
  WMF!;  
  BMP!;  
  GIF!;  
  JPG!;  
  CorelPHOTOPaint7!;  
  CorelDRAW7!;  
  CorelDRAW8!;  
  CorelDRAW9!;  
  CorelDRAW10!;

CoreIDRAWClipart!  
 AdobePhotoshop!  
 KodakPhotoCD!  
 PCPaintBrush!  
 PresentationsMaster!  
 PresentationsMaster3!  
 PresentationsShow!  
 PresentationsShow3!  
 Excel70!  
 Excel97!  
 Lotus123v4!  
 SDIF!  
 QuattroPro6!  
 QuattroPro7!  
 QuattroPro8!  
 QuattroPro9!  
 QuattroPro78!  
 Paradox!  
 Zip!))

[Get the default file name extension associated with a file type number]

Array := **FileTypeList** (ListType:Enumeration {  
 Detect!  
 Source!  
 Destination!})

[Get a list of detectable or convertible file types]

String := **FileTypeName** (FileType:Enumeration {  
 WordPerfect42!  
 WordPerfect50!  
 WordPerfect51!  
 WordPerfect6x!  
 WordPerfect7!  
 WordPerfect8!  
 WordPerfect9!  
 WordPerfect6789!  
 WordPerfectCompound!  
 WP6xText!  
 WPMac31!  
 AmiPro30!  
 WindowsWrite!  
 MSWord55!  
 Word60!  
 Word97!  
 WordStar70!  
 AsciiText!  
 AsciiDelimited!  
 AsciiCRLFtoSRt!  
 AsciiGenericWP!  
 AnsiText!  
 AnsiDelimited!  
 AnsiCRLFtoSRt!  
 AnsiGenericWP!  
 RichTextFormat!  
 UnicodeText!  
 HTML!  
 SGML!  
 XML!  
 WPG1!  
 WPG2!  
 TIFF!  
 WMF!  
 BMP!  
 GIF!  
 JPG!  
 CoreIPHOTOPaint7!  
 CoreIDRAW7!  
 CoreIDRAW8!  
 CoreIDRAW9!  
 CoreIDRAW10!  
 CoreIDRAWClipart!

AdobePhotoshop!  
 KodakPhotoCD!  
 PCPaintBrush!  
 PresentationsMaster!  
 PresentationsMaster3!  
 PresentationsShow!  
 PresentationsShow3!  
 Excel70!  
 Excel97!  
 Lotus123v4!  
 SDIF!  
 QuattroPro6!  
 QuattroPro7!  
 QuattroPro8!  
 QuattroPro9!  
 QuattroPro78!  
 Paradox!  
 Zip!))

[Get the name associated with a file type number]

Enumeration {Error!} := **FileWrite** (  
 FileID:Numeric;  
 Data:String;  
 [NewLine:Enumeration {  
 NoNewLine!  
 NewLine!  
 FlushData!};  
 {[ParameterData:String]})

[Write data to a file]

Enumeration {Error!} := **FileWriteLine** (  
 FileID:Numeric;  
 [Options:Enumeration {  
 BufferData!  
 FlushData!}])

[Write a new line terminator to a file]

Numeric := **Floor** (Value:Numeric)  
 [Return the floor of a number (greatest integer <= Value)]

**For** (  
 <ControlVariable>Variable;  
 <InitialValue>Any;  
 <TerminateExp>Boolean;  
 <IncrementExp>Any)  
 [Execute a group of statements a number of times]

**ForEach** (  
 <ControlVariable>Variable;  
 {<ValueList>Any})  
 [Execute a group of statements a number of times]

**ForNext** (  
 <ControlVariable>Variable;  
 <InitialValue>Any;  
 <FinalValue>Any;  
 [<IncrementValue>Any])  
 [Execute a group of statements a number of times]

Numeric := **Fraction** (Value:Numeric)  
 [Return the fractional portion of a numeric value]

Numeric := **FractionalPart** (Value:Numeric)  
 [Return the fractional portion of a numeric value (0.2 from 3.2)]

String := **FractionStr** (  
 Value:Numeric;  
 [Denominator:Numeric];  
 [Option:Enumeration {  
 ImproperFraction!  
 MixedFraction!}])  
 [Return a fraction string representing a value]

**Function** <Name>Label ({<Parameter>Any})  
[Begin the statements to be associated with a FUNCTION label]

**Function Prototype** <Name>Label ({<Parameter>Any})  
[Define the calling format of a FUNCTION label to be defined later]

- G -

**Numeric := gcf** (Value:Numeric)  
[Return the greatest common factor of a list of values]

**String := GetCurrentDirectory** ()  
[Get current directory (folder)]

**Enumeration** {  
  Error!;  
  Normal!;  
  ReadOnly!;  
  Hidden!;  
  System!;  
  Label!;  
  Directory!;  
  Archive!}

**:= GetFileAttributes** (Filename:String)  
[Get file attributes]

**Any {Error!} := GetFileDateAndTime** (  
  Filename:String;  
  [DateType:Enumeration {  
    CreationDateTime!;  
    LastAccessDateTime!;  
    ModificationDateTime!}])  
[Get file date and time]

**GetNumber** (  
  VariableName:Variable;  
  [Prompt:String];  
  [Title:String])  
[Prompt for, input and return a numeric value]

**Object := GetObject** (  
  [Filename:String];  
  [ClassName:String])  
[Retrieve an OLE Automation object from a file]

**GetString** (  
  VariableName:Variable;  
  [Prompt:String];  
  [Title:String];  
  [Length:Numeric])  
[Prompt for, input and return a string value]

**GetUnits** (  
  VariableName:Variable;  
  [Prompt:String];  
  [Title:String])  
[Prompt for, input and return a measurement value]

**Global** (  
  {<VariableName>Variable;  
  <Value>Any})  
[Declare a global variable or array]

**Go** (<Label>Label)  
[Jump to the macro code at the specified label]

**Numeric := GoOnline** (  
  [Service:Enumeration {  
    Internet!;  
    CompuServe!;  
    Other!}];  
  [Location:String])  
[Starts an online session to CompuServe or the Internet]

- I -

**If** (<Test>Boolean)  
[Execute statements if a condition is TRUE]

**IfPlatform** ({<PlatformID>Enumeration {  
  Win!;  
  Win32!;  
  Win95!;  
  Win98!;  
  WinNT!;  
  Win2000!;  
  \_Version10!;  
  \_Version10\_0\_0!}})  
[Specify statements to be compiled only if compiling on the specified platform(s)]

**Boolean := Any IN Any**  
[Returns TRUE if left is contained in the right array, otherwise FALSE; Precedence level 6]

**Include** (<FileName>String)  
[Include a file at compile time into the current macro file]

**Any := Indirect** (<VariableName>String)  
[Return the value of the variable specified by the string expression]

**Numeric := Integer** (<Value>Numeric)  
[Use only in OLE Automation call in-line parameter function; pass value as short int]

