

Word Pro: Abandon method

```
{button ,AL(^H_BASECONTAINER_CLASS;H_CELLCONTAINER_CLASS;H_FRAMECONTAINER_CLASS;H_NOTECONTAINER_CLASS;H_RUBYCONTAINER_CLASS;H_DROPDOWNCONTAINER_CLASS;H_PAGECONTAINER_CLASS;H_SUBPAGECONTAINER_CLASS;H_SUPERPAGECONTAINER_CLASS;H_ROWCONTAINER_CLASS;H_SUPERTABLECONTAINER_CLASS;H_TABLECONTAINER_CLASS;H_PARALLELCOLSCONTAINER_CLASS;H_TABLEONLYCONT_CLASS',0)} See list of classes
```

Converts the child container of a non-page container to the child container of the current page container.

Syntax

```
[objectreference].Abandon()
```

Parameters

None.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

An example of how to use this method is seen in the following scenario:

On a page is a parent frame that contains a child frame. When you move the parent frame across the page, the child frame anchored to the parent frame moves with it. You now want to detach the child frame from the parent frame and anchor it to the current page, so that the child frame moves across the page by itself. To do this, set the Abandon method to convert the child frame from a child of its parent frame to a child of the current page container.

Word Pro: Activate method

{button ,AL('H_DOCUMENT_CLASS;H_TEXTDOCUMENT_CLASS;H_SECTIONTABS_CLASS',0)} [See list of classes](#)

{button ,AL('H_ACTIVATE_METHOD_EXSCRIPT',1)} [See example](#)

[Document]

Use this method from a TextDocument object to make that object active.

[SectionTABS]

Causes an OLE object to become active in the Word Pro application.

Note OLE is not supported under OS/2.

Syntax

[objectreference].Activate()

Parameters

Return value

Returns an Integer value indicating success (True) or failure (False). Return value for this method is always -1 or 0. When testing the return value, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

[SectionTABS]

Use this method to activate an OLE division you created in the document.

Word Pro: AddAccelerators method

{button ,AL('H_ACCELERATORS_CLASS',0)} [See list of classes](#)

{button ,AL('H_ADDACCELERATORS_METHOD_EXSCRIPT',1)} [See example](#)

Adds accelerator key commands that implement Word Pro functions and commands.

Syntax

[objectreference].AddAccelerators(MacroName, Key, [Id],[IsTemporaryUse])

Parameters

Macroname

A String expression representing the name of the macro, including the file name, that should be assigned to a keystroke.

Key

A Numeric expression representing the specific key used as the shortcut key combination. Data type is Integer.

Id

A Numeric expression representing the identification number of the menu item to which you want to add the accelerator object. Data type is Integer. Default is 0.

IsTemporaryUse

A Boolean expression indicating whether this accelerator will persist between sessions of Word Pro (False) or will be discarded when the current session of Word Pro is terminated (True). Default is False.

Return value

Integer.

Usage

Word Pro: AddBookmark method

{button ,AL('H_BOOKMARKMANAGER_CLASS',0)} [See list of classes](#)

{button ,AL('H_ADDBOOKMARK_METHOD_EXSCRIPT',1)} [See example](#)

Adds a bookmark object to the document.

Syntax

[objectreference].AddBookmark(Name, MarkerName)

Parameters

Name

A String expression representing the bookmark object you want to add. If you attempt to create a bookmark using a duplicate name, a number is automatically added to the end of the Name parameter to make the bookmark name unique.

MarkerName

The String name of the bookmark object marker. You must create a marker before you create a bookmark. You can name the bookmark yourself or use the name provided by Word Pro.

Return value

The return value for this method will always be -1 or 0. When testing the return value, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

When adding a bookmark, you must first create a marker of type bookmark. Use the name of that marker object in the MarkerName parameter. After the bookmark is created, the AddBookmark method notifies the BookmarkManager about the newly created bookmark.

Attempting to use this method without referring to a valid marker in the MarkerName parameter causes unpredictable results.

Word Pro: AddChildToLayout method

```
{button ,AL('H_LAYOUT_CLASS;H_CELLGROUPLAYOUT_CLASS;H_CELLLAYOUT_CLASS;H_CONNECTEDLAYOUT_CLASS;H_COLUMNGROUPLAYOUT_CLASS;H_FOOTERLAYOUT_CLASS;H_FRAMELAYOUT_CLASS;H_FRAMEGROUPLAYOUT_CLASS;H_DROPDOWNLAYOUT_CLASS;H_GROUPLAYOUT_CLASS;H_HEADERLAYOUT_CLASS;H_NOTELAYOUT_CLASS;H_PAGELAYOUT_CLASS;H_ROWGROUPLAYOUT_CLASS;H_ROWLAYOUT_CLASS;H_RUBYLAYOUT_CLASS;H_SUPERTABLEGROUPLAYOUT_CLASS;H_SUPERTABLELAYOUT_CLASS;H_TABLELAYOUT_CLASS;H_ENDNOTELAYOUT_CLASS;H_FOOTNOTELAYOUT_CLASS;H_TABLEHEADINGLAYOUT_CLASS;H_TOCSUPERTABLELAYOUT_CLASS',0)} See list of classes
```

Assigns a parent layout object a child layout object.

Syntax

```
[objectreference].AddChildToLayout(ChildName)
```

Parameters

ChildName

A String expression that represents the name of child object you want to add to the layout.

Return value

The return value for this method will always be -1 or 0. When testing the return value, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Word Pro: AddDdeLink method

{button ,AL('H_DDELINKMANAGER_CLASS',0)} [See list of classes](#)

Adds a Dde link object to the document. Word Pro creates a Dde link if you use Paste Special, but only if OLE fails or is not available.

Note OLE is not supported under OS/2.

Syntax

```
[objectreference].AddDdeLink(ConversationHandle,LinkInfo,MarkerName,ServerName,ClipbrdFormatName,UpdateDataOnly)
```

Parameters

ConversationHandle

A Long expression that was added manually, representing the conversation handle used by this method. If added using LotusScript, this value is always 0. If added internally, the value may be non-0.

LinkInfo

A String expression representing the link information about the Dde link object you want to add. Consists of the server name, the topic name, and the item name.

MarkerName

A String expression representing the name of the Dde marker object. You must create a marker before you create a Ddelink. You can name the marker yourself or use the name provided by Word Pro.

ServerName

A String expression representing the executable name of the server to which you want to link.

TopicName

A String expression representing the name of the drive, directory, and name for the file that contains the data or the object name of the data.

ItemName

A String expression representing the location or name for the data, such as a range of cells, a named spreadsheet range, or a bookmark name.

ClipbrdFormatName

A String expression representing the name of the format used by the Clipboard. The format will be used to read/interpret/import the data.

UpdateDataOnly

A Boolean expression that specifies whether the new DDE link will update only data or data and formatting information. Data type is Integer. The legal values for this parameter are -1 and 0 but you may use the LotusScript constants True (-1) and False (0). If the value is True, the formatting is controlled by Word Pro.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Word Pro always attempts to utilize OLE linking before DDE linking is used.

Word Pro: AddDivisionToPrint method

{button ,AL(`H_PRINTSETTINGS_CLASS',0)} [See list of classes](#)

{button ,AL(`H_ADDDIVISIONTOPRINT_METHOD_EXSCRIPT',1)} [See example](#)

Allows you to specify the name of a division object that you want to print.

Syntax

[objectreference].AddDivisionToPrint(DivisionToPrint)

Parameters

DivisionToPrint

A String expression that allows you to print a specified division object.

Return value

Integer.

Usage

Adds a division object to a list of divisions. You can locate a list of divisions by choosing File - Print, clicking Select Pages, and selecting "Whole divisions" in the Select Pages dialog box.

{button ,AL(`H_CLEARDIVISIONLIST_METHOD_MEMDEF',0)} [See related topics](#)

Word Pro: AddDivision method

{button ,AL('H_DIVISION_CLASS;H_TEXTDOCUMENT_CLASS',0)} [See list of classes](#)

{button ,AL('H_ADDDIVISION_METHOD_EXSCRIPT',1)} [See example](#)

Adds a division object to a division or text document object. Equivalent to choosing Create - Division.

Syntax

Division

[objectreference].AddDivision(NewName, [ParentName,] [BeforeNeighbor,][NeighborName])

TextDocument

[objectreference].AddDivision(NewName, [ParentName,] [BeforeNeighbor,][NeighborName])

Parameters

NewName

A String expression that represents the name of the new division object.

ParentName

A String expression representing the name of the parent division object.

BeforeNeighbor

Data type is Integer. The legal values for this parameter are -1 and 0 but you may use the LotusScript constants True (-1) and False (0). This parameter allows you to specify whether you want to locate the new division object before its neighbor division object. Default is False (0).

NeighborName

A String expression representing the name of a neighbor division object.

Return value

Usage

Word Pro: AddEditorManager method

{button ,AL('H_EDITORMANAGER_CLASS',0)} [See list of classes](#)

{button ,AL('H_ADDEDITORMANAGER_METHOD_EXSCRIPT',1)} [See example](#)

Adds a new editor to a document.

Syntax

[objectreference].AddEditorManager(EditorName, EditorInitials)

Parameters

EditorName

A String expression representing the name of the assigned editor.

EditorInitials

A String expression representing the initials of the assigned editor.

Return value

The return value for this method is always -1 or 0. When testing the return value, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

This method returns True if Word Pro adds a new editor to the document. This method returns False if Word Pro does not add a new editor to the document, or if the specific editor name already exists.

Usage

Adds a new editor to a document and assigns the editor default editing rights, as defined by the special "All others" editor.

Word Pro: AddField method

{button ,AL('H_DOCINFOFIELDMANAGER_CLASS',0)} [See list of classes](#)

{button ,AL('H_ADDFIELD_METHOD_EXSCRIPT',1)} [See example](#)

Creates a document field object in a document.

Syntax

[objectreference].AddField(FieldName,Contents, ExportFieldToNotes)

Parameters

FieldName

A String expression representing the name of the document field you want to add.

Contents

A String expression representing the contents that will be contained in the document field you want to add.

ExportFieldToNotes

A Boolean expression specifying whether or not you want the field exported to Notes. A Boolean expression is either True or False.

Return value

The return value for this method is always -1 or 0. When testing the return value, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Equivalent to choosing File - Document Properties, choosing Document, clicking the Fields tab, and clicking New.

Creating a new document field object as part of document information can be useful when you want to track specific information. For example, you can develop a system to track documents with specific clients. First, you can create a document field called "Client" for the document and assign a client name as its contents. You can then insert the document field into your text stream, so that the client's name appears in the text of the document.

Word Pro: AddIcon method

{button ,AL('H_ICONBAR_CLASS',0)} [See list of classes](#)

{button ,AL('H_ADDICON_METHOD_EXSCRIPT',1)} [See example](#)

Adds an icon to an icon bar object.

Syntax

[objectreference].AddIcon(Position)

Parameters*Position*

Data type is Integer. Parameter is the user-defined location on the icon on the bar. Required parameter.

Return value

The return value for this method will always be -1 or 0. When testing the return value, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Adds an icon to an existing icon pallet. The position parameter is a base 0; thus, the first position is position 0, the second position is position 1, and so on.

To add an icon, you must first select it using the SelectCustomIcon or SelectStandardIcon method in the IconBarManager class. Equivalent to the interface is in the SmartIcons Setup dialog box, where you can drag an icon from the available list and drop it on the icon bar set.

Word Pro: AddIndexEntry method

{button ,AL('H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL('H_ADDINDEXENTRY_METHOD_EXSCRIPT',1)} [See example](#)

Inserts an Index power field at the insertion point.

Syntax

[objectreference].AddIndexEntry(IndexEntry)

Parameters

IndexEntry

A String expression representing the name of the entry you want to add to the index.

Return value

Usage

Equivalent to choosing Text - Mark Text As - Index Entry, selecting the desired text, and clicking Mark. For information on formatting the powerfield, see the "Index" power field help topic in Word Pro help.

Word Pro: AddLayoutOverride method

{button ,AL(^H_LAYOUTOVERRIDE_CLASS',0)} [See list of classes](#)

Adds a layout to a DivisionInfo object's layout override list.

Syntax

[objectreference].AddLayoutOverride(LayoutObjectName)

Parameters

LayoutObjectName

A String expression representing the name of the layout object you want to add to the layout override list.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

For more information about layout override list, see the [Word Pro: LayoutOverride class](#) topic.

Word Pro: AddOutlineSequenceItem method

{button ,AL('H_OUTLINESTYLESEQUENCE_CLASS',0)} [See list of classes](#)

{button ,AL('H_ADDOUTLINESEQUENCEITEM_METHOD_EXSCRIPT',1)} [See example](#)

Adds an item to an outline style sequence object.

Syntax

[objectreference].AddOutlineSequenceItem(StyleName)

Parameters

StyleName

A String expression representing the style name of the outline sequence item that you want to add to an outline style sequence object.

Return value

Usage

Word Pro: AddPopupGraphicItem method

{button ,AL(^H_STATUSBARBUTTON_CLASS',0)} [See list of classes](#)

Called when responding to the StatusBarButtonFillPopupList event. Adds the graphic into the popup list.

Syntax

[objectreference].AddPopupGraphicItem([BitmapHandle])

Parameters

BitmapHandle

The handle to the bitmap that is to be displayed in the popup list.

Return value

True if the item is added; False if the item is not added.

Usage

Can only be used if the button is of type Graphic. The first item is at the top of the list.

Word Pro: AddPopupPointSizeItem method

{button ,AL('H_STATUSBARBUTTON_CLASS',0)} [See list of classes](#)

Inserts an item in the popup list on the point size status bar button object. This method is called when responding to the StatusBarButtonFillPopupList event.

Syntax

[objectreference].AddPopupPointSizeItem(PointSize)

Parameters

PointSize

Indicates the value which should appear in the list of point sizes. Data type is [Single](#), which lets you specify the value in points and fractions of points.

Return value

Integer.

Usage

Call this method once for each item you want to add to the list of point sizes. The first item is at the top of the list.

Word Pro: AddPopupTextItem method

{button ,AL(^H_STATUSBARBUTTON_CLASS',0)} [See list of classes](#)

Inserts an item in a popup list on the text status bar button object. This method is called when responding to the StatusBarButtonFillPopupList event. This method adds the text in the popup list.

Syntax

[objectreference].AddPopupTextItem(Text, [BitmapHandle])

Parameters

Text

Indicates the text to be displayed in the popup list.

BitmapHandle

This optional Long parameter represents the handle to the bitmap that is to be displayed in the popup list. Only valid if the button is of type TextAndGraphic. Otherwise, it should be 0.

Return value

Integer

Usage

May only be used if the button is of type Text, or TextAndGraphic. The first item is at the top of the list.

Word Pro: AddSectionTabs method

{button ,AL('H_SECTIONTABS_CLASS',0)} [See list of classes](#)

{button ,AL('H_ADDSECTIONTABS_METHOD_EXSCRIPT',1)} [See example](#)

Creates a new division immediately following the division in which this method is called.

Syntax

[objectreference].AddSectionTabs()

Parameters

None.

Return value**Usage**

Equivalent to clicking the right mouse button on an existing division divider tab and choosing Quick Division.

Word Pro: AddSmartCorrect method

{button ,AL('H_SMARTCORRECT_CLASS',0)} [See list of classes](#)

{button ,AL('H_ADDSMARTCORRECT_METHOD_EXSCRIPT',1)} [See example](#)

Adds a SmartCorrect entry to the specified SmartCorrect object.

Syntax

[objectreference].AddSmartCorrect(Entry,Replacement)

Parameters

Entry

A String expression representing the entry you want to add to the SmartCorrect tool.

Replacement

A String expression representing the text you want to use to replace a SmartCorrect entry.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Word Pro: AddTOCEntry method

{button ,AL('H_WPAPPLICATION_CLASS;H_TOCSUPERTABLELAYOUT_CLASS',0)} [See list of classes](#)

{button ,AL('H_ADDTOCENTRY_METHOD_EXSCRIPT',1)} [See example](#)

Adds a table of contents entry.

Syntax

[objectreference].WPApplication.AddTOCEntry(TOCEntry)

[objectreference].TOCSuperTableLayout.AddTOCEntry()

Parameters

TOCEntry

Data type is String.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

This method is equivalent to marking text as a TOC entry.

Word Pro: AddVerbMenu method

{button ,AL('H_OLEOBJECT_CLASS',0)} [See list of classes](#)

Adds a list of available verbs that an OLE object can support.

Note This method is not implemented for OLEObject within OS/2.

Syntax

[objectreference].AddVerbMenu(MenuHandle,OleVerbMin,OleVerbMax,OleVerbConvert, [ForceAddPopup])

Parameters

MenuHandle

A Numeric expression representing the menu handle used by the AddVerbMenu method. Data type is Long as the numeric expression.

OleVerbMin

Data type is Integer.

OleVerbMax

Data type is Integer.

OleVerbConvert

Data type is Integer.

Return value

Integer.

Usage

Word Pro: Add method

{button ,AL('H_CLICKHERE_CLASS;H_TEXTMARKER_CLASS;H_TEXT_CLASS',0)} [See list of classes](#)

{button ,AL('H_ADD_METHOD_EXSCRIPT',1)} [See example](#)

Adds the selected word to the currently active user dictionary.

Syntax

[objectreference].Add(AddType)

Parameters

AddTypeSpell

Tells Word Pro that you are adding a word to the user dictionary. Data type is Variant, which allows the value of this parameter to be either a number or a constant that produces that number. There is no default value. You must include the constant \$LwpAddTypeSpell or its numeric equivalent of 4.

Return value

Integer.

Usage

If more than one user dictionary is active, Word Pro adds the word to the first dictionary listed in the Spell Check Options dialog box.

- If more than one word is selected, only the word at the beginning of the selection is added to the dictionary.
- If the word ends with a paragraph or other marker, Word Pro does not add the word to the dictionary.
- If no word is selected, the word at the insertion point is added to the dictionary.
- If the insertion point is at the end of a word, that word is added to the dictionary.
- If the insertion point is at the beginning of a word, that word is added to the dictionary.
- If the insertion point is between two spaces, no word is added to the dictionary.

Word Pro: AdjustShade method

{button ,AL('H_CLICKHERE_CLASS;H_TEXTMARKER_CLASS;H_TEXT_CLASS',0)} [See list of classes](#)

{button ,AL('H_ADJUSTSHADE_METHOD_EXSCRIPT',1)} [See example](#)

Changes the size of a text selection. The selected text can be in a Text, a TextMarker, or a ClickHere object.

Syntax

[objectreference].AdjustShade(WhichSide, Count, AdjustUnit, MarkerName)

Parameters

WhichSide

Specifies which side of the selection you are adjusting. Data type is Variant, which allows the value of this parameter to be one of the constants listed below or its numeric equivalent (in parentheses). There is no default value.

\$LwpWhichSideLeft (2068)

\$LwpWhichSideRight (2069)

\$LwpWhichTypeLeft (1989)

\$LwpWhichTypeRight (1990)

Count

An Integer expression which specifies how many units (specified in the AdjustUnit parameter) will be added or removed from the selection. To remove units, use positive integers. To add units, use negative integers.

AdjustUnit

Specifies the type of unit you can use to increment your adjustment. Specify the number of units in the Count parameter. Data type is Variant, which allows the value of this parameter to be one of the constants listed below or its numeric equivalent (in parentheses). There is no default value.

<u>Value</u>	<u>Effect</u>
\$LwpAdjustUnitCharacter (2246)	Selects or deselects the specified number of characters.
\$LwpAdjustUnitWord (2247)	Selects or deselects the specified number of words. A word is comprised of a contiguous string of alphanumeric characters. Punctuation and spaces are seen as the end of a word.
\$LwpAdjustUnitChunk (2248)	Selects or deselects the specified number of chunks. A chunk is comprised of a single word (a group of characters with no spaces) and all the contiguous spaces following that word.
\$LwpAdjustUnitSentence (2249)	Selects or deselects the specified number of sentences. A sentence is comprised of a stream of text marked on either side by either a period or a paragraph marker.
\$LwpAdjustUnitObject (2250)	Selects or deselects the specified number of objects.
\$LwpAdjustUnitParagraph (2251)	Selects or deselects the specified number of paragraphs.
\$LwpAdjustUnitMarker (2252)	Moves the specified side of the selection to the marker object named in the MarkerName parameter. If you use this value, you must use 1 for the value of the Count parameter.

MarkerName

A String expression which specifies the name of the marker object to which you want to move part of your selection. Use this parameter only when you use \$LwpAdjustUnitMarker as the value for the AdjustUnit parameter.

Return value

The return value for this method will always be -1 or 0. When testing the return value, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Word Pro sees the sides of a selection as the sides of an expandable blanket. You can stretch or shrink a blanket to fit an area and you can pick up the left side of a blanket and pull it over the right side so the blanket covers an entirely new area. The same is true of a selection in Word Pro. You can change the coverage of your text selection using the left and right sides and marker objects.

For example, if you have a marker named "MarkerOne" and you tell Word Pro to adjust the left side of the selection to match that marker, Word Pro moves the left side of the selection to the MarkerOne position. Word Pro changes the selection in one of three ways:

- If MarkerOne is located before the selection, Word Pro expands the selection to include the text between the original left side and MarkerOne.
- If MarkerOne is located between the original left and right sides of the selection, the selection is reduced to exclude the text between the original left side and MarkerOne.
- If MarkerOne is located after the selection, Word Pro moves the left side to MarkerOne so that the original selection is entirely excluded, and everything between the original right side and MarkerOne becomes selected.

Word Pro: Adopt method

```
{button ,AL(^H_BASECONTAINER_CLASS;H_CELLCONTAINER_CLASS;H_FRAMECONTAINER_CLASS;H_NOTE  
CONTAINER_CLASS;H_RUBYCONTAINER_CLASS;H_DROPDOWNCONTAINER_CLASS;H_PAGECONTAINER_  
CLASS;H_SUBPAGECONTAINER_CLASS;H_SUPERPAGECONTAINER_CLASS;H_ROWCONTAINER_CLASS;  
H_SUPERTABLECONTAINER_CLASS;H_TABLECONTAINER_CLASS;H_PARALLELCOLSCONTAINER_CLASS  
;H_TABLEONLYCONT_CLASS',0)} See list of classes
```

Converts a specific child container of a current page container to a child container of a specific parent container within the page.

Syntax

[objectreference].Adopt()

Parameters

Return value

Usage

For example, how to use this method is seen when a page container has a child frame anchored to it. Because the child frame is anchored to the page, it can be moved across the page. If you want to detach the child frame from the page and anchor it to a parent frame container so that the child frame moves within the frame, set this method.

Equivalent to choosing Frame Properties, clicking the Placement tab, and selecting "In Frame" from the "Place frame" box. Note that the Frame menu displays when the insertion point is in a frame.

Word Pro: AdviseOnRename method

{button ,AL('H_TEXTDOCUMENT_CLASS',0)} [See list of classes](#)

Syntax

[objectreference].AdviseOnRename()

Parameters

Return value

Usage

Word Pro: AdviseOnSave method

{button ,AL('H_TEXTDOCUMENT_CLASS',0)} [See list of classes](#)

Syntax

[objectreference].AdviseOnSave()

Parameters

Return value

Usage

Word Pro: Anchor method

```
{button ,AL('H_BASECONTAINER_CLASS;H_CELLCONTAINER_CLASS;H_FRAMECONTAINER_CLASS;H_NOTE  
CONTAINER_CLASS;H_RUBYCONTAINER_CLASS;H_DROPDOWNCONTAINER_CLASS;H_PAGECONTAINER_  
CLASS;H_SUBPAGECONTAINER_CLASS;H_SUPERPAGECONTAINER_CLASS;H_ROWCONTAINER_CLASS;  
H_SUPERTABLECONTAINER_CLASS;H_TABLECONTAINER_CLASS;H_PARALLELCOLSCONTAINER_CLASS  
;H_TABLEONLYCONT_CLASS',0)} See list of classes
```

```
{button ,AL('H_ANCHOR_METHOD_EXSCRIPT',1)} See example
```

Attaches an object, such as a table or frame object, to a page or a paragraph in a page.