**Numeric := Integer** (Value:Numeric)  
[Return the integer portion of a numeric value]

**Numeric := IntegerPart** (Value:Numeric)  
[Return the integer portion of a numeric value (3.0 from 3.2)]

- L -

**Label** (<Label>Label)  
[Mark a spot in the macro with the specified label name]

**Numeric := Lcm** ({Value:Numeric})  
[Return the least common multiple of a list of values]

**Numeric := Lg** (Value:Numeric)  
[Return the base 2 (binary) logarithm of a value]

**Boolean := Any LIKE Any**  
[Returns TRUE if left is equal to right (case insensitive), otherwise FALSE; Precedence level 6]

**Numeric := Ln** (Value:Numeric)  
[Return the base 'e' (natural or naperian) logarithm of a value]

**Local** (  
  {<VariableName>Variable;  
  <Value>Any})  
[Declare a local variable or array]

**Numeric := Log** (Value:Numeric)  
[Return the base 10 (common) logarithm of a value]

**Numeric := Logn** (  
  Value:Numeric;  
  Base:Numeric)  
[Return the base n logarithm of a value]

**Numeric := Long** (<Value>Numeric)  
[Use only in OLE Automation call in-line parameter function; pass value as LONG]

**Any** := **MacroArgs**[ ] (<Index>Numeric)  
[Array variable with parameter arguments passed to the macro]

**Any** {**Success**!} := **MacroCompile** (  
  **MacroFilename**:String;  
  [**Options**:Enumeration {  
    **Wait**!;  
    **DontWait**!;  
    **Debug**!;  
    **NoPrompts**!;  
    **ShowProgress**!;  
    **GenerateListing**!;  
    **ForceRecompile**!;  
    **ShowIcon**!}])  
[Compile a macro]

**String** := **MacroDialogResult**  
[Return result of last 'Dialog' command]

**Any** {  
  **TokenOK**!;  
  **UnknownToken**!;  
  **NotFoundCondition**!;  
  **CancelCondition**!;  
  **ErrorCondition**!;  
  **InvalidParameter**!;  
  **TokenInvalid**!;  
  **NotHandled**!;  
  **None**!;  
  **Centimeters**!;  
  **Inches**!;  
  **Millimeters**!;  
  **Points**!;  
  **WPUnits**!;  
  **Success**!;  
  **CancelConditionAsserted**!;  
  **ErrorConditionAsserted**!;  
  **NotFoundConditionAsserted**!;  
  **VarErrChkAsserted**!;  
  **UserDefinedCondition**!}

:= **MacroInfo** (  
  **InfoItem**:Enumeration {  
    **ComputerName**!;  
    **UserName**!;  
    **WindowsDirectory**!;  
    **SystemDirectory**!;  
    **PlatformName**!;  
    **PlatformVersion**!;  
    **PlatformVersionString**!;  
    **PerfectFitVersion**!;  
    **PerfectFitVersionString**!;  
    **PerfectScriptVersion**!;  
    **PerfectScriptVersionString**!;  
    **PerfectScriptLanguageCode**!;  
    **PerfectScriptLanguageName**!;  
    **TimeWindowsStarted**!;  
    **TimeWindowsElapsed**!;  
    **DialogsExist**!;  
    **BreakpointsExist**!;  
    **ErrorNumber**!;  
    **ErrorText**!;  
    **TokenError**!;  
    **AbortingTokenName**!;  
    **MacroDialogResult**!;  
    **MacroArgs**!;  
    **TimeMacroStarted**!;  
    **TimeElapsed**!;  
    **TimeElapsedNet**!;  
    **TimeInProductCalls**!;  
    **TimeInDLLCalls**!;

**TimeInOLEObjects**!;  
  **DLLCallBufferSize**!;  
  **SpeedDelay**!;  
  **CallDepth**!;  
  **InRoutine**!;  
  **InCallback**!;  
  **InAssert**!;  
  **Filename**!;  
  **LineNumber**!;  
  **LabelOrRoutineName**!;  
  **SourceLine**!;  
  **ChainFilename**!;  
  **ChainFileArgs**!;  
  **DefaultUnits**!;  
  **PersistAllOn**!;  
  **CancelOn**!;  
  **CancelSetup**!;  
  **CancelCall**!;  
  **CancelHandler**!;  
  **ErrorOn**!;  
  **ErrorSetup**!;  
  **ErrorCall**!;  
  **ErrorHandler**!;  
  **NotFoundOn**!;  
  **NotFoundSetup**!;  
  **NotFoundCall**!;  
  **NotFoundHandler**!;  
  **VarErrChkOn**!;  
  **VarErrChkSetup**!;  
  **VarErrChkCall**!;  
  **VarErrChkHandler**!;  
  **ExitOn**!;  
  **ExitSetup**!;  
  **ExitCall**!;  
  **ExitHandler**!;  
  **UserDefinedConditionOn**!;  
  **UserDefinedConditionSetup**!;  
  **UserDefinedConditionCall**!;  
  **UserDefinedConditionHandler**!);  
  [**CallLevel**:Enumeration {  
    **Current**!;  
    **Previous**!}];  
  {**ItemData**:Any {  
    **UserDefinedCondition**!}})  
[Return information about the current macro state]

**Boolean** := **MacrolsCompiled** (**MacroFilename**:String)  
[See if a macro has been compiled]

**Any** {**Success**!} := **MacroPlay** (  
  **MacroFilename**:String;  
  [**Options**:Enumeration {  
    **Wait**!;  
    **DontWait**!;  
    **Debug**!;  
    **ShowIcon**!}];  
  {**Parameter**:Any})  
[Play a macro]

**Numeric** := **MantissaPart** (**Value**:Numeric)  
[Return the mantissa portion of a numeric value (3.2 from 32.0e18)]

**Numeric** := **Max** ({**Value**:Numeric})  
[Return the maximum value of a list of values]

**Numeric** := **MaxN** (  
  **Nth**:Numeric;  
  {**Value**:Numeric})  
[Return the Nth maximum value of a list of values]

**Numeric** := **MeanDev** (  
  [**Type**:Enumeration {  
    **Population**!;

**Sample!};**  
**{Value:Numeric}**  
[Return the mean deviation of a list of values]

**Numeric := Median** (**{Value:Numeric}**)  
[Return the median value of a list of values]

**Numeric := Menu** (  
  **[MnemonicType:Enumeration** {  
    **Digit!**;  
    **Letter!**;  
    **Either!};**  
  **[HorizontalPosition:Numeric];**  
  **[VerticalPosition:Numeric];**  
  **{MenuChoice:String}**)  
[Display a menu and return a menu pick]

**Enumeration** {  
  **Error!**;  
  **OKButton!**;  
  **CancelButton!**;  
  **AbortButton!**;  
  **RetryButton!**;  
  **IgnoreButton!**;  
  **YesButton!**;  
  **NoButton!**  
  **:= MessageBox** (  
    **[Status:Variable];**  
    **[Caption:String];**  
    **Message:String;**  
    **[Style:Enumeration** {  
      **AbortRetryIgnore!**;  
      **OK!**;  
      **OKCancel!**;  
      **RetryCancel!**;  
      **YesNo!**;  
      **YesNoCancel!**;  
      **IconNone!**;  
      **IconQuestion!**;  
      **IconAsterisk!**;  
      **IconExclamation!**;  
      **IconHand!**;  
      **IconStop!**;  
      **IconInformation!**;  
      **IconWarning!**;  
      **IconError!**;  
      **DefButton1!**;  
      **DefButton2!**;  
      **DefButton3!**;  
      **ApplicationModal!**;  
      **SystemModal!**;  
      **TaskModal!**;  
      **Beep!**;  
      **HasParameters!};**  
    **{[ParameterData:String]}**)  
[Display a message box]

**Numeric := Min** (**{Value:Numeric}**)  
[Return the minimum value of a list of values]

**Numeric := MinN** (  
  **Nth:Numeric;**  
  **{Value:Numeric}**)  
[Return the Nth minimum value of a list of values]

**Numeric := MMPlay** (  
  **Filename:String;**  
  **[Options:Enumeration** {  
    **DontWait!**;  
    **Wait!};**)  
[Play a multimedia file]

**MMSpeak (String:String)**  
[Speak a string of text]

**MMSpeakClipboard ()**  
[Speak the contents of the clipboard]

**MMStopPlay (Handle:Numeric)**  
[Stop a playing multimedia file]

**MMStopSpeech ()**  
[Stop the previous speech command]

**Numeric := Any MOD Any**  
[Returns the integer remainder of the quotient of two numbers;  
Precedence level 3]

- N -

**Nest** (  
  **MacroFilename:String;**  
  **{[Parameter:Any]}**)  
[Execute the macro code found in the specified macro file]

**Enumeration** {  
  **Success!**;  
  **DialogCanceled!**;  
  **AccessDenied!**;  
  **AlreadyAssigned!**;  
  **AlreadyRemembered!**;  
  **CantOpenUserProfile!**;  
  **DeviceInUse!**;  
  **DeviceMismatch!**;  
  **FilesStillOpen!**;  
  **InsufficientMemory!**;  
  **InvalidDevice!**;  
  **InvalidNetOrPath!**;  
  **InvalidParameter!**;  
  **InvalidPassword!**;  
  **InvalidProvider!**;  
  **InvalidResource!**;  
  **InvalidUserProfile!**;  
  **NetworkError!**;  
  **NoConnection!**;  
  **NoNetwork!**;  
  **NotConnected!**;  
  **NotSupported!**;  
  **ProviderBusy!**;  
  **ResourceNotContainer!**  
  **:= NetAddConnection** (  
    **RemoteName:String;**  
    **[LocalName:String];**  
    **[Provider:String];**  
    **[Type:Enumeration** {  
      **Disk!**;  
      **Printer!};**  
    **[Password:String];**  
    **[UserName:String];**  
    **[Options:Enumeration** {  
      **DontAutoRestore!**;  
      **AutoRestore!};**)  
[Makes a connection to a network resource]

**Boolean := NetCancelConnection** (  
  **ResourceName:String;**  
  **[Options:Enumeration** {  
    **DontAutoRestore!**;  
    **AutoRestore!};**  
  **[Force:Boolean]**)  
[Terminates a connection to a network resource]

**Enumeration** {  
  **Success!**;  
  **DialogCanceled!**;



**AccessDenied!**;  
**AlreadyAssigned!**;  
**AlreadyRemembered!**;  
**CantOpenUserProfile!**;  
**DeviceInUse!**;  
**DeviceMismatch!**;  
**FilesStillOpen!**;  
**InsufficientMemory!**;  
**InvalidDevice!**;  
**InvalidNetOrPath!**;  
**InvalidParameter!**;  
**InvalidPassword!**;  
**InvalidProvider!**;  
**InvalidResource!**;  
**InvalidUserProfile!**;  
**NetworkError!**;  
**NoConnection!**;  
**NoNetwork!**;  
**NotConnected!**;  
**NotSupported!**;  
**ProviderBusy!**;  
**ResourceNotContainer!**

**:= NetConnectionDlg** ([Type:Enumeration {  
 Disk!})

[Starts a general browsing dialog box for connecting to network resources]

**Enumeration {**  
**Success!**;  
**DialogCanceled!**;  
**AccessDenied!**;  
**AlreadyAssigned!**;  
**AlreadyRemembered!**;  
**CantOpenUserProfile!**;  
**DeviceInUse!**;  
**DeviceMismatch!**;  
**FilesStillOpen!**;  
**InsufficientMemory!**;  
**InvalidDevice!**;  
**InvalidNetOrPath!**;  
**InvalidParameter!**;  
**InvalidPassword!**;  
**InvalidProvider!**;  
**InvalidResource!**;  
**InvalidUserProfile!**;  
**NetworkError!**;  
**NoConnection!**;  
**NoNetwork!**;  
**NotConnected!**;  
**NotSupported!**;  
**ProviderBusy!**;  
**ResourceNotContainer!**

**:= NetDisconnectDlg** ([Type:Enumeration {  
 Disk!  
 Printer!})

[Starts a general browsing dialog box for disconnecting from network resources]

**String := NetGetConnection (LocalName:String)**

[The name of a network resource associated with a local device]

**String := NetGetUniversalName (LocalPath:String)**

[The UNC name of a file]

**String := NetGetUser ()**

[A string representing the user network ID]

**NewDefault (<ProductPrefix>String)**

[Specify a new application as the default in the macro]

**Next**

[Advance to the next iteration in a loop statement]

**Boolean := NOT Boolean**

[Returns the BOOLEAN complement of a BOOLEAN value;  
 Precedence level 1]

**Enumeration {Off!;On!} := NotFound** ([State:Enumeration {  
 Off!  
 On!})

[Ignore or accept NotFound conditions]

**Numeric := NthRoot (**  
**Value:Numeric;**  
**Root:Numeric)**

[Return the Nth root of a value]

**String := NToC (<Character>Enumeration {**

**Space!**;  
**HardSpace!**;  
**Tab!**;  
**Hrt!**;  
**HardReturn!**;  
**Hpg!**;  
**HardPage!**;  
**ExtraLargeOn!**;  
**ExtraLargeOff!**;  
**VeryLargeOn!**;  
**VeryLargeOff!**;  
**LargeOn!**;  
**LargeOff!**;  
**SmallOn!**;  
**SmallOff!**;  
**FineOn!**;  
**FineOff!**;  
**SuperScriptOn!**;  
**SuperScriptOff!**;  
**SubScriptOn!**;  
**SubScriptOff!**;  
**OutlineOn!**;  
**OutlineOff!**;  
**ShadowOn!**;  
**ShadowOff!**;  
**BoldOn!**;  
**BoldOff!**;  
**ItalicsOn!**;  
**ItalicsOff!**;  
**UnderLineOn!**;  
**UnderLineOff!**;  
**DoubleUnderlineOn!**;  
**DoubleUnderlineOff!**;  
**RedlineOn!**;  
**RedlineOff!**;  
**StrikeOutOn!**;  
**StrikeOutOff!**;  
**SmallCapsOn!**;  
**SmallCapsOff!**)