Syntax

[objectreference].Anchor([AnchorWhere,] [ConditionType,] [RelativeType,] [AnchorParent])

Parameters

AnchorWhere

The value of this optional Variant parameter indicates where to attach an object to a page or a paragraph on a page. It must be one of the string constants below or its numeric equivalent. Default is \$LwpAnchorWhereDivisionInfo.

<u>Value</u>	<u>Effect</u>
\$LwpAnchorWhereDivisionInfo (12)	Attaches an object to a DivisionInfo object.
\$LwpAnchorWhereLayout (13)	Attaches an object to a layout object.

ConditionType

The value of this optional Variant parameter specifies on which pages of a document to attach an object. It must be one of the string constants below or its numeric equivalent. Default is \$LwpConditionTypeAllpages.

<u>Value</u>	<u>Effect</u>
\$LwpConditionTypeAllbutspecificpage (156)	Attaches an object to all pages of a document, except on the page you specify.
\$LwpConditionTypeAllpages (154)	Attaches an object to all pages of a document.
\$LwpConditionTypeOnlyevenpages (157)	Attaches an object only to even pages of a document.
\$LwpConditionTypeOnlyoddpages (158)	Attaches an object only to odd pages of a document.
\$LwpConditionTypeOnlyspecificpage (155)	Attaches an object only to a specific page in a document.
\$LwpConditionTypeStartatpage (159)	Specifies on which page to start attaching an object.

RelativeType

The value of this optional Variant parameter determines where in the page layout the table or frame object is anchored. It must be one of the string constants below or its numeric equivalent. Default is \$LwpRelativeTypeLytInlineNewline.

<u>Value</u>	<u>Effect</u>
\$LwpRelativeTypeLytContent (1670)	Anchors an object so that it is relative to the content box of the parent layout.
\$LwpRelativeTypeLytInline (1668)	Anchors an object to the text flow as a single character and affects the line height accordingly.
\$LwpRelativeTypeLytInlineNewline (1669)	Places an object on a new line by itself in the page layout.
\$LwpRelativeTypeLytInlineVert (1671)	Anchors an object so that it always moves vertically in the page layout.
\$LwpRelativeTypeLytPara (1667)	Anchors an object so that the text is relative to a paragraph in the page layout.
\$LwpRelativeTypeLytParent (1666)	Sets the coordinates of an object so that they are relative to the parent layout; places the anchor position on the page layout.

AnchorParent

The value of this Variant optional parameter indicates on which parent layout to attach an object. It must be one of the string constants below or its numeric equivalent. Default is \$LwpAnchorParentDefault.

<u>Value</u>	<u>Effect</u>
\$LwpAnchorParentCell (10)	Attaches an object to a parent cell object.
\$LwpAnchorParentDefault (11)	Attaches a table or frame object.
\$LwpAnchorParentFrame (9)	Attaches an object to a parent frame object.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Word Pro: AnswerMsgBox method

{button ,AL('H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL('H_ANSWERMSGBOX_METHOD_EXSCRIPT',1)} [See example](#)

Provides a response to a message box.

Syntax

[objectreference].AnswerMsgBox(MsgBoxResponse)

Parameters

MsgBoxResponse

The response you want to use. Data type is Variant, which allows the value of this parameter to be one of the string values listed below or its numeric equivalent (in parentheses). There is no default value.

\$LwpMsgboxReplyCancel (2076)

\$LwpMsgboxReplyIgnore (2078)

\$LwpMsgboxReplyNo (2080)

\$LwpMsgboxReplyOk (2075)

\$LwpMsgboxReplyRetry (2077)

\$LwpMsgboxReplyYes (2079)

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Use this method to handle any message boxes which Word Pro might display while your script is running. For example, if your script causes a warning message box to appear, you can include this method in your code *before* the statement which elicits the message box. Word Pro answers the first message box it sees with the response you provide in the MsgBoxResponse parameter.

{button ,AL('H_MESSAGES_METHOD_MEMDEF',0)} [See related topics](#)

Word Pro: AnyEdits method

{button ,AL('H_DIVISION_CLASS;H_TEXTDOCUMENT_CLASS',0)} [See list of classes](#)

{button ,AL('H_ANYEDITS_METHOD_EXSCRIPT',1)} [See example](#)

Syntax

[objectreference].AnyEdits(EditorName)

Parameters

EditorName

A String expression representing the name of the editor.

Return value

Usage

Word Pro: AppendMacro method

{button ,AL('H_MACRO_CLASS',0)} [See list of classes](#)

Syntax

[objectreference].AppendMacro()

Parameters

Return value

Usage

Word Pro: Backspace method

{button ,AL('H_CLICKHERE_CLASS;H_TEXTMARKER_CLASS;H_TEXT_CLASS',0)} [See list of classes](#)

{button ,AL('H_BACKSPACE_METHOD_EXSCRIPT',1)} [See example](#)

Executes a backspace. Similar, but not identical to, pressing the Backspace key.

Syntax

[objectreference].Backspace(Count)

Parameters

Count

An Integer expression specifying the number of backspaces Word Pro should execute. You must use positive integers for this value. Negative integers will produce unpredictable results.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

If any text is selected when you call this method, the selection is treated the same as the insertion point. The selection itself remains untouched, while the text preceding the selection becomes the subject of the backspace.

Word Pro: Backward method

```
{button ,AL('H_BASECONTAINER_CLASS;H_CELLCONTAINER_CLASS;H_FRAMECONTAINER_CLASS;H_NOTECONTAINER_CLASS;H_RUBYCONTAINER_CLASS;H_DROPCAPCONTAINER_CLASS;H_PAGECONTAINER_CLASS;H_SUBPAGECONTAINER_CLASS;H_SUPERPAGECONTAINER_CLASS;H_ROWCONTAINER_CLASS;H_SUPERTABLECONTAINER_CLASS;H_TABLECONTAINER_CLASS;H_PARALLELCOLSCONTAINER_CLASS;H_TABLEONLYCONT_CLASS;H_LAYOUT_CLASS;H_CELLGROUPLAYOUT_CLASS;H_CELLLAYOUT_CLASS;H_CONNECTEDLAYOUT_CLASS;H_COLUMNGROUPLAYOUT_CLASS;H_FOOTERLAYOUT_CLASS;H_FRAMELAYOUT_CLASS;H_FRAMEGROUPLAYOUT_CLASS;H_DROPCAPLAYOUT_CLASS;H_GROUPLAYOUT_CLASS;H_HEADERLAYOUT_CLASS;H_NOTELAYOUT_CLASS;H_PAGELAYOUT_CLASS;H_ROWGROUPLAYOUT_CLASS;H_ROWLAYOUT_CLASS;H_RUBYLAYOUT_CLASS;H_SUPERTABLEGROUPLAYOUT_CLASS;H_SUPERTABLELAYOUT_CLASS;H_TABLELAYOUT_CLASS;H_ENDNOTELAYOUT_CLASS;H_FOOTNOTELAYOUT_CLASS;H_TABLEHEADINGLAYOUT_CLASS;H_TOCSUPERTABLELAYOUT_CLASS;H_CLICKHERE_CLASS;H_TEXTMARKER_CLASS;H_TEXT_CLASS',0)} See list of classes
```

```
{button ,AL('H_BACKWARD_METHOD_EXSCRIPT',1)} See example
```

Moves an object or the insertion point backward. A Forward method is also available.

Syntax

When called from a Layout object:

```
[objectreference.]Backward()
```

When called from a container object:

```
[objectreference.]Backward(Direction)
```

When called from a Text, TextMarker, or ClickHere object:

```
[objectreference.]Backward(Unit, N[, Cursoring][, TextOnly])
```

Parameters

Direction

Specifies whether Word Pro should move the insertion point back by page or by window. Data type is Variant, which allows the value of this parameter to be one of the string constants listed below or its numeric equivalent (in parentheses). There is no default value.

<u>Value</u>	<u>Effect</u>
\$LwpDirectionPage (182)	The method moves the insertion point back by one page.
\$LwpDirectionWindow (183)	The method moves the insertion point back by one window.

The Direction parameter is only used when calling this method from a container object. A container object is any object created from a container class. A container class is any class derived from the BaseContainer class, including: CellContainer, DropCapContainer, FrameContainer, NoteContainer, PageContainer, ParallelColsContainer, RowContainer, RubyContainer, SubPageContainer, SuperPageContainer, SuperTableContainer, TableContainer, and TableOnlyCont.

Unit

Specifies the unit of measurement you want to use in moving the insertion point. Use this parameter only when calling this method from a Text, TextMarker, or ClickHere object. You must also use the N parameter to indicate how many of these units to move backward. Data type is Variant, which allows the value of this parameter to be one of the constants listed below or its numeric equivalent (in parentheses). There is no default value.

<u>Value</u>	<u>Effect</u>
\$LwpNavigateObjectTypeCharacter (1519)	Moves the insertion point the specified number of characters. Set the Cursoring parameter to True to mimic the use of the arrow keys.
\$LwpNavigateObjectTypeChunk (1522)	Moves the insertion point the specified number of chunks. A chunk is comprised of a single word (a contiguous group of characters with no spaces) and all the spaces following that word. If the insertion point is at the beginning, the end or anywhere within a word, the chunk is comprised of

	that word and the spaces that follow it. If the insertion point is between two spaces, the chunk is seen as all the spaces following the insertion point to the beginning of the next word. If there is no word between the spaces and the end of the paragraph, the chunk is comprised of all the spaces up to the end of the paragraph.
\$LwpNavigateObjectTypeObject (1520)	Any of the objects defined in this list.
\$LwpNavigateObjectTypePage (1518)	Moves the insertion point the specified number of pages, leaving it at the top of the page.
\$LwpNavigateObjectTypeParagraph (1524)	Moves the insertion point the specified number of paragraphs. A paragraph is comprised of all the text and tables between two paragraph markers, as well as any frames whose "Place frame" option is set to "With paragraph above."
\$LwpNavigateObjectTypeSentence (1523)	Moves the insertion point the specified number of sentences. A sentence is comprised of all the text between two periods.
\$LwpNavigateObjectTypeWord (1521)	Moves the insertion point the specified number of words. A word is comprised of a contiguous string of alphanumeric characters. Punctuation or a space is seen as the end of a word. If the insertion point is between two spaces, the word is comprised of all the spaces on both sides of the insertion point, as well as the word preceding the spaces.

N

An Integer expression which specifies the number of units you want to move the insertion point. Use this parameter only when calling this method from a Text, TextMarker, or ClickHere object.

Cursoring

This optional Integer parameter is valid only when the Unit parameter has a value of \$LwpNavigateObjectTypeCharacter. This parameter takes an Integer expression which indicates whether or not you want Word Pro to move the insertion point as if you were using the arrow keys to move the cursor through a document. When you use the arrows keys, Word Pro skips over hidden markers such as bookmarks. The default value is False (0), which causes Word Pro to include any hidden markers when it moves the insertion point by characters. The legal values for this parameter are -1 and 0, but you may use the LotusScript constants True (-1) and False (0) instead of the integer values. Use this parameter only when calling this method from a Text, TextMarker, or ClickHere object.

TextOnly

An optional Integer expression which indicates whether you want Word Pro to ignore objects in the stream that are not text-like. Data type is Integer. The legal values for this parameter are -1 and 0 but you may use the LotusScript constants True (-1) and False (0) instead of the integer values. The default value is False (0), which includes objects that are not text-like. A value of True will cause Word Pro to skip over objects that are not text-like. Text-like objects include text, Click Here Blocks, and bookmarks. Use this parameter only when calling this method from a Text, TextMarker, or ClickHere object.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Even though this method has the same name on each object, it behaves differently from one object to the next. For example, when you call this method from a Layout object, it behaves differently from when you call it from a Text object.

- From a Layout object
If you call this method from a Layout object, that Layout object moves backward one level in relation to the other layout objects of the same type. For example, a FrameLayout is moved backward one step in the hierarchy of

FrameLayouts.

- From a container object

If you call this method from a container object, Word Pro places the insertion point at the beginning of the previous page.

- From a Text object

If you call this method from a Text, TextMarker, or ClickHere object, Word Pro moves the insertion point backward the specified number of units.

The backward method will not enter embedded streams such as frames or tables. Use the methods provided by those objects to enter their streams or remove them completely.

Word Pro: BeginChange method

{button ,AL(`H_WPAPPLICATION_CLASS;H_DIVISION_CLASS;H_TEXTDOCUMENT_CLASS',0)} [See list of classes](#)

{button ,AL(`H_BEGINCHANGE_METHOD_EXSCRIPT',1)} [See example](#)

Syntax

[objectreference].BeginChange([DontMarkChanges])

Parameters

DontMarkChanges

An optional Boolean expression that allows you to mark (True) or not mark (False) any changes you begin in the Word Pro application, division, or text document object. A Boolean expression is either True or False.

Return value

Usage

If a script contains a .BeginChange, a corresponding .EndChange must take place before the script exits, even if the script exits due to an error. Therefore, any script that contains a .BeginChange must include an error handler that does a corresponding .EndChange, if an error occurs between the .BeginChange and .EndChange.

Word Pro: BeginCustomLines method

{button ,AL(^H_TABLELINE_CLASS',0)} [See list of classes](#)

Used by Word Pro when recording a script, in order to reflect the beginning of a customized table line style selection.

Syntax

[objectreference].BeginCustomLines()

Parameters

Return value

Always returns 0.

Usage

Word Pro: Bold method

{button ,AL('H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL('H_BOLD_METHOD_EXSCRIPT',1)} [See example](#)

Sets the bold attribute for selected text, or all following text, if no text is selected. Acts as a toggle, turning the attribute off if it is on, and on if it is off. Equivalent to choosing Text - Attributes - Bold.

Syntax

[objectreference].Bold()

Parameters

None.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Word Pro: BreakLink method

{button ,AL(^H_GRAPHIC_CLASS;H_OLEOBJECT_CLASS',0)} [See list of classes](#)

Note This method is not implemented for the OLEObject class within OS/2.

Syntax

[objectreference].BreakLink(LinkCookie)

[objectreference].BreakLink()

Parameters

LinkCookie

Data type is Long.

Return value

Integer.

Usage

Word Pro: BringFrameToFrontOne method

{button ,AL('H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL('H_BRINGFRAMETOFRONTONE_METHOD_EXSCRIPT',1)} [See example](#)

Brings the currently active frame one step forward in the frame order. Equivalent to choosing Frame - Priority, then Bring Forward One.

Syntax

[objectreference].BringFrameToFrontOne()

Parameters

None.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

When you have more than one frame on a page, Word Pro sees the frames as being stacked on top of each other, even if they don't appear to overlap on the page. The first frame you create is at the bottom of the stack. The second frame you create is on top of the first frame, but underneath the third frame, and so on. You can use this method to change the order of a frame in the stack.

Word Pro: BringFrameToFront method

{button ,AL('H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL('H_BRINGFRAMETOFRONT_METHOD_EXSCRIPT',1)} [See example](#)

Brings the currently active frame to the front of all other frames on the page. Equivalent to choosing Frame - Priority, then Bring to Front.

Syntax

[objectreference].BringFrameToFront()

Parameters

None.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

When you have more than one frame on a page, Word Pro sees the frames as being stacked on top of each other, even if they don't appear to overlap on the page. The first frame you create is at the bottom of the stack. The second frame you create is on top of the first frame, but underneath the third frame, and so on. You can use this method to change the order of a frame in the stack.

Word Pro: CalculateSmartLevels method

```
{button ,AL(^H_WPAPPLICATION_CLASS;H_CLICKHERE_CLASS;H_TEXTMARKER_CLASS;H_TEXT_CLASS',0)}
```

[See list of classes](#)

```
{button ,AL(^H_CALCULATESMARTLEVELS_METHOD_EXSCRIPT',1)} See example
```

Updates the SmartLevels for the currently active division. This update only applies to those paragraphs that are marked to "Use Smart Level" on the Misc panel in the Text Properties InfoBox.

Syntax

```
[objectreference].CalculateSmartLevels()
```

Parameters

None.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Word Pro: CascadeWindow method

{button ,AL(^H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL(^H_CASCADEWINDOW_METHOD_EXSCRIPT',1)} [See example](#)

Displays the active document window on top of all other open document windows, with the title bar for each document visible. Equivalent to choosing Window - Cascade.

Syntax

[objectreference].CascadeWindow()

Parameters

None.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Word Pro: Cascade method

{button ,AL('H_APPLICATIONWINDOW_CLASS',0)} [See list of classes](#)

{button ,AL('H_CASCADE_METHOD_EXSCRIPT',1)} [See example](#)

Cascades the document windows in the application.

Syntax

[objectreference].Cascade()

Parameters

None.

Return value

The return value for this method will always be -1 or 0. When testing the return value, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Equivalent to choosing Window - Cascade in the Word Pro interface.

Word Pro: CellLayout method

{button ,AL(`H_BASSETABLE_CLASS;H_FOOTNOTETABLE_CLASS;H_PARALLELCOLUMNS_CLASS;H_GLOSSARY_CLASS;H_TABLEHEADING_CLASS;H_TABLE_CLASS',0)} [See list of classes](#)

{button ,AL(`H_CELLLAYOUT_METHOD_EXSCRIPT',1)} [See example](#)

Returns the cell layout for a specific cell in a table.

Syntax

[objectreference].CellLayout([Row,] [Column])

Parameters

Row

An optional Integer parameter that allows you to indicate the specific row from which you want to return its layout. Default is the current row ID value.

Column

An optional Integer parameter that allows you to indicate the specific column from which you want to return its layout. Default is the current column ID value.

Return value

Returns the CellLayout object specified by the row and column parameters.

Usage

Using this method, you can set a variable to a specific cell layout as follows:

```
set mycell = .table.celllayout(1, 1)
```

You can also directly access the properties and methods of a specified cell layout, as shown in the following example:

```
.table.celllayout(0,0).background.color.setrgb 255, 0, 0
```

Word Pro: CellRevert method

{button ,AL(`H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL(`H_CELLREVERT_METHOD_EXSCRIPT',1)} [See example](#)

Reverts the currently active table cell to the attributes of the assigned table cell style.

Syntax

[objectreference].CellRevert()

Parameters

None.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Word Pro: ChangeSmartMaster method

{button ,AL(^H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL(^H_CHANGESMARTMASTER_METHOD_EXSCRIPT',1)} [See example](#)

Changes the SmartMaster for the currently active Word Pro document. Equivalent to choosing File - Choose Another SmartMaster, and specifying a new SmartMaster.

Syntax

[objectreference].ChangeSmartMaster(PathName,Type, [ApplyTo])

Parameters

PathName

A String expression specifying the new SmartMaster to which you are changing.

Type

A String expression specifying the file type of SmartMaster you want to use. Word Pro automatically recognizes and imports all of the file types listed below. Use this parameter only if the file you are using is not one of these file types:

DCA/RFT	Lotus Manuscript 2.x	MS Word for Windows 1.0
DIF	Lotus Organizer 1.x	MS Word for Windows 2.0
DisplayWrite	Lotus Word Pro	MS Word for Windows 6.0
HTML	Lotus Word Pro SmartMaster	MS Word for Windows95 7.0
Lotus 1-2-3	MS Excel	MS WordPad 1.0
Lotus 1-2-3 for OS/2	MS Excel 3.0	OfficeWriter 4,5,6
Lotus 1-2-3 R3	MS Excel 4.0	Rich Text Format(RTF)
Lotus 1-2-3 R4,5	MS Excel 5.0	SAMNA Word
Lotus 1-2-3 R6	MS Excel 7.0	WordPerfect 5.0
Lotus Ami Pro	MS Windows Write 3.x	WordPerfect 5.1
Lotus Ami Pro 3.x Macro	MS Word for DOS 3,4,5,6	WordPerfect 6.x
Lotus Ami Pro 3.x Styles	MS Word for OS/2	WordStar 2000 R3

ApplyTo

Allows you to specify the scope of the new SmartMaster. Optional parameter. There are three legal values for this parameter:

<u>Value</u>	<u>Effect</u>
Entire document	Applies the new SmartMaster to the entire document.
All divisions at same level & below	Applies the new SmartMaster to all the divisions which are at or below the same level as the currently active division (a division's level is indicated by its parent-child relationship to other divisions).
Current division only	Applies the new SmartMaster only to the currently active division.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Word Pro: ChgLineStyle method

{button ,AL('H_TABLELINE_CLASS',0)} [See list of classes](#)

Changes the style of a line in a TableLine object.

Syntax

[objectreference].ChgLineStyle(LineStyle)

Parameters

LineStyle

Data type is Variant, which allows the value of this parameter to be one of the constants listed below or its numeric equivalent (in parentheses). There is no default value.

<u>Value</u>	<u>Effect</u>
\$LwpTableLineStyleAll (1878)	All lines display on the TableLine object.
\$LwpTableLineStyleCols (1882)	Column lines display on the TableLine object.
\$LwpTableLineStyleCustom (1886)	Custom lines display on the TableLine object.
\$LwpTableLineStyleInnercols (1887)	Inner column displays on the TableLine object.
\$LwpTableLineStyleInnerRowscols (1888)	Inner rows and column lines display on the TableLine object.
\$LwpTableLineStyleMixed (1885)	Mixed lines display on the TableLine object.
\$LwpTableLineStyleNone (1877)	No lines display on the TableLine object.
\$LwpTableLineStyleOutline (1879)	Outline lines display on the TableLine object.
\$LwpTableLineStyleOutlineall (1880)	All lines and an outline border display on the TableLine object.
\$LwpTableLineStyleOutlinecols (1884)	Column lines and an outline border display on the TableLine object.
\$LwpTableLineStyleOutlinerows (1883)	Row lines and an outline border display on the TableLine object.
\$LwpTableLineStyleRows (1881)	Row lines display on the TableLine object.

Return value

Always returns 0.

Usage

Before calling this method, you must make sure that the BorderLine and OutlineBorderLine objects are set to appropriate values. For example, if you want to show only the outline of a selected Table object, you can use the following code:

```
With .Table.TableLine
    .BorderLines.AllBorders.Pattern = $LtsBorderPatternNone
    .OutlineBorderLines.AllBorders.Pattern = $LtsBorderPatternSolid
    .OutlineBorderLines.AllBorders.WidthInTwips = 20
    .ChgLineStyle $LwpTableLineStyleOutline
End With
```

Notice that before the ChgLineStyle method is called with the \$LwpTableLineStyleOutline parameter, the table line borders and outline borders are set to the appropriate values. If the ChgLineStyle method is called without making appropriate changes to the Table object's borders, unpredictable results may occur.

Word Pro: ClearAll method

{button ,AL('H_TABRACK_CLASS',0)} [See list of classes](#)

{button ,AL('H_CLEARALL_METHOD_EXSCRIPT',1)} [See example](#)

Removes all tabs from the ruler.

Syntax

[objectreference].ClearAll()

Parameters

Return value

Usage

Word Pro: ClearDivisionList method

{button ,AL('H_PRINTSETTINGS_CLASS',0)} [See list of classes](#)

{button ,AL('H_CLEARDIVISIONLIST_METHOD_EXSCRIPT',1)} [See example](#)

Clears the list of divisions to be printed in a document.

Syntax

[objectreference].ClearDivisionList()

Parameters

None.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

The AddDivisionList method adds divisions to the list of divisions to be printed. Use ClearDivisionList to clear the list of divisions to be printed.

Word Pro: ClearParaRevisionTags method

{button ,AL('H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL('H_CLEARPARAREVISIONTAGS_METHOD_EXSCRIPT',1)} [See example](#)

Removes paragraph revision tags from a document. If more than one revision tag exists for a group of revisions, Word Pro prompts you to leave the paragraphs alone and leave the revision tags intact, or clear the tags and leave both versions of the paragraph.