[Convert a number to its character equivalent]

**String := NToC (**  
**CharacterOrCharSet:Enumeration {**

**Space!**;  
**HardSpace!**;  
**Tab!**;  
**Hrt!**;  
**HardReturn!**;  
**Hpg!**;  
**HardPage!**;  
**ExtraLargeOn!**;  
**ExtraLargeOff!**;  
**VeryLargeOn!**;  
**VeryLargeOff!**;  
**LargeOn!**;  
**LargeOff!**;  
**SmallOn!**

**SmallOff!**;  
**FineOn!**;  
**FineOff!**;  
**SuperScriptOn!**;  
**SuperScriptOff!**;  
**SubScriptOn!**;  
**SubScriptOff!**;  
**OutlineOn!**;  
**OutlineOff!**;  
**ShadowOn!**;  
**ShadowOff!**;  
**BoldOn!**;  
**BoldOff!**;  
**ItalicsOn!**;  
**ItalicsOff!**;  
**UnderLineOn!**;  
**UnderLineOff!**;  
**DoubleUnderlineOn!**;  
**DoubleUnderlineOff!**;  
**RedlineOn!**;  
**RedlineOff!**;  
**StrikeOutOn!**;  
**StrikeOutOff!**;  
**SmallCapsOn!**;  
**SmallCapsOff!**;  
**[Character:Numeric]**  
 [Convert a character set and character number to their character equivalent]

**String := NumStr (**  
   **Value:Any;**  
   **[RightDigits:Numeric];**  
   **[LeftDigits:Numeric]**  
 [Return a numeric value converted to a string]

- O -

**Object :=Object (<Value>Object)**  
 [Use only in OLE Automation call in-line parameter function; pass value as DISPATCH]

**Object (**  
   **<ObjectVariable>Variable;**  
   **[<ObjectType>String];**  
   **[<Options>Enumeration {**  
     **Default!}]**  
 [Specify prefix variable for an OLE object]

**Any := ObjectInfo (**  
   **ObjectVariable:Any;**  
   **[Infoltem:Enumeration {**  
     **IsObject!;**  
     **IsConnected!;**  
     **ClassName!}]**  
 [Get information about an OLE Automation object]

**String := OemString (<Value>Undefined)**  
 [Use only in DLLCall or OLE Automation call in-line parameter function; pass value as OEM string]

**Label := OnCancel ([Label:Label])**  
 [Specify label of macro code to jump to on CANCEL conditions]

**Label := OnCancel Call ([Label:Label])**  
 [Specify label of macro code to call on CANCEL conditions]

**Label := OnCondition (**  
   **Condition:Enumeration {**  
     **CancelCondition!;**  
     **ErrorCondition!;**  
     **NotFoundCondition!;**  
     **ExitCondition!;**  
     **VarErrChkCondition!;**

**UserDefinedCondition!};**  
**[Label:Label]**  
 [Specify label of macro code to jump to on specified condition]

**Label := OnCondition Call (**  
   **Condition:Enumeration {**  
     **CancelCondition!;**  
     **ErrorCondition!;**  
     **NotFoundCondition!;**  
     **ExitCondition!;**  
     **VarErrChkCondition!;**  
     **UserDefinedCondition!};**  
**[Label:Label]**  
 [Specify label of macro code to call on specified condition]

**Enumeration {**  
   **Success!;**  
   **AdvAckTimeout!;**  
   **Busy!;**  
   **DataAckTimeout!;**  
   **DIINotInitialized!;**  
   **DIUsage!;**  
   **ExeAckTimeout!;**  
   **InvalidParameter!;**  
   **LowMemory!;**  
   **MemoryError!;**  
   **NotProcessed!;**  
   **NoConvEstablished!;**  
   **PokeAckTimeout!;**  
   **PostMsgFailed!;**  
   **Reentrancy!;**  
   **ServerDied!;**  
   **SysError!;**  
   **UnAdvAckTimeout!;**  
   **UnfoundQueueId!}**  
**:= OnDDEAdvise Call (**  
   **ConversationID:Numeric;**  
   **ItemName:String;**  
   **Label:Label)**  
 [Specify the label of macro code to call on DDE Advise conditions]

**Label := OnError ([Label:Label])**  
 [Specify label of macro code to jump to on ERROR conditions]

**Label := OnError Call ([Label:Label])**  
 [Specify label of macro code to call on ERROR conditions]

**Label := OnExit ([Label:Label])**  
 [Specify label of macro code to jump to on macro exit]

**Label := OnNotFound ([Label:Label])**  
 [Specify label of macro code to jump to on NOTFOUND conditions]

**Label := OnNotFound Call ([Label:Label])**  
 [Specify label of macro code to call on NOTFOUND conditions]

**Label := OnVarErrChk ([Label:Label])**  
 [Specify label of macro code to jump to on undefined variable accesses]

**Label := OnVarErrChk Call ([Label:Label])**  
 [Specify label of macro code to call on undefined variable accesses]

**Enumeration {Error!} := OpenFile (**  
   **Name:String;**  
   **[AccessMode:Enumeration {**  
     **Read!;**  
     **ReadWrite!;**  
     **Write!;**  
     **WriteNew!;**  
     **Append!;**  
     **Exists!}]**;  
   **[ShareMode:Enumeration {**

**None!**;  
**Compatibility!**;  
**Exclusive!**;  
**DenyNone!**;  
**DenyRead!**;  
**DenyWrite!}}**;  
**[DataType:Enumeration {**  
**AnsiText!**;  
**OEMText!**;  
**WPText!}}**)  
 [Open a file]

**Boolean := Boolean OR Boolean**  
 [Returns BOOLEAN OR of two BOOLEANS; Precedence level 9]

- P -

**Pause ()**  
 [Pause macro execution]

**Numeric := PercentChange (**  
**Value:Numeric;**  
**Value2:Numeric)**  
 [Return what percent to increase 'Value' by to get 'Value2']

**Numeric := PercentOf (**  
**Percent:Numeric;**  
**Value:Numeric)**  
 [Return 'Percent' percent of 'Value']

**Numeric := PercentOfTotal (**  
**Value:Numeric;**  
**Total:Numeric)**  
 [Return what percent 'Value' is of 'Total']

**Persist (**  
**{<VariableName>Variable;**  
**<Value>Any})**  
 [Declare a persistent variable or array]

**Enumeration {Off!;On!} := PersistAll ([State:Enumeration {**  
**Off!**;  
**On!**;  
**NoChange!}})**  
 [Create all implicitly created variables in the persistent table]

**Numeric := pi ()**  
 [Return 'pi' - the ratio of the circumference of a circle to its diameter]

**Array := PrimeFactors (N:Numeric)**  
 [Return the prime factors of a number]

**Procedure <Name>Label ({<Parameter>Any})**  
 [Begin the statements to be associated with a PROCEDURE label]

**Procedure Prototype <Name>Label ({<Parameter>Any})**  
 [Define the calling format of a PROCEDURE label to be defined later]