Syntax

[objectreference].ClearParaRevisionTags()

Parameters

None

Return value

None

Usage

Affects all tags in all divisions in the currently active document.

Word Pro: ClearSplits method

{button ,AL('H_WINVIEWPREFS_CLASS',0)} [See list of classes](#)

{button ,AL('H_CLEARSPLOTS_METHOD_EXSCRIPT',1)} [See example](#)

Clears all split views from the screen. Equivalent to choosing View - Clear All Splits.

Syntax

[objectreference].ClearSplits()

Parameters

None.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Word Pro: ClearTempFoundry method

{button ,AL(^H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

Clears the contents of the Foundry object located in the TempFoundry property on the WPAApplication object.

Syntax

[objectreference].ClearTempFoundry()

Parameters

None.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Identical to the Clear method located on the Foundry class, but only affects the Foundry object located in the TempFoundry property.

Word Pro: ClearUpdate method

{button ,AL(^H_WINVIEWPREFS_CLASS',0)} [See list of classes](#)

For internal use only.

Syntax

[objectreference].ClearUpdate()

Parameters

None.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Word Pro: Clear method

```
{button ,AL(^H_ATTRIBUTES_CLASS;H_BAG_CLASS;H_DIVISION_CLASS;H_FONT_CLASS;H_FOUNDRY_CLASS;H_JOIN_CLASS;H_CLICKHERE_CLASS;H_TEXTMARKER_CLASS;H_MERGEOPTIONS_CLASS;H_OUTLINE_STYLESEQUENCE_CLASS;H_PARAGRAPHSTYLE_CLASS;H_TEXT_CLASS;H_TEXTDOCUMENT_CLASS;H_TOCSUPERTABLELAYOUT_CLASS',0)} See list of classes
```

```
{button ,AL(^H_CLEAR_METHOD_EXSCRIPT',1)} See example
```

[Attributes]

[Bag]

[Division]

Clears the content of the Division object.

[Font]

[Foundry]

Clears all objects from a Foundry object. Use this method only on Foundry objects found in the AppFoundry or TempFoundry properties on the WPAApplication object. DO NOT use this method with the Foundry property in WPAApplication, Division, or TextDocument.

[ClickHere]

[TextMarker]

[MergeOptions]

Disconnects the Merge data file from the current document.

[OutlineStyleSequence]

[ParagraphStyle]

[Text]

[TextDocument]

[TOCSuperTableLayout]

Syntax

[Objectreference].Attributes.Clear

[Objectreference].Bag.Clear

[Objectreference].Division.Clear

[Objectreference].Font.Clear

[Objectreference].Foundry.Clear

[Objectreference].ClickHere.Clear([ClearWhat,] [p2,] [ClassName,] [SubClass])

[Object reference].TextMarker.Clear([ClearWhat,] [p2,] [ClassName,] [SubClass])

[Object reference].MergeOptions.Clear

[Object reference].OutlineStyleSequence.Clear

[Object reference].ParagraphStyle.Clear

[Object reference].Text.Clear([ClearWhat,] [p2,] [ClassName,] [SubClass])

[Object reference].TextDocument.Clear [Object reference].TOCSuperTableLayout.Clear

Parameters

[Foundry]

Reserved

Not used. Do not use this parameter when using the Clear method on a Foundry object.

[ClickHere, TextMarker, Text]

ClearWhat

Data type is Variant. Optional parameter on ClickHere, Text, and TextMarker objects. The value of this parameter must be one of the string values listed below or its Enum code. Default is \$LwpClearWhatDefault.

\$LwpClearWhatClearMisspelledWord (141)

\$LwpClearWhatDefault (145)

\$LwpClearWhatHighlighter (144)

\$LwpClearWhatObject (143)

\$LwpClearWhatTombstone (142)

[ClickHere, TextMarker, Text]

p2

Data type is Variant. Optional parameter. Default is 0.

[ClickHere, TextMarker, Text]

ClassName

Data type is String. Optional parameter.

[ClickHere, TextMarker, Text]

SubClass

Data type is String. Optional parameter.

Return value

ClickHere, TextMarker, Text, Foundry - A Boolean value indicating success (-1) or failure (0). The return value for this method will always be -1 or 0. When testing the return value, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

[Foundry]

Used from the Foundry property on Division, TextDocument, or WPApplication, this method clears all styles and everything else.

Word Pro: CloseAll method

{button ,AL('H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL('H_CLOSEALL_METHOD_EXSCRIPT',1)} [See example](#)

Closes all open documents. This method does not close hidden files, such as glossary files.

Syntax

[objectreference].CloseAll([CloseFile])

Parameters

CloseFile

Closes an untitled document without the Save Changes dialog box, as long as the document has no contents. Default is \$LwpCloseFileIfLastdocOpenUntitled, which closes empty untitled documents without a prompt. Data type is Variant, which allows you to use one of the string values below or its numeric equivalent (in parentheses).

<u>Value</u>	<u>Effect</u>
\$LwpCloseFileIfLastdocNoOpen (147)	Prompts you to save the untitled document before closing.
\$LwpCloseFileIfLastdocOpenUntitled (146)	Closes the untitled document without saving.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Word Pro: CloseDocWindow method

{button ,AL('H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL('H_CLOSEDWINDOW_METHOD_EXSCRIPT',1)} [See example](#)

Closes the currently active document window.

Syntax

[objectreference].CloseDocWindow([AskUserToSave])

Parameters

AskUserToSave

Prompts you to save the document before closing. Data type is Integer. The legal values for this parameter are -1 and 0 but you may use the LotusScript constants True (-1) and False (0). Optional parameter. Default is True, which prompts you to save the document.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Word Pro: CloseMergeDataFile method

{button ,AL(^H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL(^H_CLOSEMERGEDATAFILE_METHOD_EXSCRIPT',1)} [See example](#)

Closes the data file for the active merge document. Any changes you make are lost if this method is called before you save the file.

Syntax

[objectreference].CloseMergeDataFile()

Parameters

None.

Return value

Usage

Word Pro: CloseObject method

{button ,AL('H_CLICKHERE_CLASS;H_TEXTMARKER_CLASS;H_TEXT_CLASS',0)} [See list of classes](#)

{button ,AL('H_CLOSEOBJECT_METHOD_EXSCRIPT',1)} [See example](#)

Closes the comment note at the insertion point.

Syntax

[objectreference].CloseObject()

Parameters

None.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

This method must be called from a Text object that contains a comment note. Word Pro will do nothing if this method is called from a Text object that contains no comment note. For example, if you call this method from the Text object within a comment note, the comment note will not close.

A comment note is represented in LotusScript by a NoteContainer object, which is comprised of several class members and objects, including a NoteLayout object, a DivisionInfo object, and a Presentation object.

Word Pro: Close method

{button ,AL('H_WPAPPLICATION_CLASS;H_DOCUMENT_CLASS;H_TEXTDOCUMENT_CLASS;H_WINDOW_CLASS;H_APPLICATIONWINDOW_CLASS;H_STATUSBAR_CLASS;H_DOCWINDOW_CLASS',0)} [See list of classes](#)

{button ,AL('H_CLOSE_METHOD_EXSCRIPT',1)} [See example](#)

Closes the object from which you call this method. For example, when you call this method from WPAApplication, it closes the active document, but when you call it from a StatusBar object, it hides the status bar from which you call the method. See Usage below for details of how this method affects each type of object.

Syntax

[Objectreference].WPAApplication.Close([SaveChanges,] [DocName,] [Location,] [DocType,] [CloseFile])

[Objectreference].TextDocument.Close([SaveChanges,] [DocName,] [Location,] [DocType,] [CloseFile])

[Objectreference].ApplicationWindow.Close() Integer

[Objectreference].DocWindow.Close() Integer

[Objectreference].Window.Close() Integer

Parameters

SaveChanges

Used only on WPAApplication and TextDocument objects. Lets you choose to save or dismiss all changes before closing a document. Data type is Integer. The legal values for this parameter are -1 and 0 but you may use the LotusScript constants True (-1) and False (0). Optional parameter. Default is True.

DocName

Used when the document has never been saved, this parameter takes a string expression which specifies the name of the document you are closing.

Location

Used when the document has never been saved, this parameter takes a string expression which specifies the directory path for the document.

DocType

Used when the document has never been saved, this parameter takes a string expression which specifies the file type for the document. Default file type is Word Pro document.

CloseFile

Data type is Variant, which allows the value of this parameter to be one of the string values listed below or its numeric equivalent (in parentheses). Default value is \$LwpCloseFileIfLastdocOpenUntitled.

<u>Value</u>	<u>Effect</u>
\$LwpCloseFileIfLastdocNoOpen (147)	Leaves an empty application workspace window if you close the last document.
\$LwpCloseFileIfLastdocOpenUntitled (146)	Opens an untitled file if you close the last document.

Return value

The return values for this method will always be -1 or 0. When testing the return value, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

This method appears on several objects. It closes whatever type of object you call it from.

WPAApplication - Closes the currently active Word Pro document.

TextDocument - Closes the document from which you call the method.

DocWindow - Closes the document from which you call the method.

StatusBar - Closes/hides the status bar.

ApplicationWindow - Closes the application window. This is NOT the same as choosing File - Exit Word Pro. Although the application window closes, if Word Pro is serving an object to an external client, it remains active but not visible.

When all the served objects are released, the application terminates.

Word Pro: CombineDivisions method

{button ,AL('H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL('H_COMBINEDIVISIONS_METHOD_EXSCRIPT',1)} [See example](#)

Combines two divisions into a single division. The divisions must be adjacent to each other.

Syntax

[objectreference].CombineDivisions (StartName, EndName)

Parameters

StartName

The internal name for the first division you want to combine. This is not the name Word Pro displays in the division tab. Data type is String.

EndName

The internal name for the first division you want to combine. This is not the name Word Pro displays in the division tab. Data type is String.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

You can get the internal name for the active division by calling the Name property of the currently active Division object as shown below:

```
DIM StartName As String
```

```
StartName = .Division.Name
```

For more information on division names in LotusScript, see [Overview: Division names in LotusScript](#)

Word Pro: CombineSections method

{button ,AL('H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL('H_COMBINESECTIONS_METHOD_EXSCRIPT',1)} [See example](#)

Combines the contents of up to ten sections by removing the specified section breaks.

Syntax

```
[objectreference].CombineSections([Section1Name] [, Section2Name] [, Section3Name]  
[, Section4Name] [, Section5Name] [, Section6Name] [, Section7Name] [, Section8Name]  
[, Section9Name] [, Section10Name])
```

Parameters

Section1Name

A String expression which specifies a section break you want to remove. The contents of this section will then be placed at the end of the previous section.

Section2Name through *Section10Name* allow you to specify additional section breaks to be removed. The contents of each section are added to the end of the preceding section.

Note The names used in these parameters are the internal hexadecimal names found in the Name property on a Section object. You can access a Section object through the SectionCollection found in the Sections property in the Division's Foundry object. A sample script is provided below.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Equivalent to clicking the right mouse button on a section tab and choosing Combine Sections. However, unlike the menu command, this method allows you to combine more than two sections.

You can use the following script to get both the internal hexadecimal name (found in the Name property of each Section object) and the name which appears in the Section tab (found in the UserName property of each Section object).

```
Print "Section Label = Section Internal Name"  
Forall sec In .Division.Foundry.Sections  
    Print sec.UserName + " = " sec.Name  
End Forall
```

Word Pro: CompareFiles method

{button ,AL('H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL('H_COMPAREFILES_METHOD_EXSCRIPT',1)} [See example](#)

Compares the file(s) you specify with the currently active document.

Syntax

[objectreference].CompareFiles(FilePath, FileType, IsMultiDocs, IndexOfMultiDocToCompare)

Parameters

FilePath

A String expression specifying the name and path for the document that you want to compare against the currently active document. If you are comparing more than one document, this should be the name of the first document you want to compare. Data type is String.

FileType

The file type of document named in FilePath. Use a null string ("") to have Word Pro automatically detect the file type. Word Pro recognizes the following file types:

DCA/RFT	Lotus Manuscript 2.x	MS Word for Windows 1.0
DIF	Lotus Organizer 1.x	MS Word for Windows 2.0
DisplayWrite	Lotus Word Pro	MS Word for Windows 6.0
HTML	Lotus Word Pro SmartMaster	MS Word for Windows95 7.0
Lotus 1-2-3	MS Excel	MS WordPad 1.0
Lotus 1-2-3 for OS/2	MS Excel 3.0	OfficeWriter 4,5,6
Lotus 1-2-3 R3	MS Excel 4.0	Rich Text Format(RTF)
Lotus 1-2-3 R4,5	MS Excel 5.0	SAMNA Word
Lotus 1-2-3 R6	MS Excel 7.0	WordPerfect 5.0
Lotus Ami Pro	MS Windows Write 3.x	WordPerfect 5.1
Lotus Ami Pro 3.x Macro	MS Word for DOS 3,4,5,6	WordPerfect 6.x
Lotus Ami Pro 3.x Styles	MS Word for OS/2	WordStar 2000 R3

IsMultiDocs

Indicates whether you are comparing one file or multiple files to the active document. Data type is Integer. A True value (-1) indicates you are comparing multiple documents. A False value (0) indicates that you are comparing a single document.

IndexOfMultiDocToCompare

When comparing multiple documents, the value of this parameter specifies which document is being compared in the current iteration of the loop. Data type is Integer.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

To compare multiple files, you must use a loop which will call this method for each file you want to compare, and you must include different values for the *IndexOfMultiDocToCompare* parameter in each iteration of the loop.

Word Pro: Configure method

{button ,AL('H_ICONBAR_CLASS;H_ICONBARMANAGER_CLASS',0)} [See list of classes](#)

{button ,AL('H_CONFIGURE_METHOD_EXSCRIPT',1)} [See example](#)

Activates (loads and displays) the SmartIcons Setup dialog box.

Syntax

[objectreference].Configure()

Parameters

None.

Return value

The return value for this method will always be -1 or 0. When testing the return value, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

[IconBar]

Allows you to edit a specific icon bar set. The icon bar set from which you call this method displays in the SmartIcons Setup dialog box.

[IconBarManager]

Allows you to add, change, and edit icon bar sets by using the list in the IconBarManager.

Word Pro: ConnectCells method

{button ,AL('H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL('H_CONNECTCELLS_METHOD_EXSCRIPT',1)} [See example](#)

Connects selected table cells. Equivalent to choosing Table - Connect Cell.

Syntax

[objectreference].ConnectCells()

Parameters

None.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

If you disconnect a cell, the contents of the cell remain in a single cell rather than returning to the original separate cells. Use the DisconnectCells method to disconnect rows.

Word Pro: ConnectContainer method

{button ,AL(^H_WPAPPLICATION_CLASS;H_BASECONTAINER_CLASS;H_CELLCONTAINER_CLASS;H_FRAMECONTAINER_CLASS;H_NOTECONTAINER_CLASS;H_RUBYCONTAINER_CLASS;H_DROPDOWNCONTAINER_CLASS;H_PAGECONTAINER_CLASS;H_SUBPAGECONTAINER_CLASS;H_SUPERPAGECONTAINER_CLASS;H_ROWCONTAINER_CLASS;H_SUPERTABLECONTAINER_CLASS;H_TABLECONTAINER_CLASS;H_PARALLELCOLSCONTAINER_CLASS;H_TABLEONLYCONT_CLASS',0)} [See list of classes](#)

Groups the selected container objects.

Syntax

[objectreference].ConnectContainer()

Parameters

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Use this method to group two or more containers. When the containers are grouped, handles display on the sides. These handles can be used to move the grouped containers. This method is usually used when selecting frames, cells, and so on.

When this method is used to connect CellContainer objects, the contents of the selected cells merge into the connected cell.

For more information on container objects, see [BaseContainer class](#).

Word Pro: ConnectRows method

{button ,AL(^H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL(^H_CONNECTROWS_METHOD_EXSCRIPT',1)} [See example](#)

Connects all the cells in the same row as the active or selected cell(s). Equivalent to choosing Table - Connect Row.

Syntax

[objectreference].ConnectRows()

Parameters

None.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

If you disconnect a row, the contents of the row remain in a single cell, rather than being returned to the original separate cells and rows. Use the DisconnectCells method to disconnect rows.

Word Pro: ConnectSectionTabs method

{button ,AL('H_SECTIONTABS_CLASS',0)} [See list of classes](#)

{button ,AL('H_CONNECTSECTIONTABS_METHOD_EXSCRIPT',1)} [See example](#)

Creates a new division and makes the currently selected division the child of the new division.

Syntax

[objectreference].ConnectSectionTabs()

Parameters

None.

Return value

The return value for this method will always be -1 or 0. When testing the return value, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Equivalent to clicking the right mouse button on an existing division divider tab and choosing Group Tabs. Word Pro creates a new parent division and places the division on which you originally clicked in the new parent.

Word Pro: Connect method

{button ,AL(^H_BASSETABLE_CLASS;H_FOOTNOTETABLE_CLASS;H_PARALLELCOLUMNS_CLASS;H_GLOSSARY_CLASS;H_TABLEHEADING_CLASS;H_TABLE_CLASS',0)} [See list of classes](#)

{button ,AL(^H_CONNECT_METHOD_EXSCRIPT',1)} [See example](#)

Connects two or more rows, columns, and cells to create one large row, column or cell in a table object.

Syntax

[objectreference].Connect([StartRow],[StartCol],[EndRow],[EndCol],[MergeCon])

Parameters

StartRow

An Integer parameter that allows you to indicate the row number of the first cell to be connected. Optional parameter. Default is the ID of the first row included in the current selection of cells.

StartCol

An Integer parameter that allows you to indicate the column number of the first cell to be connected. Optional parameter. Default is the ID of the first column included in the current selection of cells.

EndRow

An Integer parameter that allows you to indicate the row number of the last cell to be connected. Optional parameter. Default is the ID of the last row included in the current selection of cells.

EndCol

An Integer parameter that allows you to indicate the column number of the last cell to be connected. Optional parameter. Default is the ID of the last column included in the current selection of cells.

MergeContents

An optional Integer parameter that indicates whether the contents of the specified cells should be merged into the connected cell. Default is -1. You can also use the LotusScript constants of True (-1) or False (0) in this parameter. If you specify a False value, the contents of the top left cell are included in the connected cell.

Return value

This method returns an Integer value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Word Pro: ConsistencyCheck method

{button ,AL('H_DIVISION_CLASS;H_TEXTDOCUMENT_CLASS',0)} [See list of classes](#)

Syntax

[objectreference].ConsistencyCheck(Fix, [RunInBackground], [FromNew])

Parameters

Fix

Data type is integer.

RunInBackground

Data type is integer.

FromNew

Data type is integer.

Return value

Usage

Word Pro: ContractOutlineLevel method

```
{button ,AL('H_WPAPPLICATION_CLASS;H_CLICKHERE_CLASS;H_TEXTMARKER_CLASS;H_TEXT_CLASS',0)}
```

[See list of classes](#)

```
{button ,AL('H_CONTRACTOUTLINELEVEL_METHOD_EXSCRIPT',1)} See example
```

Contracts the lowest level heading(s) that are subordinate to the paragraph for which you are calling the method. For example, when you call this method for a Level 1 heading, it contracts the lowest level heading(s) that are subordinate to that Level 1 heading.

Syntax

For WPAplication objects:

```
[objectreference.]ContractOutlineLevel([ContractAll])
```

For Text, TextMarker, and ClickHere objects:

```
[objectreference.]ContractOutlineLevel(ContractAll)
```

Parameters

ContractAll

Contracts all the subordinate headings under the heading from which you call this method. Data type is Integer, but the legal values for this parameter are -1 and 0. You may use the LotusScript constants True (-1) and False (0). A True value causes all subordinate headings to be contracted, regardless of their level. When called from WPAplication, this parameter is optional and has a default of True.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

This method is defined in four different classes:

[WPAplication]

Call from this object when you want to contract the outline level(s) for the heading which currently has the focus.

[ClickHere]

Call from this object when you want to contract the outline level(s) for a heading in a ClickHere object.

[TextMarker]

Call from this object when you want to contract the outline level(s) for a heading in a TextMarker object.

[Text]

Call from this object when you want to contract the outline level(s) for a specific Text object.

Word Pro: Contract method

{button ,AL('H_SECTIONTABS_CLASS',0)} [See list of classes](#)

{button ,AL('H_CONTRACT_METHOD_EXSCRIPT',1)} [See example](#)

Hides divider tabs that are the children of the division from which this method is called.

Syntax

[objectreference].Contract()

Parameters

None.

Return value

The return value for this method will always be -1 or 0. When testing the return value, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Equivalent to clicking the minus sign on the parent division tab to hide all children divider tabs and display just the parent division tab.

Word Pro: ConvertToClass method

{button ,AL(^H_GRAPHIC_CLASS',0)} [See list of classes](#)

Syntax

[objectreference].ConvertToClass(AFIDClassName)

Parameters

AFIDClassName

Data type is String.

Return value

Integer.

Usage

Word Pro: CopyItem method

{button ,AL('H_MENUITEM_CLASS',0)} [See list of classes](#)

{button ,AL('H_COPYITEM_METHOD_EXSCRIPT',1)} [See example](#)

Copies a specified menu item object from one menu item's parent object to another.

Syntax

[objectreference].CopyItem(FromItem,[After,] [TargetText,] [Caption])

Parameters

FromItem

Specifies the menu item you want to copy.

After

A True value places the copied item after the item specified in the TargetText parameter. A False value places the copied item before the item specified in the TargetText parameter. Optional Boolean parameter. Default is True.

TargetText

An optional String expression that allows you to specify any menu item object and place the copied item before or after it. If this parameter is omitted, the copied menu item will be placed at the end of the destination menu.

Caption

The name of the copied menu item that displays on the menu. You can use this optional String parameter to change the caption of a copied menu item.

Return value

The return value for this method will always be -1 or 0. When testing the return value, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

This method copies a specified menu item into the MenuItem object from which this method was called. Use this method to copy existing menu items from one location to another. If the source menu item uses a keyboard shortcut (for example, File - Save uses Ctrl-S), you can separate the menu name from the shortcut using Chr(8):

```
Save="Save" & Chr(8) & "Ctrl+S"
```

Word Pro: CopyMeaning method

{button ,AL('H_GLOSSARY_CLASS',0)} [See list of classes](#)

{button ,AL('H_COPYMEANING_METHOD_EXSCRIPT',1)} [See example](#)

Copies the meaning of a glossary term (what the glossary entry represents) to the TextCollection object of the temporary Foundry.

Syntax

[objectreference].CopyMeaning(P1)

Parameters

P1

This String parameter represents the glossary term for which the meaning will be copied.

Return value

A String value that represents the name of the meaning which has been copied to the temporary Foundry.

Usage

Used when you want to retrieve the meaning for a specific glossary term. Word Pro copies the meaning from the glossary file to the appropriate collection object of the temporary Foundry. In order to paste the glossary term meaning into the document, use the InternalPaste method of the Text class, and specify that the contents of the temporary Foundry should be pasted.

You must access the glossary object in the division Foundry of a glossary file in order to use this method. By default, Word Pro uses a glossary file named "glossary.gls."

For example, if there is a glossary entry called "XYZ", the following lines of code copy the meaning of "XYZ" to the temporary Foundry, then paste it into the text of the current document. "Glos" is the Glossary object that contains the "XYZ" glossary entry.

```
meaning$ = glos.CopyMeaning("XYZ")
.Text.InternalPaste $LwpFoundryTypeTemporary
```

Note For most purposes, you should use the GlossaryInsert method of the WPApplication object to insert glossary items into a document.

{button ,AL('H_GLOSSARYINSERT_METHOD_MEMDEF',0)} [See related topics](#)

Word Pro: CopySelection method

{button ,AL(^H_WPAPPLICATION_CLASS;H_DOCUMENT_CLASS;H_TEXTDOCUMENT_CLASS',0)} [See list of classes](#)

{button ,AL(^H_COPYSELECTION_METHOD_EXSCRIPT',1)} [See example](#)

Copies the current selection to the Clipboard and to the Foundry object, located in the AppFoundry property on WPApplication. Equivalent to choosing Edit - Copy.