**Numeric := Product ({Value:Numeric})**  
 [Return the product of a list of values]

**Prompt (**  
**Title:String;**  
**[Prompt:String];**  
**[Style:Enumeration {**  
**NoIcon!**;  
**QuestionMark!**;  
**InformationIcon!**;  
**WarningIcon!**;  
**ErrorIcon!**;  
**Asterisk!**;  
**ExclamationPoint!**;  
**StopSign!**;  
**}]**

**ApplicationIcon!**;  
**WindowsLogo!**;  
**NoPause!**;  
**Pause!**;  
**Beep!**;  
**NoButtons!**;  
**Buttons!}}**;  
**[HorizontalPosition:Numeric];**  
**[VerticalPosition:Numeric])**  
 [Display a prompt message]

- Q -

**Quit ()**  
 [Terminate macro execution]

- R -

**Numeric := RadiansToDegrees (Radians:Numeric)**  
 [Convert an angle in radians to degrees]

**Numeric := Randomize ([Seed:Numeric])**  
 [Randomize the random number seed]

**Numeric := RandomNumber (**  
**[Minimum:Numeric];**  
**[Maximum:Numeric])**  
 [Return a random number in a range]

**Numeric := RandomSeed ()**  
 [Return the random number seed that will be used to generate the next random number]

**Numeric := RandomStats (Statistic:Enumeration {**  
**Count!**;  
**Minimum!**;  
**Maximum!**;  
**Mean!**;  
**StdDev!**;  
**Variance!**;  
**Total!**;  
**Seed!})**  
 [Return statistical data about the random number generator]

**Numeric := Real (<Value>Numeric)**  
 [Use only in DLLCall or OLE Automation call in-line parameter function; pass value as 8 byte float]

**Numeric := RegionAddListItem (**  
**NamedRgn:String;**  
**{Item:String})**  
 [Add an item to a listbox or combobox list]

**Numeric := RegionAddListItemByIndex (**  
**NamedRgn:String;**  
**Index:Numeric;**  
**Item:String)**  
 [Add an item by index to a listbox or combobox list]

**Enumeration {Disabled!;Enabled!} := RegionEnableWindow (**  
**NamedRgn:String;**  
**[State:Enumeration {**  
**Disable!**;  
**Enable!})**  
 [Enable or disable mouse and keyboard input to window]

**Enumeration {**  
**Up!**;  
**Down!**;  
**Checked!**;  
**Unchecked!**;  
**Indeterminate!**;  
**Gray!}**

**:= RegionGetCheck** (NamedRgn:String)  
[Get check state of the control]

**Numeric := RegionGetChildCount** (NamedRgn:String)  
[Get count of child windows of a named region]

**String := RegionGetClass** (NamedRgn:String)  
[Get class name from a window]

**Numeric := RegionGetFocus** ()  
[Get handle of the window that currently has focus]

**Numeric := RegionGetHandle** (  
NamedRgn:String;  
[Index:Enumeration {  
Parent!;  
Self!;  
Child!}])  
[Get window handle of a named region]

**Numeric := RegionGetId** (NamedRgn:String)  
[Get control id of a named region]

**String := RegionGetListContents** (  
NamedRgn:String;  
[ItemSeparator:String])  
[Get the entire contents from a list window]

**Numeric := RegionGetListCount** (NamedRgn:String)  
[Get count of items in a listbox or combobox list]

**Numeric := RegionGetListItem** (  
NamedRgn:String;  
Item:String)  
[Retrieve item index from a list control]

**String := RegionGetListItemByIndex** (  
NamedRgn:String;  
Index:Numeric)  
[Retrieve item text by index from a list control]

**Numeric := RegionGetListSelectedCount** (NamedRgn:String)  
[Get count of selected items in a listbox or combobox list]

**Boolean := RegionGetModified** (NamedRgn:String)  
[Determine if the contents of a control have been modified]

**String := RegionGetName** (NamedRgn:String)  
[Get region name of a named region]

**Numeric := RegionGetOrder** (NamedRgn:String)  
[Get the order of a named region within its parent]

**Boolean := RegionGetPosition** (  
NamedRgn:String;  
[LeftVar:Variable];  
[TopVar:Variable];  
[WidthVar:Variable];  
[HeightVar:Variable];  
[Units:Enumeration {  
ScreenUnits!;  
DialogUnits!}])  
[Obtain position and size of a named region]

**String := RegionGetSelectedText** (  
NamedRgn:String;  
[ItemSeparator:String])  
[Get selected text from a window]

**Numeric := RegionGetStyle** (  
NamedRgn:String;  
[StyleWord:Enumeration {  
Styles!;

WindowsStyles!;  
WindowsExStyles!})  
[Get Windows style words of a named region]

**Enumeration {**  
Dialog!;  
Bitmap!;  
Button!;  
CheckBox!;  
ColorWheel!;  
ComboBox!;  
Counter!;  
Custom!;  
DateField!;  
EditField!;  
FileNameField!;  
Frame!;  
GroupBox!;  
HLine!;  
HotSpot!;  
HScrollBar!;  
Icon!;  
ListBox!;  
NameSearchListBox!;  
PopUpButton!;  
Progress!;  
PushButton!;  
RadioButton!;  
RecessedTextBox!;  
ScrollBar!;  
ShadowTextBox!;  
StaticText!;  
StaticPath!;  
Viewer!;  
VLine!;  
VScrollBar!;  
WPComboBox!;  
WPEditField!;  
WPListBox!;  
WPRecessedTextBox!;  
WPShadowTextBox!;  
WPStaticText!;  
Unknown!}

**:= RegionGetType** (NamedRgn:String)  
[Get the window type of a named region]

**String := RegionGetVariable** (NamedRgn:String)  
[Get the variable associated with a named region]

**String := RegionGetWindowText** (NamedRgn:String)  
[Get text from a window]

**Boolean := RegionIsEnabled** (NamedRgn:String)  
[Get enabled state of the named region]

**Boolean := RegionIsVisible** (NamedRgn:String)  
[Get visibility state of the named region]

**RegionMoveWindow** (  
NamedRgn:String;  
[Left:Numeric];  
[Top:Numeric];  
[Width:Numeric];  
[Height:Numeric];  
[Units:Enumeration {  
ScreenUnits!;  
DialogUnits!;  
AbsolutePosition!;  
RelativePosition!}])  
[Move the window to a new position]

**RegionRemoveListItem** (  
NamedRgn:String;  
Item:String)  
[Remove an item from a listbox or combobox list]

**RegionRemoveListItemByIndex** (  
NamedRgn:String;  
Index:Numeric)  
[Remove an item by index from a list control]

**RegionResetList** (NamedRgn:String)  
[Clear listbox contents]

**Numeric := RegionSelectListItem** (  
NamedRgn:String;  
Item:String;  
[Selection:Enumeration {  
Select!;  
Unselect!;  
Extend!}])  
[Select an item from a listbox or combobox list]

**String := RegionSelectListItemByIndex** (  
NamedRgn:String;  
Index:Numeric;  
[Selection:Enumeration {  
Select!;  
Unselect!;  
Extend!}])  
[Select an item by index from a list control]

**RegionSetBitmap** (  
NamedRgn:String;  
Filename:String;  
[Name:String])  
[Tell a bitmap control what bitmap to show]

**RegionSetCheck** (  
NamedRgn:String;  
State:Enumeration {  
Checked!;  
Unchecked!;  
Gray!;  
Indeterminate!;  
Up!;  
Down!})  
[Check or uncheck the control]

**RegionSetEditSelection** (  
NamedRgn:String;  
[Start:Numeric];  
[End:Numeric])  
[Select text in edit field]

**Numeric := RegionSetFocus** ([NamedRgn:String])  
[Set focus to specified window and return current focus window]

**RegionSetModified** (  
NamedRgn:String;  
[State:Enumeration {  
Modified!;  
NotModified!}])  
[Set or clear the 'modified' state of an edit control]

**RegionSetProgressPercent** (  
NamedRgn:String;  
Percent:Numeric)  
[Set the percentage into a progress control]

**RegionSetSelectedText** (  
NamedRgn:String;  
Item:String)  
[Set selected text of a window]

**RegionSetTabStops** (  
NamedRgn:String;  
{[Tab:Numeric]})  
[Set the tab stops in a region]

**RegionSetWindowText** (  
NamedRgn:String;  
Item:String)  
[Set text of a window]

**RegionShowWindow** (  
NamedRgn:String;  
[State:Enumeration {  
Hide!;  
Show!;  
ShowMinimized!;  
ShowMaximized!;  
Maximize!;  
ShowNoActivate!;  
Normal!;  
ShowNormal!;  
Minimize!;  
ShowMinNoActive!;  
ShowNA!;  
ShowRestore!;  
ShowDefault!}])  
[Show or hide the window]

**Enumeration** {  
Success!;  
AccessDenied!;  
TokenError!;  
InvalidKey!;  
InvalidValue!;  
NotFound!;  
KeyMissing!;  
BadIndex!;  
KeyOpenError!;  
KeyReadError!;  
KeyWriteError!;  
FileIOError!;  
KeyDeleted!;  
VolatileKey!;  
RegistryCorrupt!;  
UnknownError!}  
:= **RegistryCloseKey** (Hkey:Enumeration {  
ClassesRoot!;  
CurrentUser!;  
LocalMachine!;  
Users!;  
PerformanceData!;  
CurrentConfig!;  
DynData!})  
[Close an open key in the Windows Registry]

**Enumeration {Error!} := RegistryCreateKey** (  
Hkey:Enumeration {  
ClassesRoot!;  
CurrentUser!;  
LocalMachine!;  
Users!;  
PerformanceData!;  
CurrentConfig!;  
DynData!};  
[SubKey:String];  
[Access:Enumeration {  
ReadOnly!;  
ReadWrite!;  
WriteOnly!}])  
[Create/open a key in the Windows Registry]

```

Enumeration {
    Success!;
    AccessDenied!;
    TokenError!;
    InvalidKey!;
    InvalidValue!;
    NotFound!;
    KeyMissing!;
    BadIndex!;
    KeyOpenError!;
    KeyReadError!;
    KeyWriteError!;
    FileIOError!;
    KeyDeleted!;
    VolatileKey!;
    RegistryCorrupt!;
    UnknownError!}
:= RegistryDeleteKey (
    Hkey:Enumeration {
        ClassesRoot!;
        CurrentUser!;
        LocalMachine!;
        Users!;
        PerformanceData!;
        CurrentConfig!;
        DynData!};
    SubKey:String)
[Delete a key from the Windows Registry]

Enumeration {
    Success!;
    AccessDenied!;
    TokenError!;
    InvalidKey!;
    InvalidValue!;
    NotFound!;
    KeyMissing!;
    BadIndex!;
    KeyOpenError!;
    KeyReadError!;
    KeyWriteError!;
    FileIOError!;
    KeyDeleted!;
    VolatileKey!;
    RegistryCorrupt!;
    UnknownError!}
:= RegistryDeleteValue (
    Hkey:Enumeration {
        ClassesRoot!;
        CurrentUser!;
        LocalMachine!;
        Users!;
        PerformanceData!;
        CurrentConfig!;
        DynData!};
    [Value:String])
[Delete a value from the Windows Registry]

String := RegistryEnumKey (
    Hkey:Enumeration {
        ClassesRoot!;
        CurrentUser!;
        LocalMachine!;
        Users!;
        PerformanceData!;
        CurrentConfig!;
        DynData!};
    SubKeyNumber:Numeric)
[Enumerate a key from the Windows Registry]

```

```

String := RegistryEnumValue (
    Hkey:Enumeration {
        ClassesRoot!;
        CurrentUser!;
        LocalMachine!;
        Users!;
        PerformanceData!;
        CurrentConfig!;
        DynData!};
    ValueNumber:Numeric)
[Enumerate a value from the Windows Registry]

Enumeration {Error!} := RegistryOpenKey (
    Hkey:Enumeration {
        ClassesRoot!;
        CurrentUser!;
        LocalMachine!;
        Users!;
        PerformanceData!;
        CurrentConfig!;
        DynData!};
    [SubKey:String];
    [Access:Enumeration {
        ReadOnly!;
        ReadWrite!;
        WriteOnly!}])
[Open a key in the Windows Registry]

Numeric := RegistryQueryKeyCount (Hkey:Enumeration {
    ClassesRoot!;
    CurrentUser!;
    LocalMachine!;
    Users!;
    PerformanceData!;
    CurrentConfig!;
    DynData!})
[Get count of subkeys under a key in the Windows Registry]

Enumeration {
    Success!;
    AccessDenied!;
    TokenError!;
    InvalidKey!;
    InvalidValue!;
    NotFound!;
    KeyMissing!;
    BadIndex!;
    KeyOpenError!;
    KeyReadError!;
    KeyWriteError!;
    FileIOError!;
    KeyDeleted!;
    VolatileKey!;
    RegistryCorrupt!;
    UnknownError!}
:= RegistryQueryLastError ()
[Get the error value of the last Registry function]

Any := RegistryQueryValue (
    Hkey:Enumeration {
        ClassesRoot!;
        CurrentUser!;
        LocalMachine!;
        Users!;
        PerformanceData!;
        CurrentConfig!;
        DynData!};
    [Value:String];
    [Type:Variable])
[Obtain a value from the Windows Registry]

```

**Numeric := RegistryQueryValueCount** (Hkey:Enumeration {  
ClassesRoot!;  
CurrentUser!;  
LocalMachine!;  
Users!;  
PerformanceData!;  
CurrentConfig!;  
DynData!})

[Get count of values under a key in the Windows Registry]

**Enumeration** {  
Success!;  
AccessDenied!;  
TokenError!;  
InvalidKey!;  
InvalidValue!;  
NotFound!;  
KeyMissing!;  
BadIndex!;  
KeyOpenError!;  
KeyReadError!;  
KeyWriteError!;  
FileIOError!;  
KeyDeleted!;  
VolatileKey!;  
RegistryCorrupt!;  
UnknownError!}