Syntax

[objectreference].CopySelection()

Parameters

None.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Word Pro: Copy method

{button ,AL('H_DIVISION_CLASS;H_FOUNDRY_CLASS;H_BASSETABLE_CLASS;H_FOOTNOTETABLE_CLASS;H_PARALLELCOLUMNS_CLASS;H_GLOSSARY_CLASS;H_TABLEHEADING_CLASS;H_TABLE_CLASS;H_TEXDOCUMENT_CLASS',0)} [See list of classes](#)

{button ,AL('H_COPY_METHOD_EXSCRIPT',1)} [See example](#)

[Foundry class]

Copies objects from one Foundry object into another Foundry object.

[TextDocument class]

Performs a copy operation on a selected object. To copy selected text to the Clipboard, use CopySelection

[BaseTable]

Copies a range of selected cells in a table object.

Syntax

[Objectreference].Division.Copy(Name,[ParentName,][BeforeNeighbor,][NeighborName])

[Objectreference].Foundry.Copy([ObjectType,][ObjectName,][Foundry Type,] [p4,][NewName])

[Objectreference].BaseTable.Copy([Temporary])

[Objectreference].TextDocument.Copy(Name,[ParentName,][BeforeNeighbor,][NeighborName])

Parameters

[Division and TextDocument classes]

Name

Data type is String.

ParentName

Data type is String. Optional parameter.

BeforeNeighbor

Data type is Bool. Optional parameter. Default is False.

NeighborName

Data type is String. Optional parameter.

[Foundry class]

ObjectType

Specifies what type of object you are copying from this Foundry object. Data type is Variant. The value of this parameter must be one of the strings below or its code equivalent. Default is \$LwpCopyObjectTypeLayout.

<u>Value</u>	<u>Effect</u>
\$LwpCopyObjectTypeContents (170)	Copies any Content object. You must specify the Content object's name in the ObjectName parameter.
\$LwpCopyObjectTypeDivision (171)	Copies any Division object. You must specify the Division object's name in the ObjectName parameter.
\$LwpCopyObjectTypeLayout (168)	Copies any Layout object. You must specify the Layout object's name in the ObjectName parameter.
\$LwpCopyObjectTypeStyle (169)	Copies objects which were created from any of the following classes: CellLayout, CharacterStyle, FrameLayout, PageLayout, ParagraphStyle, TableLayout. You can specify a single object using the ObjectName and P4 parameters, or you can copy all objects created from a single class by leaving ObjectName blank and specifying the class in P4.

ObjectName

The name of the object you are copying. Required parameter for copying Content, Layout, and Division objects. When copying SmartMaster-derived objects, you can specify one object by name or leave this parameter empty to copy all SmartMaster-derived objects. Data type is String.

FoundryType

Indicates the Foundry object to which you are copying the object. Data type is Variant. Optional parameter. The value must be one of the strings below or its code equivalent. Default is \$LwpFoundryTypeDocument.

<u>Value</u>	<u>Effect</u>
\$LwpFoundryTypeApplication (346)	Copies to AppFoundry property.
\$LwpFoundryTypeDocument (345)	Copies to Division.Foundry property.
\$LwpFoundryTypeTemporary (347)	Copies to TempFoundry property.

p4

When you use \$LwpCopyObjectTypeStyle as the value for ObjectType, you can specify further which class of object you are copying. You can choose objects created from one of the following classes: CellLayout, CharacterStyle, FrameLayout, PageLayout, ParagraphStyle, TableLayout. If you leave this blank and use a null string ("") as the value of ObjectName, all objects created from any of these classes are copied. Data type is Variant. The value must be one of the strings below or its numeric equivalent (shown in parentheses).

<u>Value</u>	<u>Style Object Copied</u>
7	Page
8	Frame
23	Table
25	Cell
35	Paragraph
39	Character
107	Header
108	Footer
676	DropCap

NewName

Allows you to specify a new name for the copied object. Data type is String. Optional parameter.

[BaseTable class]

Temporary

An Integer value of -1 or 0 indicating whether the selection is copied to the temporary Foundry (-1), or to the Clipboard (0). You can use the LotusScript constants of True (-1) and False (0) as the value for this parameter.

Return value

[Division, TextDocument classes]

[Foundry class]

A String value indicating the name of the object created. No return value is given when ObjectType is \$LwpCopyObjectTypeStyle.

[BaseTable class]

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Word Pro: CreateDataFile method

{button ,AL('H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL('H_CREATEDATAFILE_METHOD_EXSCRIPT',1)} [See example](#)

Creates a new Merge data file including the records and fields you specify. The currently active document becomes the Merge document for the new data file. Equivalent to clicking the Create New button in the Mail Merge Assistant dialog box to display the Create Data File dialog box.

Syntax

[objectreference].CreateDataFile(Delimiters,FieldNames,[IsAscii], [FullPath])

Parameters

Delimiters

A String expression specifying the characters or symbols which you want Word Pro to use to delineate between fields and records. The value of this parameter can be any two alphanumeric characters. Default is "~|". Tilde is the separator for fields and the bar is the separator for records. If *IsAscii* is True, the value of this parameter must be "Fixed length ASCII."

FieldNames

A String expression representing the names of the fields in the new data file. Each field name is separated by the first delimiter character specified in the *Delimiters* parameter.

IsAscii

Indicates whether or not the data file is fixed length ASCII. Data type is Integer. The legal values for this parameter are -1 and 0 but you may use the LotusScript constants True (-1) and False (0). Optional parameter. Default is False.

FullPath

An optional String expression representing the path and filename of the new data file.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Word Pro: CreateDivision method

{button ,AL('H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL('H_CREATEDIVISION_METHOD_EXSCRIPT',1)} [See example](#)

Creates a new division in a document using a SmartMaster or another file. Equivalent to choosing Create - Division.

Syntax

[objectreference].CreateDivision (MasterFileName [, FileType] [, DivisionLocation] [,Parent] [,NeighborName])

Parameters

MasterFileName

A String expression which specifies the name of a SmartMaster file or an external file from which you want to create the new division.

FileType

An optional String expression which specifies the file type of the file used in creating the division. Word Pro automatically recognizes and imports all of the file types listed below. Use this parameter only if the file you are using is not one of these file types:

DCA/RFT	Lotus Manuscript 2.x	MS Word for Windows 1.0
DIF	Lotus Organizer 1.x	MS Word for Windows 2.0
DisplayWrite	Lotus Word Pro	MS Word for Windows 6.0
HTML	Lotus Word Pro SmartMaster	MS Word for Windows95 7.0
Lotus 1-2-3	MS Excel	MS WordPad 1.0
Lotus 1-2-3 for OS/2	MS Excel 3.0	OfficeWriter 4,5,6
Lotus 1-2-3 R3	MS Excel 4.0	Rich Text Format(RTF)
Lotus 1-2-3 R4,5	MS Excel 5.0	SAMNA Word
Lotus 1-2-3 R6	MS Excel 7.0	WordPerfect 5.0
Lotus Ami Pro	MS Windows Write 3.x	WordPerfect 5.1
Lotus Ami Pro 3.x Macro	MS Word for DOS 3,4,5,6	WordPerfect 6.x
Lotus Ami Pro 3.x Styles	MS Word for OS/2	WordStar 2000 R3

DivisionLocation

Indicates where you want the new division inserted. Data type is Variant, which allows the value of this parameter to be one of the three division locations listed below or its numeric equivalent (in parentheses). Default is \$LwpDivLocInsertAtInsertionPt.

\$LwpDivLocInsertBeforeCurrentdiv (184) Inserts the new division before the currently active division.

\$LwpDivLocInsertAfterCurrentdiv (185) Inserts the new division after the currently active division.

\$LwpDivLocInsertAtInsertionPt (186) Inserts the new division at the insertion point. All items that fall before the insertion point remain part of the active division. All items after the insertion point become part of the new division.

Note If the insertion point is in a table cell or a frame, Word Pro splits the contents of the cell or frame, leaving the items before the insertion point intact and moving the items after the insertion point into the new division. Items outside the cell or frame are not affected and remain in the original division.

Parent

An optional String expression representing the internal name of the division which you want to become the parent of the new division.

NeighborName

An optional String expression representing the name of the division which you want to become the neighbor of the new division.

Return value

A String expression which represents the internal name of the new division.

For more information on division names in LotusScript, see [Overview: Division names in LotusScript](#)

Usage

Word Pro: CreateDropCap method

{button ,AL('H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL('H_CREATEDROPCAP_METHOD_EXSCRIPT',1)} [See example](#)

Creates a DropCap at the insertion point. Each DropCap is comprised of several objects and is accessible through the DropCaps property on a Foundry object.

Syntax

[objectreference].CreateDropCap(NumLines, Position)

Parameters

NumLines

Specifies the height of the DropCap in lines of text. Data type is Integer. If you specify 3 lines, the DropCap will be as high as three lines of text in the current paragraph style.

Position

An Integer which allows you to specify the position of the DropCap. There are three legal values for this parameter:

<u>Value</u>	<u>Effect</u>
1 = Below	Aligns the top edge of the DropCap with the top edge of the first line of text, and places the DropCap inside the page margin so that the remaining lines of text flow around the DropCap.
2 = Above	Aligns the bottom edge of the DropCap with the bottom edge of the first line of text, and places the DropCap inside the page margin so that the preceding lines of text flow above the DropCap.
3 = Beside	Equivalent to choosing Below, except the DropCap is placed in the margin beside the text so there is no text flowing around the DropCap.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Word Pro: CreateExternalDivision method

{button ,AL('H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL('H_CREATEEXTERNALDIVISION_METHOD_EXSCRIPT',1)} [See example](#)

Creates a new division which is linked to the contents of an external file. If you edit the contents of the division, you can save your changes to the external file. Similarly, if you edit the external file, the external division reflects those changes the next time you open the Word Pro document.

Syntax

[objectreference].CreateExternalDivision(Path [, FileType][, DivisionLocation][, Parent][, NeighborName])

Parameters

Path

A String expression representing the drive and directory location of the external file you want to use as the source for the external division.

FileType

An optional String expression which specifies the file type of the file specified in the Path parameter. Word Pro automatically recognizes and imports many file types. Use this parameter only if the file specified in the Path parameter is not one of these file types:

DCA/RFT	Lotus Manuscript 2.x	MS Word for Windows 1.0
DIF	Lotus Organizer 1.x	MS Word for Windows 2.0
DisplayWrite	Lotus Word Pro	MS Word for Windows 6.0
HTML	Lotus Word Pro SmartMaster	MS Word for Windows95 7.0
Lotus 1-2-3	MS Excel	MS WordPad 1.0
Lotus 1-2-3 for OS/2	MS Excel 3.0	OfficeWriter 4,5,6
Lotus 1-2-3 R3	MS Excel 4.0	Rich Text Format(RTF)
Lotus 1-2-3 R4,5	MS Excel 5.0	SAMNA Word
Lotus 1-2-3 R6	MS Excel 7.0	WordPerfect 5.0
Lotus Ami Pro	MS Windows Write 3.x	WordPerfect 5.1
Lotus Ami Pro 3.x Macro	MS Word for DOS 3,4,5,6	WordPerfect 6.x
Lotus Ami Pro 3.x Styles	MS Word for OS/2	WordStar 2000 R3

DivisionLocation

A String or Integer value which indicates where you want the new division inserted. Data type is Variant, which allows the value of this parameter to be one of the three division locations listed below or its numeric equivalent (in parentheses). Default is \$LwpDivLocInsertAtInsertionPt.

\$LwpDivLocInsertBeforeCurrentdiv (184) Inserts the new division before the currently active division.

\$LwpDivLocInsertAfterCurrentdiv (185) Inserts the new division after the currently active division.

\$LwpDivLocInsertAtInsertionPt (186) Inserts the new division at the insertion point. All items which fall before the insertion point remain part of the active division. All items after the insertion point become part of the new division.

Note If the insertion point is in a table cell or a frame, Word Pro splits the contents of the cell or frame, leaving the items before the insertion point intact and moving the items after the insertion point into the new division. Items outside the cell or frame are not affected and remain in the original division.

Parent

An optional String expression representing the internal name of the division which you want to become the parent of the external division.

NeighborName

An optional String expression representing the name of the division which you want to become the neighbor of the external division.

Return value

A String expression which represents the internal name of the external division.

For more information on division names in LotusScript, see [Overview: Division names in LotusScript](#)

Usage

Word Pro displays the contents of the external source file within the external division in the Word Pro document. The contents are displayed in a format which approximates the way in which they would be displayed in the source application.

You can specify where you want Word Pro to place the new external division by using the DivisionLocation, Parent, or NeighborName parameters.

Word Pro: CreateFrame method

{button ,AL('H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL('H_CREATEFRAME_METHOD_EXSCRIPT',1)} [See example](#)

Creates a frame object in a document. Equivalent to choosing Create - Frame.

Syntax

[objectreference].CreateFrame([UseDefault],[FrameStyle],[Width],[Height]

Parameters

UseDefault

Allows you to use the default frame style (True). Data type is Integer. The legal values for this parameter are -1 and 0 but you may use the LotusScript constants True (-1) and False (0). Optional parameter. Default is True.

FrameStyle

A String expression specifying the name of the frame style for the new frame.

Width

An Integer that specifies the width of the new frame in Twips. This parameter is only needed if you do not use the default frame style.

Height

An Integer that specifies the height of the new frame in Twips. This parameter is only needed if you do not use the default frame style.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Word Pro: CreateFromBitmap method

{button ,AL('H_GRAPHICOLEBJECT_CLASS;H_GRAPHIC_CLASS;H_OLEOBJECT_CLASS',0)} [See list of classes](#)

Creates a graphic object from a bitmap.

Syntax

[objectreference].CreateFromBitmap(BitmapHandle,IsDeviceIndependent)

Parameters

BitmapHandle

Data type is Long.

IsDeviceIndependent

Data type is Integer.

Return value

Integer.

Usage

Word Pro: CreateFromClipBrd method

{button ,AL(^H_GRAPHIC_CLASS',0)} [See list of classes](#)

Creates a graphic object from the clipboard.

Syntax

[objectreference].CreateFromClipBrd([ClipBrdFormat])

Parameters

ClipBrdFormat

Data type is String. Optional parameter.

Return value

Integer.

Usage

Word Pro: CreateFromDataObject method

{button ,AL(^H_GRAPHIC_CLASS',0)} [See list of classes](#)

Creates a graphic from a data object.

Syntax

[objectreference].CreateFromDataObject(DataObjPtr,[ClipBrdFormat])

Parameters

DataObjPtr

Data type is Long.

ClipBrdFormat

An optional String expression representing the format of the Clipboard.

Return value

Integer.

Usage

Word Pro: CreateFromMetafile method

{button ,AL('H_GRAPHICOLEBJECT_CLASS;H_GRAPHIC_CLASS;H_OLEOBJECT_CLASS',0)} [See list of classes](#)

Creates a graphic from a metafile.

Syntax

[objectreference].CreateFromMetafile(MetaFileHandle,EnhancedMetafile)

Parameters

MetaFileHandle

Data type is Long.

EnhancedMetafile

Data type is Integer.

Return value

Integer.

Usage

Word Pro: CreateGlossaryEntry method

{button ,AL('H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL('H_CREATEGLOSSARYENTRY_METHOD_EXSCRIPT',1)} [See example](#)

Inserts a glossary entry for the current selection in the specified Glossary file. Equivalent to choosing Edit - Glossary and inserting a new glossary entry for the current selection.

Syntax

[objectreference].CreateGlossaryEntry(GlossFilePath, KeyName)

Parameters

GlossFilePath

A String expression which specifies the path and name of the Glossary file (.GLS) to which you are adding this entry.

Keyname

A String expression you want to use as the abbreviation for the new glossary entry. Equivalent to the string value you provide as the "Glossary entry name" in the Glossary dialog box.

Return value

None.

Usage

Word Pro: CreateGlossary method

{button ,AL('H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL('H_CREATEGLOSSARY_METHOD_EXSCRIPT',1)} [See example](#)

Creates a Glossary object at the insertion point in the active document. Word Pro displays the Glossary object in parallel columns with "Name" at the top of the first column and "Contents" at the top of the second column.

Syntax

[objectreference].CreateGlossary()

Parameters

None.

Return value

None.

Usage

Word Pro: CreateGraphic method

{button ,AL('H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL('H_CREATEGRAPHIC_METHOD_EXSCRIPT',1)} [See example](#)

Creates a graphic object in a Word Pro document. You can use the parameters to specify the type of graphic object and whether you want yourself or Word Pro to draw the graphic's frame.

Syntax

[objectreference].CreateGraphic(AFIDClassName, ScratchOutFrame)

Parameters

AFIDClassName

A String expression which specifies the type of graphic you are creating. There are three types of graphics which are native to Word Pro: equations, drawings, and charts.

<u>Value</u>	<u>Effect</u>
WordProEqn	Use if you want to create an Equation graphic. When you do so, Word Pro switches to equation mode and places the insertion point in the equation frame.
WordProDraw	Use to create a drawing using the Word Pro drawing tools. When you do so, Word Pro switches to drawing mode and displays the drawing tools.
LotusChart (or WordProChart)	If you are using the 32-bit or OS/2 version of Word Pro, you can use LotusChart to create a chart graphic. If you are using the 16-bit version of Word Pro, you must use WordProChart. The effect is the same. Either value launches the charting tool for Word Pro, which allows you to create a chart for the new chart graphic frame.

ScratchOutFrame

An Integer value which indicates whether you want to draw the new graphic frame by hand or let Word Pro draw the frame based on a frame style. If you want to draw the frame yourself, use a True value (-1) for this parameter. If you want Word Pro to draw the frame based on an existing style, use a False value (0).

Return value

None.

Usage

When ScratchOutFrame is set to False, Word Pro checks the type of graphic you are creating and uses the default style for that type of graphic frame. If no default style exists for the type of graphic you are creating, Word Pro uses a predefined style to create that frame.

The drawing and chart graphics share the same default frame style. Equation graphics have their own default frame style.

Word Pro: CreateNewButton method

{button ,AL('H_STATUSBAR_CLASS',0)} [See list of classes](#)

{button ,AL('H_CREATENEWBUTTON_METHOD_EXSCRIPT',1)} [See example](#)

Creates a new button in the status bar. After the button is created, the InvalidateWholeBar method should be called to repaint the status bar.

Syntax

[objectreference].CreateNewButton(ParentButtonId, InsertAfterButtonId, ButtonWidth, ButtonType)

Parameters

ParentButtonId

Data type is Long. Required parameter. Value should be 0 unless you are creating a child button.

InsertAfterButtonId

Data type is Long. Required parameter. Value of 0 causes the button to be added to the beginning of the bar.

ButtonWidth

Data type is Integer. Required parameter.

ButtonType

Data type is Variant, which allows the value of this parameter to be one of the constants listed below or its hexadecimal equivalent (in parentheses). You can combine these constants when you want Word Pro to combine the features listed below. Use the OR operator to combine constants.

Note If you use the text instead of the hexadecimal equivalent as a constant, you must include WPBITMSK.LSS in your script.

<u>Value</u>	<u>Effect</u>
LwpButtonBehaviorClickable (&H8)	Allows the button to be left-clicked.
LwpButtonBehaviorCollapsible (&H10)	Allows the button to shrink or grow so that the status bar can fill up the window. Only one is allowed per status bar. Word Pro's collapsible button is the Date/Time button.
LwpButtonBehaviorContainer (&H20)	Allows the button to contain child buttons.
LwpButtonBehaviorLeftclick (&H8)	Allows the button to be left-clicked.
LwpButtonBehaviorPopup (&H4)	Allows the button to pop up a list of alternatives.
LwpButtonBehaviorThermometer (&H80000)	Allows the button to display a thermometer graphic.
LwpButtonCanBeDepressed (&H40000)	Allows the button to stay depressed.
LwpButtonContentsCenterAligned (&H80)	Allows the button contents to be center-aligned.
LwpButtonContentsGray (&H200)	Allows the button contents to be grayed.
LwpButtonContentsHilited (&H400)	Allows the button contents to be highlighted (red in Word Pro).
LwpButtonContentsLeftAligned (&H40)	Allows the button contents to be left-aligned.
LwpButtonContentsRightAligned (&H100)	Allows the button contents to be right-aligned.
LwpButtonHasAutorepeat (&H4000)	Allows the button to repeat a command.
LwpButtonHasUpdownCtrl (&H20000)	Allows the button to have up/down control.
LwpButtonNoTextFromHost (&H800)	Allows the button to keep its user-defined text without changing; in other words, the text on this button is never going to require text from a host.
LwpButtonReserved (&H8000)	
LwpButtonSpacer (&H10000)	A spacer status bar button.
LwpButtonSupportDbClick (&H2000)	Allows the button to respond to a double-click.
LwpButtonSupportRightClick (&H1000)	Allows the button to support a right mouse click.

LwpButtonTypeGraphics (&H2)

Allows the button to display a graphic.

LwpButtonTypeText (&H1)

Allows the button to display text.

Note **Return value**

String.

Usage

Use this method to add a new button to the status bar object. Some of the ButtonType parameters can be combined together. For example, you can combine the left-click, text, and left-aligned values to allow the button to respond to a left-mouse click and display left-aligned text. You cannot combine certain parameters that are in obvious conflict with each other, such as left-align and right-align.

Word Pro: CreateNew method

{button ,AL('H_GRAPHICOLEBJECT_CLASS;H_GRAPHIC_CLASS;H_OLEOBJECT_CLASS',0)} [See list of classes](#)

Creates a new graphic in a document.

Syntax

[objectreference].CreateNew(NameOfGraphicType)

Parameters

NameOfGraphicType

Data type is String.

Return value

Integer.