**:= RegistrySetValue** (  
Hkey:Enumeration {  
ClassesRoot!;  
CurrentUser!;  
LocalMachine!;  
Users!;  
PerformanceData!;  
CurrentConfig!;  
DynData!};  
[Value:String];  
Data:Any;  
Type:Enumeration {  
Binary!;  
DWord!;  
String!;  
ExpandableString!;  
PFBinary!;  
PFWord!;  
PFString!;  
PFExpandableString!;  
PFWPString!;  
PFByte!;  
PFWord!;  
PFBool!;  
PFLogical!})

[Save a value in the Windows Registry]

**Boolean := RenameDirectory** (  
OldDirectoryName:String;  
NewDirectoryName:String;  
[Prompts:Enumeration {  
NoPrompts!;  
Prompts!}])

[Rename a directory (folder)]

**Boolean := RenameFile** (  
OldFilename:String;  
NewFilename:String;  
[Prompts:Enumeration {  
NoPrompts!;  
Prompts!}])

[Rename or move a file]

**Repeat**

[Repeat statements until a condition is TRUE]

**Return** (  
<Condition>Enumeration {  
CancelCondition!;  
ErrorCondition!;  
NotFoundCondition!};  
[<Value>Any])

[Return to place of call in macro code, and assert a condition]

**Return** ([<Value>Any])

[Return to place of call in macro code]

**Any := RoundOff** (  
Value:Any;

[RoundTo:Any])

[Return a value rounded to the nearest multiple of another value]

**Run** (MacroFile:String;{[Parameter:Any]})

[Execute the macro code found in the specified macro file]

- S -

**SendKeys** (  
KeyCode:String;

[MarkupLanguage:Enumeration {  
Old!;  
New!}])

[Send the specified keystrokes to the current application]

**Boolean := SetCurrentDirectory** (DirectoryName:String)

[Change current directory (folder)]

**Numeric := SetDefaultParent** ([Parent:Any {  
ResetToOriginal!;  
NoChange!}])

[Set the parent window used for macro dialogs, Prompt, MessageBox, GetString, GetNumber, and GetUnits]

**Boolean := SetFileAttributes** (  
Filename:String;

[Attributes:Enumeration {  
Normal!;  
ReadOnly!;  
Hidden!;  
System!;  
Label!;  
Directory!;  
Archive!}];  
[Prompts:Enumeration {  
NoPrompts!;  
Prompts!}])

[Change file attributes]

**Boolean := SetFileDateAndTime** (  
Filename:String;

[DateAndTime:Date/Time];  
[DateType:Enumeration {  
CreationDateTime!;  
LastAccessDateTime!;  
ModificationDateTime!}])

[Change file date and time]

**Numeric := Sign** (Value:Numeric)

[Return the sign of a value (-1 if < 0, 0 if = 0, 1 if > 0)]

**Numeric := sin** (Angle:Numeric)

[Return the sine of an angle in radians]

**Numeric := Single** (<Value>Numeric)

[Use only in OLE Automation call in-line parameter function; pass value as 4 byte float]

**Numeric := sinh** (Angle:Numeric)

[Return the hyperbolic sine of an angle in radians]

**Numeric := SizeOf** ([Value:Any])  
[Return the size required to store the value]

**Numeric := Speed** ([TenthsOfSeconds:Numeric])  
[Wait the specified amount of time between macro statements]

**Numeric := SquareRoot** (Value:Numeric)  
[Return the square root of a value]

**Numeric := StdDev** (  
  [Type:Enumeration {  
    Population!;  
    Sample!};]  
  {Value:Numeric})  
[Return the standard deviation of a list of values]

**Step** (State:Enumeration {Off!;On!})  
[Single Step macro statements]

**String := StrFill** (Count:Numeric;[String:String])  
[Fill a string multiple times]

**Numeric := StrFraction** (String:String)  
[Return a string containing a fraction converted to a numeric value]

**String := String** (<Value>String)  
[Use only in DLLCall or OLE Automation call in-line parameter function; pass value as ANSI string]

**String := StrInsert** (  
  String:String;  
  [SubString:String];  
  [Beginning:Numeric];  
  [NumberOfChars:Numeric])  
[Insert a substring into a string, replace characters in a string with a substring, or remove characters from a string]

**Boolean := StrIsChar** (  
  String:String;  
  [Position:Numeric];  
  [Option:Enumeration {  
    EqualTo!;  
    NotEqualTo!});]  
  [CharSet:Any {  
    Alphabetic!;  
    AlphaNumeric!;  
    Numeric!;  
    Punctuation!;  
    WhiteSpace!;  
    UpperCase!;  
    LowerCase!})]  
[Determine if string or character is of a particular type]

**String := StrLeft** (String:String;[Length:Numeric])  
[Return the left part of a string]

**Numeric := StrLen** (String:String)  
[Return the character length of a string]

**String := StrMakeList** (  
  [Separator:String];  
  {[SubString:String]})  
[Join array of substrings into string list separated by characters]

**Numeric := StrNum** (String:String)  
[Return a string converted to a numeric value]

**String := StrPad** (  
  String:String;  
  Length:Numeric;  
  [Option:Enumeration {  
    PadRight!;  
    PadLeft!};]

  PadEnds!;  
  PadWords!});]  
  [PadString:String])  
[Pad a string with another string to a length]

**Array := StrParseList** (  
  String:String;  
  [Separators:Any {  
    Punctuation!;  
    WhiteSpace!};]  
  [Option:Enumeration {  
    Characters!;  
    Strings!})]  
[Parse string list separated by characters into array of substrings]

**Numeric := StrPos** (  
  String:String;  
  SubString:String;  
  [Beginning:Numeric])  
[Return the beginning character position where a substring begins in a string]

**String := StrReverse** (String:String)  
[Reverse all characters in a string]

**String := StrRight** (  
  String:String;  
  [Length:Numeric])  
[Return the right part of a string]

**Numeric := StrScan** (  
  String:String;  
  [Beginning:Numeric];  
  [Option:Enumeration {  
    EqualTo!;  
    NotEqualTo!});]  
  [CharSet:Any {  
    Alphabetic!;  
    AlphaNumeric!;  
    Numeric!;  
    Punctuation!;  
    WhiteSpace!;  
    UpperCase!;  
    LowerCase!})]  
[Return index of first matching or non-matching character]

**String := StrToChars** (  
  String:String;  
  [Option:Enumeration {  
    Keep!;  
    Remove!});]  
  [CharSet:Any {  
    Alphabetic!;  
    AlphaNumeric!;  
    Numeric!;  
    Punctuation!;  
    WhiteSpace!;  
    UpperCase!;  
    LowerCase!})]  
[Remove or keep specified characters in a string]

**String := StrTransform** (  
  String:String;  
  FromChars:String;  
  [ToChars:String];  
  [Options:Enumeration {  
    Characters!;  
    Strings!;  
    All!;  
    FirstOnly!})]  
[Transform characters in a string to other characters]