Usage

```
'Example: Abilities property
'This example sets the editing rights of all editors except the current editor to "Not
Allowed".
Forall DocEditor In .ActiveDocument.EditorManager.Editors
    If DocEditor.Name <> .ActiveDocument.EditorManager.CurrentEditor.Name Then
        DocEditor.Abilities =$LwpEditAbilEditingNotAllowed
    End If
End Forall
```

```
'Example: Accelerator property
' This example assigns function keys to paragraph styles in the document.
' Assign Default Text to F2
.Foundry.ParagraphStyles.Item("Default Text").Accelerator = 2
' Assign Number List to F3
.Foundry.ParagraphStyles.Item("Number List").Accelerator = 3
' Reset Number List to not use a function key
.Foundry.ParagraphStyles.Item("Number List").Accelerator = 0
```

'Example: AccessibilityOn property

'This example asks the user whether to enable the use of the system caret

'while Word Pro is running, then sets the appropriate option.

```
stat = MessageBox ("Do you want to enable the accessibility option?", 36, "Example  
Script")
```

```
If stat = 6 Then ' user said yes
```

```
    .Application.Preferences.AccessibilityOn = True
```

```
Else
```

```
    .Application.Preferences.AccessibilityOn = False
```

```
End If
```

```
'Example: Activate method
' This example prompts for a document name then cycles through all open
' documents and attempts to find the one requested.
' RUNTIME DEPENDENCIES: You must have a document open for this script to work.

Dim DocName As String
DocName = Inputbox("Enter the name of the document to activate:")
Forall Doc In .Documents
    If Doc.Name = DocName Then ' show the document and get out
        Doc.Activate
        .ActiveDocWindow.Show
        exit sub
    End If
End Forall
' tell the user we couldn't find the document
Msgbox "Unable to find the document " & DocName, MB_OK, "Example Script"
```

```
'Example: ActiveDocument property
''This example retrieves the name of the currently active Word
Pro document,
'assigns it to a variable and prints the name in the Output panel
of the Script Editor.
'You must have a document open for this script to work.
'Paste this script into Sub Main in the Globals section.
Dim DocName as String
DocName = CurrentApplication.ActiveDocument.Name
Print DocName
```

```
'Example: ActiveDocWindow property
''This example retrieves the text displayed in the title bar of
the currently active Word Pro document's window.
'It assigns that text to a variable and prints the text in the
Output panel of the Script Editor.
'You must have a document open for this script to work.
'Paste this script into Sub Main in the Globals section.
Dim DocWindowCaption as String
DocWindowCaption = CurrentApplication.ActiveDocWindow.Name
Print DocWindowCaption
```



```
'Example: ActualName property
Sub Main
Print "======"
Forall x In .division.foundry.paragraphstyles
    Print x.font.ActualName & " = " & x.font.size
End Forall
Forall x In .Division.foundry.paragraphstyles
    x.font.FontName = "Arial"
End Forall
Print "-----"
Forall x In .Division.foundry.paragraphstyles
    Print x.font.ActualName & " = " & x.font.size
End Forall
End Sub
```

```
'Example: AddAccelerators method
' This example assigns the 'AcceleratorTest' subroutine to the run every time ' the
Control and 1 keys are simultaneously pressed.
' RUNTIME DEPENDENCIES: You must have a document open for this script to work.
' The AcceleratorTest subroutine must also be included in your script module.

Const SHIFT = &H1000
Const CTRL = &H4000
Const ALT = &H2000
Dim FunctionName As String
Dim Key As Integer
' name of the function we want to run when key is pressed
FunctionName = .ActiveDocument.FullName & "!AcceleratorTest"
' assign the key; add value for CTRL to indicate it's with the Ctrl Key.
Key = CTRL + Asc("1")
.ApplicationWindow.Accelerators.AddAccelerators FunctionName, Key, 0,True

Sub AcceleratorTest
' this is the sub which will run when the user hits the shortcut key.
Messagebox "Hello There"
End Sub
```

```
'Example: AddBookmark method
' This example creates a new bookmark named 'NewBookMark' in the active
' division of the current document.
' RUNTIME DEPENDENCIES: You must have a document open for this script to work.
```

```
Dim MarkerName as String
' First, specify the location of the bookmark
MarkerName = .Mark($LwpMarkerTypeBookmark)
' Then add it to the list of bookmarks
.Division.BookmarkManager.AddBookmark "NewBookMark", MarkerName
```

```
'Example: AddDivisionToPrint method
' This example prints the current division to the default printer.
' RUNTIME DEPENDENCIES: You must have a document open for this script to work.
```

```
Dim CurrentDivName As String
```

```
CurrentDivName = .Division.Name
```

```
.ActiveDocument.GetPageRange $LwpPresentationTypeLayout, 100
```

```
.ActiveDocument.PrintSettings.ClearDivisionList
```

```
.ActiveDocument.PrintSettings.AddDivisionToPrint CurrentDivName
```

```
.ActiveDocument.PrintSettings.SelectedPages = "1-9999"
```

```
.ActiveDocument.PrintSettings.PrintRange = $LwpPrintRangeSelectedDivisions
```

```
.ActiveDocument.PrintSettings.Copies = 1
```

```
.ActiveDocument.PrintSettings.PrintPagesFrom = 1
```

```
.ActiveDocument.PrintSettings.PrintPagesTo = 1
```

```
.ActiveDocument.PrintSettings.PrintPageType = $LwpPrintPageEvenAndOddPages
```

```
.ActiveDocument.PrintSettings.Collate = False
```

```
.ActiveDocument.PrintSettings.OutputToFile = False
```

```
'Example: AddDivision method
' This example creates two parent divisions each containing two child divisions.
' RUNTIME DEPENDENCIES: You must have a document open for this script to work.
Dim ParentDivName As String
Dim ChildDivName As String
Dim ParentDiv As String
ParentDivName = "Parent "
ChildDivName = "Child "

For ParentDivCount = 1 To 2
    ParentDiv = .ActiveDocument.AddDivision(ParentDivName & CStr(ParentDivCount))
    For ChildDivCount = 1 To 2
        .ActiveDocument.AddDivision ChildDivName & CStr(ChildDivCount), ParentDiv
    Next
Next
```

```
'Example: AddEditorManager method
' This example adds a new editor with read only rights to the current
' document.
' RUNTIME DEPENDENCIES: You must have a document open for this script to work.

Dim NewEditorName As String
Dim NewEditorInitials As String

NewEditorName = "Lotus User"
NewEditorInitials = "LU"
.ActiveDocument.EditorManager.AddEditorManager NewEditorName, NewEditorInitials
.ActiveDocument.EditorManager.Editors(NewEditorName).Abilities =
$LwpEditAblEditingNotAllowed
```

```
'Example: AddField method
' This example adds a new field named 'ExampleField' for the current document
' where it is then inserted.
' RUNTIME DEPENDENCIES: You must have a document open for this script to work.

.ActiveDocument.DocInfo.FieldManager.AddField "ExampleField", "Some data for
ExampleField ", False
.InsertDocInfo $LwpDocVarField, "ExampleField"
```

```
'Example: AddIcon method
Dim IcnMgr As IconBarManager
Set IcnMgr = .ApplicationWindow.IconBarManager

' Select the icon to add.
' For this example to work, the icon and the script must already be linked
IcnMgr.SelectCustomIcon "c:\lotus\wordpro\icons\mynew.bmp","c:\lotus\wordpro\scripts\
mynew.lss"
' Add the icon to the Internet icon bar
IcnMgr.IconBars("Internet Tools").AddIcon 2
' Set this bar to show in its context
IcnMgr.IconBars("Internet Tools").ShowInContext = True
IcnMgr.IconBars("Internet Tools").Show
' This will force a redraw of IconBars
IcnMgr.ShowIconBars
```


'Example: AddIndexAllEntry method

'This example inserts some text in the current document, and marks all
'occurrences of the word 'entry' as an index entry.

'The primary term is "Entry", and the secondary term
'is "Index". The page number of the entry will be printed.

.Text.InsertText "This is some sample text. This will be an index entry. "

.Text.InsertText "This will be some more sample text. And finally, another index
entry."

' add index entry with 'vice-versa' option selected

.AddIndexAllEntry "entry", "Index ""entry"" # ""index"" ", "Index ""index"" #
""entry"" "

'add just a plain entry to index all occurrences of 'sample'

.AddIndexAllEntry "sample", "Index ""sample"" # "" "" ", ""

'Example: AddIndexEntry method

'This example inserts some text in the current document, and marks the last
'word as an index entry. The primary term is "Entry", and the secondary term
'is "Index". The page number of the entry will be printed.

.Text.InsertText "This will be an index entry."

.AddIndexEntry "Index "Entry" # "Index" "

'Example: AddOutlineSequenceItem method

'This example creates an outline style sequence, and adds three items to it.

'First, create the sequence, name it, and make it empty

```
NewSequence = .Division.Foundry.Create($LwpFoundryCreateTypeOutlineseq)
.Division.Foundry.OutlineStyleSequences(NewSequence).Name = "Typical Outline"
.Division.Foundry.OutlineStyleSequences("Typical Outline").Clear
.Division.Foundry.OutlineStyleSequences("Typical Outline").Heading = 1
```

' Now, add a sequence item for the Heading 1 paragraph style

```
.Division.Foundry.OutlineStyleSequences("Typical Outline").AddOutlineSequenceItem
"Heading 1"
```

' And set it's position

```
.Division.Foundry.OutlineStyleSequences("Typical
Outline").OutlineSeqItems("Heading 1").Position = 1
```

'Add Heading 2 and Heading 3 paragraph styles

```
.Division.Foundry.OutlineStyleSequences("Typical Outline").AddOutlineSequenceItem
"Heading 2"
```

```
.Division.Foundry.OutlineStyleSequences("Typical
Outline").OutlineSeqItems("Heading 2").Position = 2
```

```
.Division.Foundry.OutlineStyleSequences("Typical Outline").AddOutlineSequenceItem
"Heading 3"
```

```
.Division.Foundry.OutlineStyleSequences("Typical
Outline").OutlineSeqItems("Heading 3").Position = 3
```

'Example: AddSectionTabs method

' This example adds a quick division after the current division.

' RUNTIME DEPENDENCIES: You must have a document open for this script to work.

.ApplicationWindow.SectionTabs.AddSectionTabs

```
'Example: AddSmartCorrect method
' This example adds a new entry to the SmartCorrect list
' RUNTIME DEPENDENCIES: You must have a document open for this script to work.
Language = "English (United States)"
With .Application.SmartCorrects(Language)
    .AddSmartCorrect "lts", "Lotus Development"
End With
```

'Example: AddStringToList method

'This example creates a SmartFill list named "Day Parts", and adds

'four items to the list.

'Note that 1033 is the numeric value for the American language code.

```
.Application.SmartFill(1033).CreateEmptyList "Day Parts", True
```

```
.Application.SmartFill(1033).AddStringToList "Day Parts", "Morning"
```

```
.Application.SmartFill(1033).AddStringToList "Day Parts", "Afternoon"
```

```
.Application.SmartFill(1033).AddStringToList "Day Parts", "Evening"
```

```
.Application.SmartFill(1033).AddStringToList "Day Parts", "Night"
```

```
'Example: AddTOCEntry method
' This example adds the current sentence to the table of contents.
' RUNTIME DEPENDENCIES: You must have a document open for this script to work.
Dim TOCName As String
Dim TOCCommand As String

TOCName = .Text.GetText($LwpGetObjectTypesentence,False)
TOCCommand = "TOC 1" & """" & TOCName & """"
.AddTOCEntry TOCCommand
```

```
'Example: Add method
' This example prompts for a word to be added to the user dictionary.  The
' word is inserted into the current document, selected and then added to the
' user dictionary.
' RUNTIME DEPENDENCIES: You must have a document open for this script to work.

Dim NewWord As String
' get the word from the user
NewWord = Inputbox ("Enter a word to add to the user dictionary:", "Example Script", "")
If NewWord <> "" Then
    'if he typed something, put it in the document, then select it
    .Type NewWord
    .Text.MoveToStart $LwpLocationTypeWord
    .SelectWord
    ' and add it to the dictionary.
.Text.Add $LwpAddTypeSpell
End If
```



```
'Example: AdjustShade method
' This example inserts 20 words into the current document and shades(selects) the
' last 5 words. After the message box is closed, 5 characters to the right
' of the insertion point are unshaded.
' RUNTIME DEPENDENCIES: You must have a document open for this script to work.

Dim WordNumber as Integer
For WordNumber = 1 To 20
    .Text.InsertText "Word" & Format$(WordNumber) & " "
Next
.Text.Shade $LwpLocationTypeWord,$LwpNavigateDirectionLeft,5
MessageBox "Click OK to adjust the shading.",MB_OK,"Example Script"
.Text.AdjustShade $LwpWhichSideLeft, 5, $LWPAdjustUnitCharacter
```

'Example: Afid property

'This example creates a chart in the current document, then changes the
'attributes of the chart.

```
.CreateGraphic "LotusChart", False
```

```
.Graphic.Afid.IsStacked = 1
```

```
.Graphic.Afid.IsDepthEffect = 1
```

'Example: AlignmentType property

' This example sets up numeric alignment, with the decimal point 1/2 inch from the right margin.

.Text.Alignment.**AlignmentType** = \$LwpAlignmentTypeNumericright

.Text.Alignment.Position = 720

'Example: Alignment property

' This example sets up numeric alignment, with the decimal point 1/2 inch from the right margin.

.Text.**Alignment**.AlignmentType = \$LwpAlignmentTypeNumericright

.Text.**Alignment**.Position = 720

```

'Example: AlignStyleName property
'This example determines the hierarchy of the current paragraph style's
'alignment attribute. If alignment is inherited, the user is given a
'chance to make it local.

If .Text.ParagraphStyle.AlignStyleName = "" Then 'this style's alignment is local
    MsgBox .Text.ParagraphStyle.Name & " stores its alignment locally.", MB_OK,
    "Example Script"
Else ' Give user a chance to make it local
    Stat = MsgBox (.Text.ParagraphStyle.Name & " inherits its alignment from "
    & .Text.ParagraphStyle.AlignStyleName & ". Do you want to make it local?", 36,
    "Example Script")
    If stat = 6 Then 'user said yes
        'get the current hierarchy, and add alignment to the local attrs.
        StyleAttrs = .Text.ParagraphStyle.Definition + &H10
        .SetStyle $LwpStyleTypeParagraph, .Text.ParagraphStyle.Name, StyleAttrs
        ' set the alignment style name to the empty string.
        .Text.ParagraphStyle.AlignStyleName = ""
    End If
End If

```

'Example: AllBorders property

'This example creates a frame, and then sets double ruled lines 40 twips wide
'for all borders of the frame.

.CreateFrame

.Frame.Layout.BorderLines.**AllBorders**.Pattern = \$LtsBorderPatternDouble

.Frame.Layout.BorderLines.**AllBorders**.WidthInTwips = 40

```
'Example: AllowAlternateVerification property
'This example asks the user whether to allow less strict verification of the
'editor's identity than what was specified in the TeamSecurity dialog box,
' and then sets the appropriate option.

stat = Messagebox ("Do you want to permit alternate TeamSecurity verification?", 36,
"Example Script")
If stat = 6 Then ' user said yes
    .ActiveDocument.DocControl.AllowAlternateVerification = True
Else
    .ActiveDocument.DocControl.AllowAlternateVerification = False
End If
```

```
'Example: All property
'This example creates a new document, types a paragraph of text, and indents the
paragraph 1 inch.
.NewDocument
For i = 1 To 20
    .type "Indention test "
Next
.Text.Indent.All = 1440 ' indent everything one inch
MessageBox "Click OK to revert to the style indents.", MB_OK, "Example Script"
.Text.Indent.RevertToStyle
```



```
'Example: Always property
'This example asks the user whether paragraph spacing should be applied to all
'paragraphs, even those at the beginning or end of pages,
' and then sets the appropriate option.

stat = MessageBox ("Do you want to add paragraph spacing every time?", 36, "Example
Script")
If stat = 6 Then ' user said yes
    .Text.Spacing.Always = True
Else
    .Text.Spacing.Always = False
End If
```

'Example: AmountOfSpaceAbove property

'This example sets the spacing above the current paragraph to a custom value
'of 12 points.

.Text.Spacing.TypeAbove = \$LwpSpacingTypeCustom

.Text.Spacing.**AmountOfSpaceAbove** = 240 ' 240 twips = 12 points

'Example: AmountOfSpaceBelow property

'This example sets the spacing above the current paragraph to a custom value
'of 12 points.

.Text.Spacing.TypeBelow = \$LwpSpacingTypeCustom

.Text.Spacing.**AmountOfSpaceBelow** = 240 ' 240 twips = 12 points

'Example: Amount property

'This example sets the line spacing to custom, then sets the amount of spacing
'to 18 points.

.Text.Spacing.Type = \$LwpSpacingTypeCustom

.Text.Spacing.RevertToStyle \$LwpSpacingPropertyNumber

.Text.Spacing.**Amount** = 360 ' 360 twips = 18 points.

'Example: AmtTether property

'This example creates a frame, and ties the frame's anchor point
'to the center of the frame relative to the upper left of the page.

```
.NewFrame 2223, 1406, 2539, 2972, "Default Frame"
```

```
.Frame.Layout.AmtTether = $LwpWhereTypeMiddle
```

```
.Frame.Layout.AmtToTetherFrom = $LwpWhereTypeUpperLeft
```

'Example: AmtToRotateContent property

'This example imports a bitmap image, then rotates it.

```
PicToImport = .Path & "\\helpbutn.bmp"
```

```
.ImportGraphic PicToImport, ".bmp", False, False, "Default Graphic/OLE"
```

```
' Rotate graphic 45 degrees clockwise; 270 degrees counterclockwise
```

```
.Layout.AmtToRotateContent = 2700
```

```
' now rotate 150 degrees clockwise; 210 degrees counterclockwise
```

```
.Layout.AmtToRotateContent = 2100
```

'Example: AmtToTetherFrom property
'This example creates a frame, and ties the frame's anchor point
'to the center of the frame relative to the upper left of the page.

```
.NewFrame 2223, 1406, 2539, 2972, "Default Frame"  
.Frame.Layout.AmtTether = $LwpWhereTypeMiddle  
.Frame.Layout.AmtToTetherFrom = $LwpWhereTypeUpperLeft
```

'Example: Anchor method

' This example creates a frame and then anchors the frame 'In text'.

' RUNTIME DEPENDENCIES: You must have a document open for this script to work.

.CreateFrame False, "Default Frame", 1440, 1440

**.Frame.Anchor \$LwpAnchorWhereDivisionInfo, \$LwpConditionTypeAllpages,
\$LwpRelativeTypeLytInline**

'Example: AnswerMsgBox method

' This example uses the AnswerMsgBox function to keep the warning message

' about saving files in another format from displaying.

' RUNTIME DEPENDENCIES: You must have a document open for this script to work.

.AnswerMsgBox \$LwpMsgboxReplyNo

.SaveAs "Test.txt", , "Text"

```
'Example: AnyEdits method
' This example prints the number of edits made by the current editor to the
' active document.
' RUNTIME DEPENDENCIES: You must have a document open for this script to work.

Dim UserName As String
UserName = .Preferences.UserName
UName = InputBox ("Specify the name of the editor to check:", "Example Script",
UserName)
If UName = "" then exit sub
If .ActiveDocument.AnyEdits(UName) then
    MessageBox Uname & " has made edits to this document.", MB_OK, "Example Script"
Else
    MessageBox Uname & " has not made edits to this document.", MB_OK, "Example
Script"
```

'Example: AnyNumber property

'This example creates a table, then sets the numeric format of positive numbers
'to contain a leading dollar sign.

```
.CreateTable False, "Default Table", 3, 2
```

```
.Type "123.456"
```

```
.Table.CurrentCell.NumericFormat.AnyNumber.Prefix = "$"
```

```
.Type "[Tab]"
```

```
'Example: AppFoundry property
' This script copies some text to the foundry clipboard,
' then lists the name of the paragraph style used in the copied text.
.Type "This is some text to be copied"
.SelectParagraph
.CopySelection
Forall ParaStyles In .AppFoundry.ParagraphStyles
    Print ParaStyles.Name
End Forall
```

```
'Example: ApplicationWindow property
'This example first prints the value of the Height property on
' the ApplicationWindow object to the Output Panel.
' Then it sets Height to half of its previous value and prints the new value to the
Output Panel.
'You must have a document open for this script to work.
'Paste this script into Sub Main in the Globals section.
Print .ApplicationWindow.Height
.ApplicationWindow.Height = (.5 * .ApplicationWindow.Height)
Print .ApplicationWindow.Height
```

```
'Example: Application property
'This script gets the name of the currently active document by going
'through the Application property of whatever object has the focus.
'It prints the name to the Output panel in the Script Editor.
'You must have a document open to run this script.
'Paste this code example in Sub Main and run it.
DIM AppName As String
AppName = .Application.ActiveDocument.Name
Print AppName
```

```
'Example: ApplyAgreementWithHereThere property
'This example asks the user whether to check for agreement between here and there
'and their objects in grammar check, and then sets the appropriate option.

stat = MessageBox ("Do you want to check for here/there agreement?", 36, "Example
Script")
If stat = 6 Then ' user said yes
    .ApplicationWindow.UserInterfacePrefs.GrammarOptions.ApplyAgreementWithHereThere =
True
Else
    .ApplicationWindow.UserInterfacePrefs.GrammarOptions.ApplyAgreementWithHereThere =
False
End If
```

'Example: ApplyArchaicExpressions property

'This example asks the user whether to check for use of outdated expressions in
'grammar check, and then sets the appropriate option.

```
stat = MessageBox ("Do you want to check for archaic expressions?", 36, "Example  
Script")
```

```
If stat = 6 Then ' user said yes
```

```
    .ApplicationWindow.UserInterfacePrefs.GrammarOptions.ApplyArchaicExpressions =  
True
```

```
Else
```

```
    .ApplicationWindow.UserInterfacePrefs.GrammarOptions.ApplyArchaicExpressions =  
False
```

```
End If
```


'Example: ApplyArticleAgreement property

'This example asks the user whether to check for article agreement in

'grammar check, and then sets the appropriate option.

```
stat = MessageBox ("Do you want to check for article agreement?", 36, "Example  
Script")
```

```
If stat = 6 Then ' user said yes
```

```
    .ApplicationWindow.UserInterfacePrefs.GrammarOptions.ApplyArticleAgreement = True
```

```
Else
```

```
    .ApplicationWindow.UserInterfacePrefs.GrammarOptions.ApplyArticleAgreement = False
```

```
End If
```

```
'Example: ApplyBadPrepositions property
'This example asks the user whether to check for proper use of prepositions in
'grammar check, and then sets the appropriate option.

stat = MessageBox ("Do you want to check for correct preposition use?", 36, "Example
Script")
If stat = 6 Then ' user said yes
    .ApplicationWindow.UserInterfacePrefs.GrammarOptions.ApplyBadPrepositions = True
Else
    .ApplicationWindow.UserInterfacePrefs.GrammarOptions.ApplyBadPrepositions = False
End If
```

```
'Example: ApplyCapitalizationCheck property
'This example asks the user whether to check for correct capitalization
'grammar check, and then sets the appropriate option.

stat = MessageBox ("Do you want to check for proper capitalization?", 36, "Example
Script")
If stat = 6 Then ' user said yes
    .ApplicationWindow.UserInterfacePrefs.GrammarOptions.ApplyCapitalizationCheck =
True
Else
    .ApplicationWindow.UserInterfacePrefs.GrammarOptions.ApplyCapitalizationCheck =
False
End If
```

'Example: ApplyClauseErrors property

'This example asks the user whether to check for errors in clause punctuation in
'grammar check, and then sets the appropriate option.

```
stat = MessageBox ("Do you want to check for clause errors?", 36, "Example Script")
If stat = 6 Then ' user said yes
    .ApplicationWindow.UserInterfacePrefs.GrammarOptions.ApplyClauseErrors = True
Else
    .ApplicationWindow.UserInterfacePrefs.GrammarOptions.ApplyClauseErrors = False
End If
```

'Example: ApplyCliches property

'This example asks the user whether to check for cliches in

'grammar check, and then sets the appropriate option.

```
stat = MessageBox ("Do you want to check for cliches?", 36, "Example Script")
```

```
If stat = 6 Then ' user said yes
```

```
    .ApplicationWindow.UserInterfacePrefs.GrammarOptions.ApplyCliches = True
```

```
Else
```

```
    .ApplicationWindow.UserInterfacePrefs.GrammarOptions.ApplyCliches = False
```

```
End If
```

```
'Example: ApplyCommonlyConfusedWords property
'This example asks the user whether to check for words that are commonly
'confused with others in grammar check, and then sets the appropriate option.

stat = MessageBox ("Do you want to check for confusing words?", 36, "Example Script")
If stat = 6 Then ' user said yes
    .ApplicationWindow.UserInterfacePrefs.GrammarOptions.ApplyCommonlyConfusedWords =
True
Else
    .ApplicationWindow.UserInterfacePrefs.GrammarOptions.ApplyCommonlyConfusedWords =
False
End If
```

'Example: ApplyConsecutiveNouns property

'This example sets the grammar check option for flagging consecutive nouns in the document to four nouns in a row.

.ApplicationWindow.UserInterfacePrefs.GrammarOptions.**ApplyConsecutiveNouns** = 2

'Example: ApplyContractions property

'This example asks the user whether to check for contractions in

'grammar check, and then sets the appropriate option.

```
stat = MessageBox ("Do you want to check for contractions?", 36, "Example Script")
```

```
If stat = 6 Then ' user said yes
```

```
    .ApplicationWindow.UserInterfacePrefs.GrammarOptions.ApplyContractions = True
```

```
Else
```

```
    .ApplicationWindow.UserInterfacePrefs.GrammarOptions.ApplyContractions = False
```

```
End If
```


'Example: ApplyOpenUsage property

'This example asks the user whether to check for usage errors in

'grammar check, and then sets the appropriate option.

```
stat = MessageBox ("Do you want to check for overused phrases?", 36, "Example Script")
```

```
If stat = 6 Then ' user said yes
```

```
    .ApplicationWindow.UserInterfacePrefs.GrammarOptions.ApplyOpenUsage = True
```

```
Else
```

```
    .ApplicationWindow.UserInterfacePrefs.GrammarOptions.ApplyOpenUsage = False
```

```
End If
```

'Example: ApplyOverUsedPhrases property

'This example asks the user whether to check for expressions that are overused in
'grammar check, and then sets the appropriate option.

```
stat = MessageBox ("Do you want to check for overused phrases?", 36, "Example Script")
If stat = 6 Then ' user said yes
    .ApplicationWindow.UserInterfacePrefs.GrammarOptions.ApplyOverusedPhrases = True
Else
    .ApplicationWindow.UserInterfacePrefs.GrammarOptions.ApplyOverusedPhrases = False
End If
```

'Example: ApplyPassiveVerbErrors property

'This example asks the user whether to check for overuse of passive verbs in

'grammar check, and then sets the appropriate option.

```
stat = MessageBox ("Do you want to check for passive verbs?", 36, "Example Script")
```

```
If stat = 6 Then ' user said yes
```

```
    .ApplicationWindow.UserInterfacePrefs.GrammarOptions.ApplyPassiveVerbErrors = True
```

```
Else
```

```
    .ApplicationWindow.UserInterfacePrefs.GrammarOptions.ApplyPassiveVerbErrors =
```

```
False
```

```
End If
```

'Example: ApplyPunctuationErrors property

'This example asks the user whether to check for common punctuation errors in

'grammar check, and then sets the appropriate option.

```
stat = MessageBox ("Do you want to check for punctuation errors?", 36, "Example  
Script")
```

```
If stat = 6 Then ' user said yes
```

```
    .ApplicationWindow.UserInterfacePrefs.GrammarOptions.ApplyPunctuationErrors = True
```

```
Else
```

```
    .ApplicationWindow.UserInterfacePrefs.GrammarOptions.ApplyPunctuationErrors =
```

```
False
```

```
End If
```

```
'Example: ApplyStockPhrase property
'This example asks the user whether to check for stock phrases in
'grammar check, and then sets the appropriate option.

stat = MessageBox ("Do you want to check for stock phrases?", 36, "Example Script")
If stat = 6 Then ' user said yes
    .ApplicationWindow.UserInterfacePrefs.GrammarOptions.ApplyStockPhrase = True
Else
    .ApplicationWindow.UserInterfacePrefs.GrammarOptions.ApplyStockPhrase = False
End If
```

```
'Example: ApplyUngrammaticalExpressions property
'This example asks the user whether to check for non standard expressions in
'grammar check, and then sets the appropriate option.

stat = MessageBox ("Do you want to check for ungrammatical expressions?", 36, "Example
Script")
If stat = 6 Then ' user said yes
    .ApplicationWindow.UserInterfacePrefs.GrammarOptions.ApplyUnGrammaticalExpressions
= True
Else
    .ApplicationWindow.UserInterfacePrefs.GrammarOptions.ApplyUnGrammaticalExpressions
= False
End If
```

```
'Example: ApplyWordCompoundingCheck property
'This example asks the user whether to check for correct hyphenation of compound
'words in grammar check, and then sets the appropriate option.

stat = MessageBox ("Do you want to check for correct compounding?", 36, "Example
Script")
If stat = 6 Then ' user said yes
    .ApplicationWindow.UserInterfacePrefs.GrammarOptions.ApplyWordCompoundingCheck =
True
Else
    .ApplicationWindow.UserInterfacePrefs.GrammarOptions.ApplyWordCompoundingCheck =
False
End If
```

'Example: ApplyWordyPhraseCheck property

'This example asks the user whether to check for wordy phrases in

'grammar check, and then sets the appropriate option.

```
stat = MessageBox ("Do you want to check for wordy phrases?", 36, "Example Script")
```

```
If stat = 6 Then ' user said yes
```

```
    .ApplicationWindow.UserInterfacePrefs.GrammarOptions.ApplyWordyPhraseCheck = True
```

```
Else
```

```
    .ApplicationWindow.UserInterfacePrefs.GrammarOptions.ApplyWordyPhraseCheck = False
```

```
End If
```


'Example: Action property

'This example creates a Click Here block, and sets the behavior of the block
'to insert typed text.

```
Dim ClickHereName as String
```

```
ClickHereName = .InsertClickHere()
```

```
.Division.Foundry.ClickHeres(ClickHereName).Action = 1
```

Word Pro: Abilities property

{button ,AL('H_EDITOR_CLASS',0)} [See list of classes](#)

{button ,AL('H_ABILITIES_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-Write) Controls which version of a document a specific editor can edit.

Data Type

Data type is [Variant](#), which allows the value of this property to be one of the constants listed below or its numeric equivalent (in parentheses).

Syntax

abilitiesvalue = [objectreference].Abilities

[objectreference].Abilities = abilitiesvalue

Legal values

<u>Value</u>	<u>Effect</u>
\$LwpEditAbilEditAsAnnotationOnly (2519)	An assigned editor can only create annotations in the document.
\$LwpEditAbilEditCurrentOrNewVer (223)	An assigned editor can work in the current document or any new versions of the document.
\$LwpEditAbilEditCurrentVersionOnly (221)	An assigned editor can only work in the current version of the document and cannot edit previous versions of the document.
\$LwpEditAbilEditingNotAllowed (220)	A specific editor cannot edit the document.
\$LwpEditAbilEditNewVersionsOnly (222)	An assigned editor can only work in a new version of a document that is automatically created when the document opens. The editor can review previous versions of the document.

Usage

Use any one of the above values to determine what version of a document an editor can edit.

Word Pro: AboutToReturnToOriginator property

{button ,AL(^H_MAILROUTING_CLASS',0)} [See list of classes](#)

(Read-only)

This language element has not yet been defined. You can check for a newer Help file on the Lotus web site.

Data Type

Integer

Syntax

abouttoreturntooriginatorvalue = [objectreference].AboutToReturnToOriginator

Legal values

This property is read-only. The value of this property cannot be set by a script.

Usage

Word Pro: AbsoluteOn property

{button ,AL('H_LAYOUT_CLASS;H_CELLGROUPLAYOUT_CLASS;H_CELLLAYOUT_CLASS;H_CONNECTEDLAY
OUT_CLASS;H_COLUMNGROUPLAYOUT_CLASS;H_FOOTERLAYOUT_CLASS;H_FRAMELAYOUT_CLASS;H
_FRAMEGROUPLAYOUT_CLASS;H_DROP_CAPLAYOUT_CLASS;H_GROUPLAYOUT_CLASS;H_HEADERLAY
OUT_CLASS;H_NOTELAYOUT_CLASS;H_PAGELAYOUT_CLASS;H_ROWGROUPLAYOUT_CLASS;H_ROWLA
YOUT_CLASS;H_RUBYLAYOUT_CLASS;H_SUPERTABLEGROUPLAYOUT_CLASS;H_SUPERTABLELAYOUT
_CLASS;H_TABLELAYOUT_CLASS;H_ENDNOTELAYOUT_CLASS;H_FOOTNOTELAYOUT_CLASS;H_TABLE
HEADINGLAYOUT_CLASS;H_TOCSUPERTABLELAYOUT_CLASS',0)} [See list of classes](#)

(Read-write) A Boolean value that indicates whether the top left corner of the layout (the origin) is positioned relative to the PageLayout origin, or relative to the parent layout's origin. Default is False. If this property is set, the coordinates are specified in the AbsoluteXPos and AbsoluteYPos properties.

Data Type

Data type is [Integer](#)

Syntax

[objectreference].AbsoluteOn = absoluteonvalue

absoluteonvalue = [objectreference].AbsoluteOn

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Allows import filters to process relative page coordinates.

Word Pro: AbsoluteTextOrientation property

```
{button ,AL(^H_BASECONTAINER_CLASS;H_CELLCONTAINER_CLASS;H_FRAMECONTAINER_CLASS;H_NOTECONTAINER_CLASS;H_RUBYCONTAINER_CLASS;H_DROPDOWNCONTAINER_CLASS;H_PAGECONTAINER_CLASS;H_SUBPAGECONTAINER_CLASS;H_SUPERPAGECONTAINER_CLASS;H_ROWCONTAINER_CLASS;H_SUPERTABLECONTAINER_CLASS;H_TABLECONTAINER_CLASS;H_PARALLELCOLSCONTAINER_CLASS;H_TABLEONLYCONT_CLASS',0)} See list of classes
```

(Read-only) Returns the orientation of text in a container, taking into account the orientation of any parent containers.

Data Type

Data type is Integer. Always contains one of the values listed below, under Legal values.

Syntax

absolutetextorientationvalue = [objectreference].AbsoluteTextOrientation

Legal values

The legal values for this property are listed below:

Value	Effect
0	The text, in relation to its container plus any parent containers, is oriented from left to right and top to bottom.
1	The text, in relation to its container plus any parent containers, is oriented from top to bottom and right to left.
2	The text, in relation to its container plus any parent containers, is oriented from right to left and bottom to top.
3	The text, in relation to its container plus any parent containers, is oriented from bottom to top and left to right.

Usage

This property takes into account the orientation of all parent containers. For example, frame A is oriented from right to left and bottom to top. Frame A contains table cell B which is also oriented from right to left and bottom to top. Text contained in table cell B appears on the screen as if it is oriented from left to right and top to bottom. Therefore, this property contains a 0 value, which indicates that the text is oriented from left to right and top to bottom, taking into account the orientation of all of its parent containers.

Word Pro: AbsoluteXPos property

```
{button ,AL(^H_LAYOUT_CLASS;H_CELLGROUPLAYOUT_CLASS;H_CELLLAYOUT_CLASS;H_CONNECTEDLAY  
OUT_CLASS;H_COLUMNGROUPLAYOUT_CLASS;H_FOOTERLAYOUT_CLASS;H_FRAMELAYOUT_CLASS;H  
_FRAMEGROUPLAYOUT_CLASS;H_DROP_CAPLAYOUT_CLASS;H_GROUPLAYOUT_CLASS;H_HEADERLAY  
OUT_CLASS;H_NOTELAYOUT_CLASS;H_PAGELAYOUT_CLASS;H_ROWGROUPLAYOUT_CLASS;H_ROWLA  
YOUT_CLASS;H_RUBYLAYOUT_CLASS;H_SUPERTABLEGROUPLAYOUT_CLASS;H_SUPERTABLELAYOUT  
_CLASS;H_TABLELAYOUT_CLASS;H_ENDNOTELAYOUT_CLASS;H_FOOTNOTELAYOUT_CLASS;H_TABLE  
HEADINGLAYOUT_CLASS;H_TOCSUPERTABLELAYOUT_CLASS',0)} See list of classes
```

(Read-write) The top left position X coordinate for a layout.

Data Type

Long

Syntax

[objectreference].AbsoluteXPos = absolutexposvalue

absolutexposvalue = [objectreference].AbsoluteXPos

Legal values

Data type is Long but the unit of measurement used for this property is Twips. There are 1440 Twips per inch.

Usage

The AbsoluteOn property uses this property to set the layout object's top left position. This property allows import filters to process relative page coordinates.

Word Pro: AbsoluteYPos property

```
{button ,AL(^H_LAYOUT_CLASS;H_CELLGROUPLAYOUT_CLASS;H_CELLLAYOUT_CLASS;H_CONNECTEDLAY  
OUT_CLASS;H_COLUMNGROUPLAYOUT_CLASS;H_FOOTERLAYOUT_CLASS;H_FRAMELAYOUT_CLASS;H  
_FRAMEGROUPLAYOUT_CLASS;H_DROP_CAPLAYOUT_CLASS;H_GROUPLAYOUT_CLASS;H_HEADERLAY  
OUT_CLASS;H_NOTELAYOUT_CLASS;H_PAGELAYOUT_CLASS;H_ROWGROUPLAYOUT_CLASS;H_ROWLA  
YOUT_CLASS;H_RUBYLAYOUT_CLASS;H_SUPERTABLEGROUPLAYOUT_CLASS;H_SUPERTABLELAYOUT  
_CLASS;H_TABLELAYOUT_CLASS;H_ENDNOTELAYOUT_CLASS;H_FOOTNOTELAYOUT_CLASS;H_TABLE  
HEADINGLAYOUT_CLASS;H_TOCSUPERTABLELAYOUT_CLASS',0)} See list of classes
```

(Read-write) The top left position Y coordinate for a layout.

Data Type

Long

Syntax

[objectreference].AbsoluteYPos = absoluteyposvalue

absoluteyposvalue = [objectreference].AbsoluteYPos

Legal values

Data type is Long but the unit of measurement used for this property is Twips. There are 1440 Twips per inch.

Usage

The AbsoluteOn property uses this property to set the layout object's top left position. This property allows import filters to process relative page coordinates.

Word Pro: Accelerators property

{button ,AL('H_APPLICATIONWINDOW_CLASS',0)} [See list of classes](#)

(Read-only) An instance of the Accelerators class which is a short-cut key assignment for any Word Pro script or Ami Pro macro.

Data Type

[Accelerators](#)

Syntax

acceleratorsvalue = [objectreference].Accelerators

Legal values

Always contains an instance of the Accelerators class.

Usage

With the methods in this class, you can add or remove any accelerator key assignment.

Word Pro: Accelerator property

{button ,AL(^H_CHARACTERSTYLE_CLASS;H_LAYOUT_CLASS;H_CELLGROUPLAYOUT_CLASS;H_CELLLAYO
UT_CLASS;H_CONNECTEDLAYOUT_CLASS;H_COLUMNGROUPLAYOUT_CLASS;H_FOOTERLAYOUT_CLA
SS;H_FRAMELAYOUT_CLASS;H_FRAMEGROUPLAYOUT_CLASS;H_DROPPLAYOUT_CLASS;H_GROUPL
AYOUT_CLASS;H_HEADERLAYOUT_CLASS;H_NOTELAYOUT_CLASS;H_PAGELAYOUT_CLASS;H_ROWGR
OUPPLAYOUT_CLASS;H_ROWLAYOUT_CLASS;H_RUBYLAYOUT_CLASS;H_SUPERTABLEGROUPLAYOUT_
CLASS;H_SUPERTABLELAYOUT_CLASS;H_TABLELAYOUT_CLASS;H_ENDNOTELAYOUT_CLASS;H_FOOT
NOTELAYOUT_CLASS;H_TABLEHEADINGLAYOUT_CLASS;H_TOCSUPERTABLELAYOUT_CLASS;H_PARAG
RAPHSTYLE_CLASS;')0)} [See list of classes](#)

{button ,AL(^H_ACCELERATOR_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Represents the function key that a specific style is assigned to, when function keys represent styles instead of CycleKeys.

Data Type

Integer.

Syntax

acceleratorvalue = [objectreference].Accelerator

[objectreference].Accelerator = acceleratorvalue

Legal values

Integer. The values are:

<u>Value</u>	<u>Effect</u>
0	This style is not assigned to a function key.
2 - 9, 11-12	This style is assigned to the appropriate function key.

Usage

Only one style may be defined as a specific function key assignment.

Word Pro: AccessibilityOn property

{button ,AL('H_PREFERENCES_CLASS',0)} [See list of classes](#)

{button ,AL('H_ACCESSIBILITYON_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write)

Enables Word Pro to use a screenreader.

Data Type

[Integer](#)

Syntax

accessibilityonvalue = [objectreference].AccessibilityOn

[objectreference].AccessibilityOn = accessibilityonvalue

Legal values

The data type for this property is Integer. The legal values for this property are -1 and 0 but you may use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Equivalent to choosing File - Word Pro Preferences, clicking the Enable panel, and selecting "Accessibility Options. "

Word Pro: ActiveDocument property

{button ,AL('H_APPLICATIONWINDOW_CLASS;H_APPLICATION_CLASS;H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL('H_ACTIVEDOCUMENT_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-only) Returns the TextDocument object which is active when this property is read. Its contents depend on which document is active at the time. You can use the global variable, CurrentDocument, instead of this property.

This is a current context property in the WPAApplication class.

Data Type

[TextDocument](#)

Syntax

activedocumentvalue = [objectreference].ActiveDocument

Legal values

Always contains an instance of the TextDocument class.

Usage

WPAApplication - Use this property from the WPAApplication object when you want to access the active Word Pro document.

ApplicationWindow - Use this property from the ApplicationWindow object when you want to access the active Word Pro document.

Word Pro: ActiveDocWindow property

{button ,AL('H_APPLICATION_CLASS;H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL('H_ACTIVEDOCWINDOW_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-only) Returns the currently active DocWindow object. The DocWindow object provides access to the most basic functions of a document window, including the position, size, and appearance of a particular document's window. This is a current context property. Its contents depend on which document is active at the time.

Data Type

[DocWindow](#)

Syntax

activedocwindowvalue = [objectreference].ActiveDocWindow

Legal values

Always contains an instance of the DocWindow class.

Usage

Use this property when you want to access the currently active Word Pro document's window.

Word Pro: Active property

{button ,AL(^H_WINDOW_CLASS;H_APPLICATIONWINDOW_CLASS;H_STATUSBAR_CLASS;H_DOCWINDOW_CLASS',0)} [See list of classes](#)

(Read-only)

[StatusBar]

A flag that indicates if the status bar is usable.

[ApplicationWindow]

A flag that indicates if the application window is active.

Data Type

Integer

Syntax

activevalue = [objectreference].Active

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

[Application Window]

Default is None.

Usage

IconBarManager - Not valid.

ApplicationWindow - Not implemented for ApplicationWindow in Word Pro 97 or in Word Pro for OS/2 Warp 4.

Word Pro: ActualEnumName property

{button ,AL(^H_FONT_CLASS',0)} [See list of classes](#)

(Read-only)

Data Type

Data type is String.

Syntax

actualenumnamevalue = [objectreference].ActualEnumName

Legal values**Usage**

Word Pro: ActualName property

{button ,AL('H_FONT_CLASS',0)} [See list of classes](#)

{button ,AL('H_ACTUALNAME_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-only)

Data Type

[String](#)

Syntax

actualnamevalue = [objectreference].ActualName

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Word Pro: Address1 property

{button ,AL('H_PREFERENCES_CLASS',0)} [See list of classes](#)

(Read-write)

Data Type

String

Syntax

address1value = [objectreference].Address1

[objectreference].Address1 = address1value

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Word Pro: Address2 property

{button ,AL('H_PREFERENCES_CLASS',0)} [See list of classes](#)

(Read-write)

Data Type

String

Syntax

address2value = [objectreference].Address2

[objectreference].Address2 = address2value

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Word Pro: AfidClassName property

{button ,AL(`H_GRAPHIC_CLASS',0)} [See list of classes](#)

{button ,AL(`H_AFIDCLASSNAME_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-only)

Data Type

Data type is String.

Syntax

afidclassnamevalue = [objectreference].AfidClassName

Legal values**Usage**

{button ,AL(`H_AFID_PROPERTY_MEMDEF;H_CREATEGRAPHIC_METHOD_MEMDEF',0)} [See related topics](#)

Word Pro: Afid property

{button ,AL('H_GRAPHIC_CLASS',0)} [See list of classes](#)

{button ,AL('H_AFID_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-only) Contains an Afid object.

Data Type

Variant

Syntax

afid = [objectreference].Afid

Legal values

This property is either empty or contains an Afid object.

Usage

If the graphic object from which this property is called is not a Lotus Chart, this property will be empty.

If the graphic object from which this property is called is a Lotus Chart, you can access the chart properties and methods through this property.

Word Pro: AlignmentChar property

{button ,AL('H_ALIGNMENT_CLASS',0)} [See list of classes](#)

(Read-write)

Data Type

[Integer](#)

Syntax

alignmentcharvalue = [objectreference].AlignmentChar

[objectreference].AlignmentChar = alignmentcharvalue

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Word Pro: AlignmentType property

{button ,AL(`H_ALIGNMENT_CLASS',0)} [See list of classes](#)

{button ,AL(`H_ALIGNMENTTYPE_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write)

Data Type

Variant (Enumerated)

AlignmentType

Syntax

alignmenttypevalue = [objectreference].AlignmentType

[objectreference].AlignmentType = alignmenttypevalue

Legal values

\$LtsAlignmentHorizCenter (1056964611)

\$LtsAlignmentJustify (1056964613)

\$LtsAlignmentLeft (1056964609)

\$LtsAlignmentRight (1056964610)

\$LtsAlignmentSmart (1056964612)

\$LwpAlignmentTypeAlignRevert (8)

\$LwpAlignmentTypeJustifyall (5)

\$LwpAlignmentTypeNumericleft (6)

\$LwpAlignmentTypeNumericright (7)

Usage

{button ,AL(`H_POSITION_PROPERTY_MEMDEF',0)} [See related topics](#)

Word Pro: Alignment property

{button ,AL(^H_GRAPHICOLEBJECT_CLASS;H_GRAPHIC_CLASS;H_OLEOBJECT_CLASS;H_CLICKHERE_CLASS;H_TEXTMARKER_CLASS;H_PARAGRAPHSTYLE_CLASS;H_TEXT_CLASS;H_FRAMECAPTIONOPTIONS_CLASS',0)} [See list of classes](#)

{button ,AL(^H_ALIGNMENT_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-only) An Alignment object used for controlling the alignment attributes of a paragraph within a Text, TextMarker, or ClickHere object.

Data Type

[Alignment](#)

Syntax

alignmentvalue = [objectreference].Alignment

Legal values

Always contains an instance of the Alignment class.

Usage

Word Pro: AlignStyleName property

{button ,AL('H_PARAGRAPHSTYLE_CLASS',0)} [See list of classes](#)

{button ,AL('H_ALIGNSTYLENAME_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write)

Data Type

[String](#)

Syntax

alignstylevalue = [objectreference].AlignStyleName

[objectreference].AlignStyleName = alignstylevalue

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

{button ,AL('H_SETSTYLE_METHOD_MEMDEF',0)} [See related topics](#)

Word Pro: Align property

{button ,AL(^H_FONT_CLASS',0)} [See list of classes](#)

(Read-write)

Data Type

Data type is Integer.

Syntax

[objectreference].Align = alignvalue

alignvalue = [objectreference].Align

Legal values

Usage

Word Pro: AllBorders property

{button ,AL('H_BORDERLINES_CLASS;H_GUTTER_CLASS;H_BETWEENLINES_CLASS',0)} [See list of classes](#)

{button ,AL('H_ALLBORDERS_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-only) Allows you to simultaneously access all of an object's border objects.

Data Type

[Border](#)

Syntax

allbordersvalue = [objectreference].AllBorders

Legal values

Always contains an instance of the Border class.

Usage

Use this property in order to simultaneously access an object's BottomBorder, LeftBorder, RightBorder, and TopBorder objects.

Word Pro: AllowAlternateVerification property

{button ,AL('H_DOCCONTROL_CLASS',0)} [See list of classes](#)

{button ,AL('H_ALLOWALTERNATEVERIFICATION_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Provides alternate ways for Word Pro to verify authorized users of a document.

Data Type

[Integer](#)

Syntax

allowalternateverificationvalue = [objectreference].AllowAlternateVerification

[objectreference].AllowAlternateVerification = allowalternateverificationvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Enables you to structure a hierarchy of verification types to verify an assigned user of a document. The verification types are e-mail login, operating system login, and the Word Pro user name. For example, suppose you set up an e-mail login for assigned users of a document. However, a user with an operating system login wants to gain access to that document. When the user attempts to login, Word Pro denies access.

If you set this property to True, Word Pro displays the other two verification types: operating system login and the Word Pro user name. The user can now gain access when the operating system login displays. If you set this property to False, Word Pro does not display the other verification types, thereby denying access to the user.

Equivalent to choosing File - TeamSecurity and selecting "Allow alternate verification" on the Access panel.