**String := StrTrim** (  
 String:String;  
 [Length:Numeric];  
 [Option:Enumeration {  
 TrimRight!;  
 TrimLeft!;  
 TrimEnds!;  
 TrimWords!}]);  
 [TrimChars:Any {  
 Alphabetic!;  
 AlphaNumeric!;  
 Numeric!;  
 Punctuation!;  
 WhiteSpace!;  
 UpperCase!;  
 LowerCase!}])  
 [Trim characters from a string]

**Raw Binary Data := Structure** (<Value>Any)  
 [Use only in DLLCall or OLE Automation call in-line parameter function; pass value as a C struct]

**Numeric := StrUnit** (String:String)  
 [Return a string converted to a units value]

**String := SubChar** (  
 String:String;  
 Beginning:Numeric;  
 [NumberOfChars:Numeric])  
 [Return the specified number of characters from the specified beginning character position in a string]

**String := SubStr** (  
 String:String;  
 Beginning:Numeric;  
 [NumberOfChars:Numeric])  
 [Return the specified number of characters from the specified beginning character position in a string]

**Numeric := Sum** ({Value:Numeric})  
 [Return the sum of a list of values]

**Switch** (<Test>Any)  
 [Execute code depending on a value]

- T -

**Numeric := tan** (Angle:Numeric)  
 [Return the tangent of an angle in radians]

**Numeric := tanh** (Angle:Numeric)  
 [Return the hyperbolic tangent of an angle in radians]

**Numeric := TimeHour** ([Time:Date/Time])  
 [A value representing the hour on a 24-hour clock]

**Numeric := TimeHundredth** ([Time:Date/Time])  
 [A value representing the hundredths of a second]

**Numeric := TimeMinute** ([Time:Date/Time])  
 [A value representing the minute]

**Date/Time := TimePart** ([Time:Date/Time])  
 [Return the time part of a date and time value]

**Numeric := TimeSecond** ([Time:Date/Time])  
 [A value representing the second]

**String := TimeString** (  
 [Time:Date/Time];  
 [Format:String])  
 [A string representing the time]

**Any** {  
 UnknownTime!;  
 StandardTime!;  
 DaylightSavingTime!}  
 := **TimeZoneInfo** (  
 Infoltem:Enumeration {  
 TimeZoneNumber!;  
 TimeZoneName!;  
 TimeZoneUTCOffset!;  
 TimeChangeoverStartDate!;  
 TimeChangeoverOffset!;  
 TimeChangeoverType!;  
 IsStandardTime!;  
 IsDaylightSavingTime!};  
 [TimeChangeoverType:Enumeration {  
 CurrentTime!;  
 StandardTime!;  
 DaylightSavingTime!}])  
 [Information about time the current time zone and time changeover]

**String := ToInitialCaps** (  
 String:String;  
 [Style:Enumeration {  
 FirstCharOnly!;  
 FirstOfEachWord!}])  
 [Return a string with first characters capitalized]

**String := ToLower** (String:String)  
 [Return the lowercase equivalent of a string]

**String := ToUpper** (String:String)  
 [Return the uppercase equivalent of a string]

**Boolean := True**  
 [The constant boolean value 'TRUE']

- U -

**String := UnitStr** (  
 Value:Any;  
 Units:Enumeration {  
 Centimeters!;  
 Inches!;  
 Millimeters!;  
 Points!;  
 WPUnts!;  
 WP1200ths!})  
 [Return a units value converted to a string]

**Until** (<Test>Boolean)  
 [Specify the condition to terminate the execution of a group of statements]

**Use** (<MacroFile>String)  
 [Specify the name of a macro library file]

- V -

**Any := Value** (<Value>Any)  
 [Return the value of an enumeration or token]

**Any** {  
 Boolean!;  
 WPString!;  
 AnsiString!;  
 OemString!;  
 Float!;  
 Integer!;  
 DateTime!;  
 Centimeters!;  
 Inches!;  
 Millimeters!;  
 Points!;

```

WPUnits!;
RawBinary!;
Object!}
:= ValueType (
Value:Any;
[Type:Enumeration {
CurrentType!;
Boolean!;
String!;
WPString!;
AnsiString!;
OemString!;
Numeric!;
Number!;
Float!;
Integer!;
DateTime!;
Measurement!;
Centimeters!;
Inches!;
Millimeters!;
Points!;
WPUnits!;
RawBinary!;
Object!}])

```

[Return or test the data type of a variable or value]

```

Enumeration {Off!;On!} := VarErrChk ([State:Enumeration {
Off!;
On!}])

```

[Ignore or generate errors on undeclared variables]

```

Numeric := Variance (
[Type:Enumeration {
Population!;
Sample!}];
{Value:Numeric})

```

[Return the variance of a list of values]

```

Any := Variant (<Value>Any)

```

[Use only in OLE Automation call in-line parameter function; pass value as a VARIANT]

```

Any := VarLocate (
Pool:Enumeration {
Local!;
Global!;
Persistent!};
[Index:Numeric])

```

[Returns the name of a variable by index]

```

Any { } := VersionInfo (
ObjectItem:Any {
PlatformVersion!;
PerfectFitVersion!;
PerfectScriptVersion!;
WordPerfectVersion!;
PresentationsVersion!;
QuattroProVersion!};
[InfoItem:Enumeration {
FileVersionString!;

```

```

FileVersionNumber!;
FileMajorVersion!;
FileMinorVersion!;
FileBuildNumber!;
FileSubBuildNumber!;
ProductTitle!;
ProductTitleInfo!;
ProductVersionString!;
ProductVersionNumber!;
ProductMajorVersion!;
ProductMinorVersion!;
ProductBuildNumber!;
ProductSubBuildNumber!;
ProductSPMajorVersion!;
ProductSPMinorVersion!;
ProductSuiteMask!;
ProductTypeMask!;
CompanyName!;
ProductName!;
InternalName!;
LegalCopyright!;
OriginalFilename!;
FileDescription!}])

```

[Return version information]

**- W -**

```

Wait (TenthsOfSeconds:Numeric)

```

[Stop the macro for the specified amount of time]

```

While (<Test>Boolean)

```

[Repeat a group of statements while a condition is TRUE]

```

With (<ObjectVariable>Variable)

```

[Specify default OLE object to use for non-prefixed object access]

```

Numeric := Word (<Value>Numeric)

```

[Use only in DLLCall or OLE Automation call in-line parameter function; pass value as unsigned WORD]

```

Numeric := WordCount (
String:String;
[WordLength:Numeric];
[SeparatorChars:String];
[Option:Enumeration {
CountWords!;
CountShorter!;
CountLonger!;
ShortestLength!;
LongestLength!;
AverageLength!}])

```

[Count words in a string]

```

String := WPString (<Value>String)

```

[Use only in DLLCall or OLE Automation call in-line parameter function; pass value as WP string]

**- X -**

```

Boolean := Boolean XOR Boolean

```

[Returns BOOLEAN XOR of two BOOLEANS; Precedence level 9]

[Go To Top](#) [Go To PerfectScript Index](#) [Go To WordPerfect Product Command Index](#)