Word Pro: AllowListEdit property

{button ,AL('H_CLICKHERE_CLASS',0)} [See list of classes](#)

{button ,AL('H_ALLOWLISTEDIT_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Allows users to enter Click Here keyword values that are not in the keyword list.

Data Type

[Integer](#).

Syntax

[objectreference].AllowListEdit = allowlisteditvalue

allowlisteditvalue = [objectreference].AllowListEdit

Legal values

The legal values for this property are -1 and 0 but you may use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

This property is equivalent to the "Allow values not in list" option in the Design Keyword Format dialog box for Keyword Click Here Blocks.

Word Pro: AllowListMultiValues property

{button ,AL('H_CLICKHERE_CLASS',0)} [See list of classes](#)

{button ,AL('H_ALLOWLISTMULTIVALUES_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Allows multiple values to be selected in a Click Here Keyword list.

Data Type

[Integer](#).

Syntax

[objectreference].AllowListMultiValues = allowlistmultivaluesvalue

allowlistmultivaluesvalue = [objectreference].AllowListMultiValues

Legal values

The data type for this property is Integer. The legal values for this property are -1 and 0 but you may use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

This property is equivalent to the "Allow Multi-values" option in the Design Keyword Format dialog box for Keyword Click Here Blocks.

Word Pro: AllowResizeWhenCrop property

```
{button ,AL('H_LAYOUT_CLASS;H_CELLGROUPLAYOUT_CLASS;H_CELLLAYOUT_CLASS;H_CONNECTEDLAY  
OUT_CLASS;H_COLUMNGROUPLAYOUT_CLASS;H_FOOTERLAYOUT_CLASS;H_FRAMELAYOUT_CLASS;H  
_FRAMEGROUPLAYOUT_CLASS;H_DROP_CAPLAYOUT_CLASS;H_GROUPLAYOUT_CLASS;H_HEADERLAY  
OUT_CLASS;H_NOTELAYOUT_CLASS;H_PAGELAYOUT_CLASS;H_ROWGROUPLAYOUT_CLASS;H_ROWLA  
YOUT_CLASS;H_RUBYLAYOUT_CLASS;H_SUPERTABLEGROUPLAYOUT_CLASS;H_SUPERTABLELAYOUT  
_CLASS;H_TABLELAYOUT_CLASS;H_ENDNOTELAYOUT_CLASS;H_FOOTNOTELAYOUT_CLASS;H_TABLE  
HEADINGLAYOUT_CLASS;H_TOCSUPERTABLELAYOUT_CLASS',0)} See list of classes
```

(Read-only)

This language element has not yet been defined. You can check for a newer Help file on the Lotus web site.

Data Type

Integer

Syntax

```
allowresizewhencropvalue = [objectreference].AllowResizeWhenCrop
```

Legal values

This property is read-only. The value of this property cannot be set by a script.

Usage

Word Pro: All property

{button ,AL('H_INDENT_CLASS;H_RELATIVEINDENT_CLASS',0)} [See list of classes](#)

{button ,AL('H_ALL_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) The amount of indent for all lines of text.

Data Type

Long

Syntax

allvalue = [objectreference].All

[objectreference].All = allvalue

Legal values

Data type is Long but the unit of measurement used for this property is Twips.

Usage

This property is equivalent to the "All lines from left" setting in the Indent Options dialog box.

Word Pro: AlternateName property

{button ,AL('H_FONT_CLASS',0)} [See list of classes](#)

(Read-write)

Data Type

String

Syntax

alternatenamevalue = [objectreference].AlternateName

[objectreference].AlternateName = alternatenamevalue

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Word Pro: Always property

{button ,AL('H_SPACING_CLASS',0)} [See list of classes](#)

{button ,AL('H_ALWAYS_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Determines whether spacing options are applied always (-1) or only when not at breaks (0).

Data Type

[Integer](#)

Syntax

alwaysvalue = [objectreference].Always

[objectreference].Always = alwaysvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Word Pro: AmikakeName property

{button ,AL('H_PARAGRAPHSTYLE_CLASS',0)} [See list of classes](#)

(Read-write) The name of the text background object for the Asian language versions of Word Pro. If you are using an English language version of Word Pro, this property is not available.

Data Type

String

Syntax

amikakevalue = [objectreference].AmikakeName

[objectreference].AmikakeName = amikakevalue

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Word Pro: AmikakeType property

{button ,AL(^H_AMIKAKE_CLASS',0)} [See list of classes](#)

(Read-write)

This property is available only in Asian language versions of Word Pro.

Data Type

Integer

Syntax

[objectreference].AmikakeType = amikaketypevalue

amikaketypevalue = [objectreference].AmikakeType

Legal values

Usage

Word Pro: Amikake property

{button ,AL(^H_CHARACTERSTYLE_CLASS;H_CLICKHERE_CLASS;H_TEXTMARKER_CLASS;H_PARAGRAPHS
STYLE_CLASS;H_TEXT_CLASS',0)} [See list of classes](#)

(Read-only) Holds a text background object for the Asian language versions of Word Pro. If you are using an English language version of Word Pro, this property is not available.

Data Type

[Amikake](#)

Syntax

amikakevalue = [objectreference].Amikake

Legal values

Always contains an instance of the Amikake class.

Usage

Word Pro: AmountOfSpaceAboveLine property

{button ,AL(^H_SPACING_CLASS',0)} [See list of classes](#)

(Read-write)

Data Type

Long

Syntax

[objectreference].AmountOfSpaceAboveLine = amountofspaceabovelinevalue

amountofspaceabovelinevalue = [objectreference].AmountOfSpaceAboveLine

Legal values

Data type is Long but the unit of measurement used for this property is Twips. There are 1440 Twips per inch.

Usage

Word Pro: AmountOfSpaceAbove property

{button ,AL('H_SPACING_CLASS',0)} [See list of classes](#)

{button ,AL('H_AMOUNTOFSPACEABOVE_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write)The amount of spacing to use when spacing above paragraph is set to custom.

Data Type

[Long](#)

Syntax

amountofspaceabovevalue = [objectreference].AmountOfSpaceAbove

[objectreference].AmountOfSpaceAbove = amountofspaceabovevalue

Legal values

Data type is Long but the unit of measurement used for this property is [Twips](#). There are 1440 Twips per inch.

Usage

Word Pro: AmountOfSpaceBelow property

{button ,AL('H_SPACING_CLASS',0)} [See list of classes](#)

{button ,AL('H_AMOUNTOFSPACEBELOW_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write)The amount of spacing when spacing below a paragraph is set to custom.

Data Type

[Long](#)

Syntax

amountofspacebelowvalue = [objectreference].AmountOfSpaceBelow

[objectreference].AmountOfSpaceBelow = amountofspacebelowvalue

Legal values

Data type is Long but the unit of measurement used for this property is [Twips](#). There are 1440 Twips per inch.

Usage

Word Pro: Amount property

{button ,AL('H_SPACING_CLASS',0)} [See list of classes](#)

{button ,AL('H_AMOUNT_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) The amount of custom line spacing to use for the a Spacing object.

Data Type

Long

Syntax

amountvalue = [objectreference].Amount

[objectreference].Amount = amountvalue

Legal values

Data type is Long but the unit of measurement used for this property is Twips. There are 1440 Twips per inch.

Usage

For the Spacing object in the Text class, this property is equivalent to setting a custom line spacing value in the Text Alignment properties panel of the InfoBox.

Word Pro: AmtTether property

```
{button ,AL('H_LAYOUT_CLASS;H_CELLGROUPLAYOUT_CLASS;H_CELLLAYOUT_CLASS;H_CONNECTEDLAYOUT_CLASS;H_COLUMNGROUPLAYOUT_CLASS;H_FOOTERLAYOUT_CLASS;H_FRAMELAYOUT_CLASS;H_FRAMEGROUPLAYOUT_CLASS;H_DROPDOWNLAYOUT_CLASS;H_GROUPLAYOUT_CLASS;H_HEADERLAYOUT_CLASS;H_NOTELAYOUT_CLASS;H_PAGELAYOUT_CLASS;H_ROWGROUPLAYOUT_CLASS;H_ROWLAYOUT_CLASS;H_RUBYLAYOUT_CLASS;H_SUPERTABLEGROUPLAYOUT_CLASS;H_SUPERTABLELAYOUT_CLASS;H_TABLELAYOUT_CLASS;H_ENDNOTELAYOUT_CLASS;H_FOOTNOTELAYOUT_CLASS;H_TABLEHEADINGLAYOUT_CLASS;H_TOCSUPERTABLELAYOUT_CLASS',0)} See list of classes
```

```
{button ,AL('H_AMTTETHER_PROPERTY_EXSCRIPT',1)} See example
```

(Read-write) Indicates the location of a layout object's knot. The knot is the position on the frame from which it is anchored.

Data Type

Data type is [Variant](#), which allows the value of this property to be one of the constants listed below or its numeric equivalent (in parentheses).

Syntax

[objectreference].AmtTether = amttethervalue

amttethervalue = [objectreference].AmtTether

Legal values

<u>Value</u>	<u>Effect</u>
\$LwpWhereTypeUpperLeft (1976)	Attaches the knot in the upper left side of a layout object.
\$LwpWhereTypeMiddleTop (1977)	Attaches the knot in the middle of the top area of a layout object.
\$LwpWhereTypeLayoutUpperRight (1978)	Attaches the knot in the upper right side of a layout object.
\$LwpWhereTypeMiddleLeft (1979)	Attaches the knot in the middle of the left side of a layout object.
\$LwpWhereTypeMiddleRight (1980)	Attaches the knot in the middle of the right side of a layout object.
\$LwpWhereTypeLowerLeft (1981)	Attaches the knot in the lower left side of a layout object.
\$LwpWhereTypeMiddleBottom (1982)	Attaches the knot in the middle of the bottom area of a layout object.
\$LwpWhereTypeLowerRight (1983)	Attaches the knot in the lower right side of a layout object.
\$LwpWhereTypeMiddle (1984)	Attaches the knot in the middle of a layout object.

Usage

Applies only to frame and table layout objects. Equivalent to the "Tie anchor to frame" setting in the Placement and Anchoring Options dialog box. Use this property in conjunction with the AmtToTetherFrom property, in order to properly place the layout object's knot.

Word Pro: AmtToRotateContent property

```
{button ,AL(^H_LAYOUT_CLASS;H_CELLGROUPLAYOUT_CLASS;H_CELLLAYOUT_CLASS;H_CONNECTEDLAY  
OUT_CLASS;H_COLUMNGROUPLAYOUT_CLASS;H_FOOTERLAYOUT_CLASS;H_FRAMELAYOUT_CLASS;H  
_FRAMEGROUPLAYOUT_CLASS;H_DROP_CAPLAYOUT_CLASS;H_GROUPLAYOUT_CLASS;H_HEADERLAY  
OUT_CLASS;H_NOTELAYOUT_CLASS;H_PAGELAYOUT_CLASS;H_ROWGROUPLAYOUT_CLASS;H_ROWLA  
YOUT_CLASS;H_RUBYLAYOUT_CLASS;H_SUPERTABLEGROUPLAYOUT_CLASS;H_SUPERTABLELAYOUT  
_CLASS;H_TABLELAYOUT_CLASS;H_ENDNOTELAYOUT_CLASS;H_FOOTNOTELAYOUT_CLASS;H_TABLE  
HEADINGLAYOUT_CLASS;H_TOCSUPERTABLELAYOUT_CLASS',0)} See list of classes
```

```
{button ,AL(^H_AMTTOROTATECONTENT_PROPERTY_EXSCRIPT',1)} See example
```

(Read-write) Allows you to rotate the content of a layout in tenths of degrees.

Data Type

[Integer](#)

Syntax

amttorotatecontentvalue = [objectreference].AmtToRotateContent

[objectreference].AmtToRotateContent = amttorotatecontentvalue

Legal values

The legal values for this property are 0 - 3600.

Usage

This property is equivalent to the "Rotate Image" box in the Image properties panel of the InfoBox. If the content of the layout object is not a graphic type that can be rotated, the value of this property has no effect.

To determine the appropriate value for this property, use the following formula:

$AmtToRotateContent = 3600 - (\text{degrees of rotation} * 10)$

For example, to rotate the content of a frame 45 degrees, assign this property a value of 3150.

Word Pro: AmtToTetherFrom property

```
{button ,AL('H_LAYOUT_CLASS;H_CELLGROUPLAYOUT_CLASS;H_CELLLAYOUT_CLASS;H_CONNECTEDLAY  
OUT_CLASS;H_COLUMNGROUPLAYOUT_CLASS;H_FOOTERLAYOUT_CLASS;H_FRAMELAYOUT_CLASS;H  
_FRAMEGROUPLAYOUT_CLASS;H_DROP_CAPLAYOUT_CLASS;H_GROUPLAYOUT_CLASS;H_HEADERLAY  
OUT_CLASS;H_NOTELAYOUT_CLASS;H_PAGELAYOUT_CLASS;H_ROWGROUPLAYOUT_CLASS;H_ROWLA  
YOUT_CLASS;H_RUBYLAYOUT_CLASS;H_SUPERTABLEGROUPLAYOUT_CLASS;H_SUPERTABLELAYOUT  
_CLASS;H_TABLELAYOUT_CLASS;H_ENDNOTELAYOUT_CLASS;H_FOOTNOTELAYOUT_CLASS;H_TABLE  
HEADINGLAYOUT_CLASS;H_TOCSUPERTABLELAYOUT_CLASS',0)} See list of classes
```

```
{button ,AL('H_AMTTOTETHERFROM_PROPERTY_EXSCRIPT',1)} See example
```

(Read-write) Indicates whether a layout object's knot is tied to the interior, border, or exterior of the object. The knot is the position on the frame from which it is anchored.

Data Type

Data type is [Variant](#), which allows the value of this property to be one of the constants listed below or its numeric equivalent (in parentheses).

Syntax

amttotetherfromvalue = [objectreference].AmtToTetherFrom

[objectreference].AmtToTetherFrom = amttotetherfromvalue

Legal values

<u>Value</u>	<u>Effect</u>
\$LwpWhereTypeUpperLeft (1976)	Attaches the knot to the interior of a layout object.
\$LwpWhereTypeMiddleTop (1977)	Attaches the knot to the exterior of a layout object.
\$LwpWhereTypeLayoutUpperRight (1978)	Attaches the knot to the border of a layout object.

Usage

Applies only to frame and table layout objects. Equivalent to the "Tie anchor to frame" setting in the Placement and Anchoring Options dialog box. Use this property in conjunction with the AmtTether property in order to properly place the layout object's knot.

Word Pro: Animated property

{button ,AL('H_OLEOBJECT_CLASS',0)} [See list of classes](#)

(Read-only)

Indicates whether or not a graphic object is an animated GIF.

Data Type

Integer

Syntax

animatedvalue = [objectreference].Animated

Legal values

A value of -1 indicates that the OLEObject is an animated GIF. A value of 0 indicates that the OLEObject is not an animated GIF. This property is read-only. The value of this property cannot be set by a script.

Usage

Call this property from the OLEObject for which you are seeking this information. If you aren't sure which OLE objects might be animated GIFs, you can use a Forall loop to iterate over all the OLEObjects in a division, checking the value of this property for each OLEObject.

Note In Word Pro 98, all OLE objects, including graphics, are stored as type OLEObject. In previous versions of Word Pro, graphics were stored as GraphicOleObjects, while other OLE objects were stored as type OLEObject.

Word Pro: AnyNumber property

{button ,AL('H_NUMERICFORMAT_CLASS',0)} [See list of classes](#)

{button ,AL('H_ANYNUMBER_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-only) Modifies the "Any number" condition of the current number format.

Data Type

[NumericFormatSubset](#)

Syntax

anynumbervalue = [objectreference].AnyNumber

Legal values

Always contains an instance of the NumericFormatSubset class.

Usage

Equivalent to the "Any number" condition for edit in the Edit Format dialog box. You can access this dialog box by clicking the Format Options button on the Number Format panel of the InfoBox for cell layout objects.

By accessing the "Any number" condition of a number format, you can modify how most values appear within table cells, and choose prefix or suffix text.

Word Pro: AnyOLEDELinks property

{button ,AL('H_DIVISIONINFO_CLASS',0)} [See list of classes](#)

{button ,AL('H_ANYOLEDELINKS_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-only)

Data Type

Integer

Syntax

anyoledelinksvalue = [objectreference].AnyOLEDELinks

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Word Pro: AppFoundry property

{button ,AL('H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL('H_APPFOUNDRY_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-only) A Foundry object which contains all the objects located in the Word Pro Clipboard.

Data Type

[Foundry](#)

Syntax

appfoundryvalue = [objectreference].AppFoundry

Legal values

Always contains an instance of the Foundry class.

Usage

AppFoundry is a property on the WPApplication object (always stored in the CurrentApplication variable). It contains a Foundry object which Word Pro uses as the Clipboard. This is the same Clipboard you use when you copy or cut items in a Word Pro document. When you cut or copy a selection, Word Pro takes all objects from your selection and places them in their respective collection objects in the Foundry object stored in the AppFoundry property.

For example, if you select some text and a table, and choose Edit - Copy, Word Pro places all objects that comprise that text and table into their respective collection objects in AppFoundry. This means that all the Layout objects are stored in the corresponding layout collection objects. All CharacterStyle objects are stored in the CharacterStyleCollection object. All CellEngine objects are stored in the CellCollection object. The text objects are stored in the TextCollection object. When you choose Edit - Paste, all of these objects are reassembled in their original form and displayed in the document at the insertion point.

Because Word Pro uses the Foundry object in AppFoundry as its Clipboard, you must exercise caution when working with AppFoundry. Any objects you place in AppFoundry will be included in the next Paste operation. Any objects you remove from AppFoundry will be excluded from the next Paste operation and may adversely affect the user's ability to paste from the Clipboard.

You can get an object from AppFoundry and store it in a variable, using the following statement:

```
myobject = CurrentApplication.AppFoundry.collectionpropertyname(itemreference)
```

In this statement, *myobject* is the variable in which you want to store the object; CurrentApplication is a global variable that always contains the WPApplication object; *collectionpropertyname* is the name of the property that contains the collection object where the object you want is stored; *itemreference* is the index that specifies the object you want.

For more information about collection classes, see [Overview: Word Pro LotusScript Collection Classes](#).

Note While you may retrieve objects from AppFoundry, you should not use LotusScript to place objects in the AppFoundry collections. This could interfere with normal user operations, such as Cut and Copy. When creating and storing your own Word Pro objects, use the Foundry object in the TempFoundry property.

Word Pro: ApplicationWindow property

{button ,AL('H_APPLICATION_CLASS;H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL('H_APPLICATIONWINDOW_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-only) An instance of the ApplicationWindow class. The object in this property represents the Word Pro application window that acts as the container for all of your document windows. Also known as the application workspace, this is the window that remains after you close all of your documents and leave Word Pro running.

Data Type

[ApplicationWindow](#)

Syntax

applicationwindowvalue = [objectreference].ApplicationWindow

Legal values

Always contains an instance of the ApplicationWindow class.

Usage

The ApplicationWindow object allows you to control the size, position, and appearance of the Word Pro application window. See the definition of the [ApplicationWindow](#) class for more information on how to make use of this property.

Word Pro: Application property

{button ,AL('H_BASEOBJECT_CLASS',0)} [See list of classes](#)

{button ,AL('H_APPLICATION_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-only) An instance of the WPAApplication object. The Application property is inherited from BaseObject and provides universal access to the WPAApplication object.

Data Type

[WPAApplication](#)

Syntax

applicationvalue = [objectreference].Application

Legal values

Always contains an instance of the WPAApplication class.

Usage

This property always contains a pointer to the WPAApplication object, so you can reach the WPAApplication object, regardless of where your focus is. In most cases, you simply use the leading dot feature to gain access to the WPAApplication object. However, if you are controlling a Word Pro object from another application, you cannot use the leading dot notation to get the WPAApplication object. In that circumstance, you can use the Application property on the object to access the WPAApplication object and thereby all of Word Pro.

Word Pro: ApplyAdjectivePos property

{button ,AL('H_GRAMMAR_CLASS',0)} [See list of classes](#)

(Read-write) A flag that turns the Grammar Check rule on or off for checking adjective positions.

Data Type

Integer

Syntax

applyadjectiveposvalue = [objectreference].ApplyAdjectivePos

[objectreference].ApplyAdjectivePos = applyadjectiveposvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Use this property when you are setting the corresponding rule for proofing the document in Grammar Check. Equivalent to choosing Edit - Check Grammar, clicking Options, and selecting the appropriate rule in the "Rule type" field on the Rules panel.

Note This rule is not applicable in every language. Some Apply properties can only be found in specific language versions of Word Pro.

Word Pro: ApplyAdjectNounPart property

{button ,AL('H_GRAMMAR_CLASS',0)} [See list of classes](#)

(Read-write) A flag that turns the Grammar Check rule on or off for checking adjective/noun parts.

Data Type

Integer

Syntax

applyadjectnounpartvalue = [objectreference].ApplyAdjectNounPart

[objectreference].ApplyAdjectNounPart = applyadjectnounpartvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Use this property when you are setting the corresponding rule for proofing the document in Grammar Check. Equivalent to choosing Edit - Check Grammar, clicking Options, and selecting the appropriate rule in the "Rule type" field on the Rules panel.

Note This rule is not applicable in every language. Some Apply properties can only be found in specific language versions of Word Pro.

Word Pro: ApplyAgreementWithHereThere property

{button ,AL('H_GRAMMAR_CLASS',0)} [See list of classes](#)

{button ,AL('H_APPLYAGREEMENTWITHHERETHERE_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) A flag that turns the Grammar Check rule on or off for agreement between usage of the words "here" and "there."

Data Type

[Integer](#)

Syntax

applyagreementwithheretherevalue = [objectreference].ApplyAgreementWithHereThere
[objectreference].ApplyAgreementWithHereThere = applyagreementwithheretherevalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Use this property when you are setting the corresponding rule for proofing the document in Grammar Check. Equivalent to choosing Edit - Check Grammar, clicking Options, and selecting "Agreement with 'here'/'there' " in the "Rule type" field on the Rules panel.

This rule flags errors of agreement between verbs and their predicate nouns when the sentence has "here" or "there" as its apparent subject. A predicate noun identifies or restates the subject of the sentence. When the apparent subject of the sentence is "here" or "there," the verb must agree with the predicate noun. For example, one rule will flag the sentence, "Here comes the two men who can help us," because the singular verb ("comes") does not agree with the plural predicate noun ("men").

Note This rule is not applicable in every language. Some Apply properties can only be found in specific language versions of Word Pro.

Word Pro: ApplyAnglicisms property

{button ,AL('H_GRAMMAR_CLASS',0)} [See list of classes](#)

(Read-write) A flag that turns the specific Grammar Check rule on or off.

Data Type

Integer

Syntax

applyanglicismsvalue = [objectreference].ApplyAnglicisms

[objectreference].ApplyAnglicisms = applyanglicismsvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Use this property when you are setting the corresponding rule for proofing the document in Grammar Check. Equivalent to choosing Edit - Check Grammar, clicking Options, and selecting the appropriate rule in the "Rule type" field on the Rules panel.

Note This rule is not applicable in every language. This property can only be found in specific language versions of Word Pro.

Word Pro: ApplyArchaicExpressions property

{button ,AL('H_GRAMMAR_CLASS',0)} [See list of classes](#)

{button ,AL('H_APPLYARCHAICEXPRESSIONS_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) A flag that turns the Grammar Check rule on or off for checking archaic expressions.

Data Type

[Integer](#)

Syntax

applyarchaicexpressionsvalue = [objectreference].ApplyArchaicExpressions

[objectreference].ApplyArchaicExpressions = applyarchaicexpressionsvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Use this property when you are setting the corresponding rule for proofing the document in Grammar Check. Equivalent to choosing Edit - Check Grammar, clicking Options, and selecting "Archaic expressions" in the "Rule type" field on the Rules panel.

This rule flags words and expressions that are no longer current in standard usage. These words or expressions may be appropriate in certain contexts, but might seem stilted or awkward in everyday writing. They should be replaced with contemporary equivalents, whenever possible. For example, the sentence, "Would you perchance be free for lunch on Tuesday?" can be revised using the more contemporary term, "possibly," in place of "perchance."

Note This rule is not applicable in every language. Some Apply properties can only be found in specific language versions of Word Pro.

Word Pro: ApplyArticleAgreement property

{button ,AL('H_GRAMMAR_CLASS',0)} [See list of classes](#)

{button ,AL('H_APPLYARTICLEAGREEMENT_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) A flag that turns the Grammar Check rule on or off for checking "A" and "An" article agreements.

Data Type

[Integer](#)

Syntax

applyarticleagreementvalue = [objectreference].ApplyArticleAgreement [objectreference].ApplyArticleAgreement = applyarticleagreementvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Use this property when you are setting the corresponding rule for proofing the document in Grammar Check. Equivalent to choosing Edit - Check Grammar, clicking Options, and selecting "A vs. An " in the "Rule type" field on the Rules panel.

This rule flags incorrect indefinite articles (a/an) in noun phrases. Words that begin with a vowel usually take "an" as the indefinite article (an army), and words that begin with a consonant usually take "a" as the indefinite article (a carrot).

Note This rule is not applicable in every language. Some Apply properties can only be found in specific language versions of Word Pro.

Word Pro: ApplyBadComparatives property

{button ,AL('H_GRAMMAR_CLASS',0)} [See list of classes](#)

(Read-write) A flag that turns the specific Grammar Check rule on or off.

Data Type

Integer

Syntax

applybadcomparativesvalue =[objectreference].ApplyBadComparatives [objectreference].ApplyBadComparatives = applybadcomparativesvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Use this property when you are setting the corresponding rule for proofing the document in Grammar Check. Equivalent to choosing Edit - Check Grammar, clicking Options, and selecting the appropriate rule in the "Rule type" field on the Rules panel.

Note This rule is not applicable in every language. Some Apply properties can only be found in specific language versions of Word Pro.

Word Pro: ApplyBadInflection property

{button ,AL(^H_GRAMMAR_CLASS',0)} [See list of classes](#)

(Read-write) A flag that turns the specific Grammar Check rule on or off.

Data Type

Integer

Syntax

applybadinflectionvalue = [objectreference].ApplyBadInflection

[objectreference].ApplyBadInflection = applybadinflectionvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Use this property when you are setting the corresponding rule for proofing the document in Grammar Check. Equivalent to choosing Edit - Check Grammar, clicking Options, and selecting the appropriate rule in the "Rule type" field on the Rules panel.

Note This rule is not applicable in every language. Some Apply properties can only be found in specific language versions of Word Pro.

Word Pro: ApplyBadNounGender property

{button ,AL('H_GRAMMAR_CLASS',0)} [See list of classes](#)

(Read-write) A flag that turns the specific Grammar Check rule on or off.

Data Type

Integer

Syntax

applybadnougendervalue = [objectreference].ApplyBadNounGender

[objectreference].ApplyBadNounGender = applybadnougendervalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Use this property when you are setting the corresponding rule for proofing the document in Grammar Check. Equivalent to choosing Edit - Check Grammar, clicking Options, and selecting the appropriate rule in the "Rule type" field on the Rules panel.

Note This rule is not applicable in every language. Some Apply properties can only be found in specific language versions of Word Pro.

Word Pro: ApplyBadNoun property

{button ,AL('H_GRAMMAR_CLASS',0)} [See list of classes](#)

(Read-write) A flag that turns the specific Grammar Check rule on or off.

Data Type

Integer

Syntax

applyBadNounvalue = [objectreference].ApplyBadNoun

[objectreference].ApplyBadNoun = applyBadNounvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Use this property when you are setting the corresponding rule for proofing the document in Grammar Check. Equivalent to choosing Edit - Check Grammar, clicking Options, and selecting the appropriate rule in the "Rule type" field on the Rules panel.

Note This rule is not applicable in every language. Some Apply properties can only be found in specific language versions of Word Pro.

Word Pro: ApplyBadPlural property

{button ,AL('H_GRAMMAR_CLASS',0)} [See list of classes](#)

(Read-write) A flag that turns the specific Grammar Check rule on or off.

Data Type

Integer

Syntax

applybadpluralvalue = [objectreference]. ApplyBadPlural

[objectreference]. ApplyBadPlural= applybadpluralvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Use this property when you are setting the corresponding rule for proofing the document in Grammar Check. Equivalent to choosing Edit - Check Grammar, clicking Options, and selecting the appropriate rule in the "Rule type" field on the Rules panel.

Note This rule is not applicable in every language. Some Apply properties can only be found in specific language versions of Word Pro.

Word Pro: ApplyBadPrepositions property

{button ,AL('H_GRAMMAR_CLASS',0)} [See list of classes](#)

{button ,AL('H_APPLYBADPREPOSITIONS_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) A flag that turns the specific Grammar Check rule on or off.

Data Type

[Integer](#)

Syntax

applybadprepositionsvalue = [objectreference].ApplyBadPrepositions

[objectreference].ApplyBadPrepositions = applybadprepositionsvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Use this property when you are setting the corresponding rule for proofing the document in Grammar Check. Equivalent to choosing Edit - Check Grammar, clicking Options, and selecting the appropriate rule in the "Rule type" field on the Rules panel.

Note This rule is not applicable in every language. Some Apply properties can only be found in specific language versions of Word Pro.

Word Pro: ApplyBelgianExpression property

{button ,AL('H_GRAMMAR_CLASS',0)} [See list of classes](#)

(Read-write) A flag that turns the specific Grammar Check rule on or off.

Data Type

Integer

Syntax

applybelgianexpressionvalue = [objectreference].ApplyBelgianExpression

[objectreference].ApplyBelgianExpression = applybelgianexpressionvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Use this property when you are setting the corresponding rule for proofing the document in Grammar Check. Equivalent to choosing Edit - Check Grammar, clicking Options, and selecting the appropriate rule in the "Rule type" field on the Rules panel.

Note This rule is not applicable in every language. Some Apply properties can only be found in specific language versions of Word Pro.

Word Pro: ApplyBorrowedForeign property

{button ,AL('H_GRAMMAR_CLASS',0)} [See list of classes](#)

(Read-write) A flag that turns the specific Grammar Check rule on or off.

Data Type

Integer

Syntax

applyborrowedforeignvalue = [objectreference].ApplyBorrowedForeign

[objectreference].ApplyBorrowedForeign = applyborrowedforeignvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Use this property when you are setting the corresponding rule for proofing the document in Grammar Check. Equivalent to choosing Edit - Check Grammar, clicking Options, and selecting the appropriate rule in the "Rule type" field on the Rules panel.

Note This rule is not applicable in every language. Some Apply properties can only be found in specific language versions of Word Pro.

Word Pro: ApplyBureuaJargon property

{button ,AL('H_GRAMMAR_CLASS',0)} [See list of classes](#)

(Read-write) A flag that turns the specific Grammar Check rule on or off.

Data Type

Integer

Syntax

applybureujargonvalue = [objectreference].ApplyBureuaJargon

[objectreference].ApplyBureuaJargon = applybureujargonvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Use this property when you are setting the corresponding rule for proofing the document in Grammar Check. Equivalent to choosing Edit - Check Grammar, clicking Options, and selecting the appropriate rule in the "Rule type" field on the Rules panel.

Note This rule is not applicable in every language. Some Apply properties can only be found in specific language versions of Word Pro.

Word Pro: ApplyCalque property

{button ,AL('H_GRAMMAR_CLASS',0)} [See list of classes](#)

(Read-write) A flag that turns the specific Grammar Check rule on or off.

Data Type

Integer

Syntax

applycalquevalue = [objectreference].ApplyCalque

[objectreference].ApplyCalque = applycalquevalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Use this property when you are setting the corresponding rule for proofing the document in Grammar Check. Equivalent to choosing Edit - Check Grammar, clicking Options, and selecting the appropriate rule in the "Rule type" field on the Rules panel.

Note This rule is not applicable in every language. This property can only be found in specific language versions of Word Pro, for example, Spanish.

Word Pro: ApplyCapitalizationCheck property

{button ,AL('H_GRAMMAR_CLASS',0)} [See list of classes](#)

{button ,AL('H_APPLYCAPITALIZATIONCHECK_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) A flag that turns the Grammar Check rule on or off for checking capitalization.

Data Type

[Integer](#)

Syntax

applycapitalizationcheckvalue = [objectreference].ApplyCapitalizationCheck

[objectreference].ApplyCapitalizationCheck = applycapitalizationcheckvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Use this property when you are setting the corresponding rule for proofing the document in Grammar Check. Equivalent to choosing Edit - Check Grammar, clicking Options, and selecting "Capitalization errors" in the "Rule type" field on the Rules panel.

This rule flags the most common capitalization errors, including uncapitalized proper names, uncapitalized salutations, and incorrectly capitalized or uncapitalized names of days, months, seasons, holidays and abbreviations. It also flags sentences that begin with a lowercase letter. For example, this rule flags "chicago" as "Chicago," "best wishes" as "Best Wishes," and "Memorial day" as "Memorial Day."

Note This rule is not applicable in every language. Some Apply properties can only be found in specific language versions of Word Pro.

Word Pro: ApplyClauseErrors property

{button ,AL('H_GRAMMAR_CLASS',0)} [See list of classes](#)

{button ,AL('H_APPLYCLAUSEERRORS_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) A flag that turns the Grammar Check rule on or off for checking clause errors.

Data Type

[Integer](#)

Syntax

applyclauseerrorsvalue = [objectreference].ApplyClauseErrors

[objectreference].ApplyClauseErrors = applyclauseerrorsvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Use this property when you are setting the corresponding rule for proofing the document in Grammar Check.

Equivalent to choosing Edit - Check Grammar, clicking Options, and selecting "Clause errors" in the "Rule type" field on the Rules panel.

This rule flags general errors of sentence structure, such as run-on sentences and sentence fragments. It checks to see that conjunctions are used correctly and that correct punctuation appears between clauses. For example, this rule will flag the sentence, "We chopped up fruit, and we diced the potatoes, and we made a pie crust," since only one "and" is necessary when three clauses appear in sequence.

Note This rule is not applicable in every language. Some Apply properties can only be found in specific language versions of Word Pro.

Word Pro: ApplyCliches property

{button ,AL('H_GRAMMAR_CLASS',0)} [See list of classes](#)

{button ,AL('H_APPLYCLICHES_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) A flag that turns the Grammar Check rule on or off for checking expressions that are cliches.

Data Type

[Integer](#)

Syntax

applyclichesvalue = [objectreference].ApplyCliches

[objectreference].ApplyCliches = applyclichesvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Use this property when you are setting the corresponding rule for proofing the document in Grammar Check. Equivalent to choosing Edit - Check Grammar, clicking Options, and selecting "Cliches" in the "Rule type" field on the Rules panel.

This rule flags clichés, colorful expressions used so often that they have lost their original force. Although clichés may occasionally be appropriate, you should avoid using them casually or excessively. Sometimes a cliché can be replaced by a more direct term. In other cases, the sentence must be rephrased to avoid the cliché. For example, the phrase, "make a mountain out of a molehill," might become "exaggerate" or "overreact."

Note This rule is not applicable in every language. Some Apply properties can only be found in specific language versions of Word Pro.

Word Pro: ApplyColloquialExpression property

{button ,AL('H_GRAMMAR_CLASS',0)} [See list of classes](#)

(Read-write) A flag that turns the Grammar Check rule on or off for checking colloquial expressions.

Data Type

Integer

Syntax

applycolloquialexpressionvalue = [objectreference].ApplyColloquialExpression
[objectreference].ApplyColloquialExpression = applycolloquialexpressionvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Use this property when you are setting the corresponding rule for proofing the document in Grammar Check. Equivalent to choosing Edit - Check Grammar, clicking Options, and selecting the appropriate rule in the "Rule type" field on the Rules panel.

Note This rule is not applicable in every language. Some Apply properties can only be found in specific language versions of Word Pro.

Word Pro: ApplyCommonlyConfusedWords property

{button ,AL('H_GRAMMAR_CLASS',0)} [See list of classes](#)

{button ,AL('H_APPLYCOMMONLYCONFUSEDWORDS_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) A flag that turns the Grammar Check rule on or off for checking words that are commonly confused.

Data Type

[Integer](#)

Syntax

applycommonlyconfusedwordsvalue = [objectreference].ApplyCommonlyConfusedWords

[objectreference].ApplyCommonlyConfusedWords = applycommonlyconfusedwordsvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Use this property when you are setting the corresponding rule for proofing the document in Grammar Check.

Equivalent to choosing Edit - Check Grammar, clicking Options, and selecting "Commonly confused words" in the "Rule type" field on the Rules panel.

This rule flags commonly confused words that have similar, though not identical, pronunciations. The confused pairs include words that involve confusion between a noun and a verb. For example, the rule will flag the sentence, "You would be wise to seek legal advise before signing a contract," because the verb, "advise," is mistakenly used instead of the noun, "advice."

Note This rule is not applicable in every language. Some Apply properties can only be found in specific language versions of Word Pro.

Word Pro: ApplyCommonMisspell property

{button ,AL('H_GRAMMAR_CLASS',0)} [See list of classes](#)

(Read-write) A flag that turns the Grammar Check rule on or off for checking common misspellings.

Data Type

Integer

Syntax

applycommonmisspellvalue = [objectreference].ApplyCommonMisspell [objectreference].ApplyCommonMisspell = applycommonmisspellvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Use this property when you are setting the corresponding rule for proofing the document in Grammar Check. Equivalent to choosing Edit - Check Grammar, clicking Options, and selecting the appropriate rule in the "Rule type" field on the Rules panel.

Note This rule is not applicable in every language. Some Apply properties can only be found in specific language versions of Word Pro.

Word Pro: ApplyComplexWords property

{button ,AL('H_GRAMMAR_CLASS',0)} [See list of classes](#)

(Read-write) A flag that turns the specific Grammar Check rule on or off.

Data Type

Integer

Syntax

applycomplexwordsvvalue = [objectreference].ApplyComplexWords

[objectreference].ApplyComplexWords = applycomplexwordsvvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Use this property when you are setting the corresponding rule for proofing the document in Grammar Check. Equivalent to choosing Edit - Check Grammar, clicking Options, and selecting the appropriate rule in the "Rule type" field on the Rules panel.

Note This rule is not applicable in every language. Some Apply properties can only be found in specific language versions of Word Pro.

Word Pro: ApplyConfusedEasy property

{button ,AL(^H_GRAMMAR_CLASS',0)} [See list of classes](#)

(Read-write) A flag that turns the specific Grammar Check rule on or off.

Data Type

Integer

Syntax

applyconfusedeasyvalue = [objectreference].ApplyConfusedEasy

[objectreference].ApplyConfusedEasy = applyconfusedeasyvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Use this property when you are setting the corresponding rule for proofing the document in Grammar Check. Equivalent to choosing Edit - Check Grammar, clicking Options, and selecting the appropriate rule in the "Rule type" field on the Rules panel.

Note This rule is not applicable in every language. This property can only be found in specific language versions of Word Pro, for example, French.

Word Pro: ApplyConfusedEnglish property

{button ,AL('H_GRAMMAR_CLASS',0)} [See list of classes](#)

(Read-write) A flag that turns the specific Grammar Check rule on or off.

Data Type

Integer

Syntax

applyconfusedenglishvalue = [objectreference].ApplyConfusedEnglish [objectreference].ApplyConfusedEnglish = applyconfusedenglishvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Use this property when you are setting the corresponding rule for proofing the document in Grammar Check. Equivalent to choosing Edit - Check Grammar, clicking Options, and selecting the appropriate rule in the "Rule type" field on the Rules panel.

Note This rule is not applicable in every language. This property can only be found in specific language versions of Word Pro, for example, French.

Word Pro: ApplyConfusedHard property

{button ,AL(^H_GRAMMAR_CLASS',0)} [See list of classes](#)

(Read-write) A flag that turns the specific Grammar Check rule on or off.

Data Type

Integer

Syntax

applyconfusedhardvalue = [objectreference].ApplyConfusedHard

[objectreference].ApplyConfusedHard = applyconfusedhardvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Use this property when you are setting the corresponding rule for proofing the document in Grammar Check. Equivalent to choosing Edit - Check Grammar, clicking Options, and selecting the appropriate rule in the "Rule type" field on the Rules panel.

Note This rule is not applicable in every language. This property can only be found in specific language versions of Word Pro, for example, French.

Word Pro: ApplyConfusedMedium property

{button ,AL(^H_GRAMMAR_CLASS',0)} [See list of classes](#)

(Read-write) A flag that turns the specific Grammar Check rule on or off.

Data Type

Integer

Syntax

applyconfusedmediumvalue = [objectreference].ApplyConfusedMedium [objectreference].ApplyConfusedMedium = applyconfusedmediumvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Use this property when you are setting the corresponding rule for proofing the document in Grammar Check. Equivalent to choosing Edit - Check Grammar, clicking Options, and selecting the appropriate rule in the "Rule type" field on the Rules panel.

Note This rule is not applicable in every language. This property can only be found in specific language versions of Word Pro, for example, French.

Word Pro: ApplyConfusedVerb property

{button ,AL(^H_GRAMMAR_CLASS',0)} [See list of classes](#)

(Read-write) A flag that turns the specific Grammar Check rule on or off.

Data Type

Integer

Syntax

applyconfusedverbvalue = [objectreference].ApplyConfusedVerb

[objectreference].ApplyConfusedVerb = applyconfusedverbvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Use this property when you are setting the corresponding rule for proofing the document in Grammar Check. Equivalent to choosing Edit - Check Grammar, clicking Options, and selecting the appropriate rule in the "Rule type" field on the Rules panel.

Note This rule is not applicable in every language. This property can only be found in specific language versions of Word Pro, for example, Dutch.

Word Pro: ApplyConsecutiveNouns property

{button ,AL('H_GRAMMAR_CLASS',0)} [See list of classes](#)

{button ,AL('H_APPLYCONSECUTIVENOUNS_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Sets the Grammar Check behavior for consecutive nouns.

Data Type

[Integer](#)

Syntax

applyconsecutivenounsvalue = [objectreference].ApplyConsecutiveNouns

[objectreference].ApplyConsecutiveNouns = applyconsecutivenounsvalue

Legal values

<u>Value</u>	<u>Effect</u>
0	Never flag consecutive nouns.
1	Flag three or more consecutive nouns in a row.
2	Flag four or more consecutive nouns in a row.
3	Flag five or more consecutive nouns in a row.

Usage

Use this property when you are setting the corresponding rule for proofing the document in Grammar Check. Equivalent to setting the "Flag consecutive nouns" option in the Grammar Options dialog box.

Note This rule is not applicable in every language. Some Apply properties can only be found in specific language versions of Word Pro.

Word Pro: ApplyContractions property

{button ,AL('H_GRAMMAR_CLASS',0)} [See list of classes](#)

{button ,AL('H_APPLYCONTRACTIONS_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) A flag that turns the Grammar Check rule on or off for checking contractions.

Data Type

[Integer](#)

Syntax

applycontractionsvalue = [objectreference].ApplyContractions

[objectreference].ApplyContractions = applycontractionsvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Use this property when you are setting the corresponding rule for proofing the document in Grammar Check. Equivalent to choosing Edit - Check Grammar, clicking Options, and selecting "Contractions" in the "Rule type" field on the Rules panel.

This rule flags contractions and recommends the appropriate expanded forms. For example, one rule flags the sentence, "I've completed the course," and suggests replacing "I've" with "I have." Contractions are acceptable in many written contexts, especially if you are striving for a conversational tone. They may be inappropriate, however, in some formal documents. This rule helps you identify and revise contractions when you are working in formal documents.

Note This rule is not applicable in every language. Some Apply properties can only be found in specific language versions of Word Pro.

Word Pro: ApplyDerogatory property

{button ,AL('H_GRAMMAR_CLASS',0)} [See list of classes](#)

(Read-write) A flag that turns the specific Grammar Check rule on or off.

Data Type

Integer

Syntax

applyderogatoryvalue = [objectreference].ApplyDerogatory

[objectreference].ApplyDerogatory = applyderogatoryvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Use this property when you are setting the corresponding rule for proofing the document in Grammar Check. Equivalent to choosing Edit - Check Grammar, clicking Options, and selecting the appropriate rule in the "Rule type" field on the Rules panel.

Note This rule is not applicable in every language. Some Apply properties can only be found in specific language versions of Word Pro.

Word Pro: ApplyDifferentPrep property

{button ,AL('H_GRAMMAR_CLASS',0)} [See list of classes](#)

(Read-write) A flag that turns the Grammar Check rule on or off for checking incorrect prepositions.

Data Type

Integer

Syntax

applydifferentprepvalue = [objectreference].ApplyDifferentPrep

[objectreference].ApplyDifferentPrep = applydifferentprepvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Use this property when you are setting the corresponding rule for proofing the document in Grammar Check. Equivalent to choosing Edit - Check Grammar, clicking Options, and selecting "Inappropriate prepositions" in the "Rule type" field on the Rules panel.

This rule flags expressions that include an incorrect preposition and offers the appropriate preposition as a correction. For example, the rule will flag "adhere by" as "adhere to," "center around" as "center on," and so on.

Note This rule is not applicable in every language. Some Apply properties can only be found in specific language versions of Word Pro.

Word Pro: ApplyDoubleNegative property

{button ,AL('H_GRAMMAR_CLASS',0)} [See list of classes](#)

{button ,AL('H_APPLYDOUBLENEGATIVE_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) A flag that turns the Grammar Check rule on or off for checking double negatives.

Data Type

[Integer](#)

Syntax

applydoublenegativevalue = [objectreference].ApplyDoubleNegative [objectreference].ApplyDoubleNegative = applydoublenegativevalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Use this property when you are setting the corresponding rule for proofing the document in Grammar Check. Equivalent to choosing Edit - Check Grammar, clicking Options, and selecting "Double negatives" in the "Rule type" field on the Rules panel.

This rule flags confusing or awkward phrases that may contain more than one negative word. For example, a sentence may be flagged that uses "can't hardly" instead of "cannot" or "can hardly," and "in no uncertain terms" instead of "clearly" or "specifically."

Note This rule is not applicable in every language. Some Apply properties can only be found in specific language versions of Word Pro.

Word Pro: ApplyDoublePlural property

{button ,AL('H_GRAMMAR_CLASS',0)} [See list of classes](#)

(Read-write) A flag that turns the specific Grammar Check rule on or off.

Data Type

Integer

Syntax

applydoublepluralvalue = [objectreference].ApplyDoublePlural

[objectreference].ApplyDoublePlural = applydoublepluralvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Use this property when you are setting the corresponding rule for proofing the document in Grammar Check. Equivalent to choosing Edit - Check Grammar, clicking Options, and selecting the appropriate rule in the "Rule type" field on the Rules panel.

Note This rule is not applicable in every language. Some Apply properties can only be found in specific language versions of Word Pro.

Word Pro: ApplyDoubleWordCheck property

{button ,AL('H_GRAMMAR_CLASS',0)} [See list of classes](#)

{button ,AL('H_APPLYDOUBLEWORDCHECK_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) A flag that turns the Grammar Check rule on or off for checking double words.

Data Type

[Integer](#)

Syntax

applydoublewordcheckvalue = [objectreference].ApplyDoubleWordCheck

[objectreference].ApplyDoubleWordCheck = applydoublewordcheckvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Use this property when you are setting the corresponding rule for proofing the document in Grammar Check.

Equivalent to choosing Edit - Check Grammar, clicking Options, and selecting "Doubled words" in the "Rule type" field on the Rules panel.

This rule flags sequences of two identical words. Language-specific exceptions are made for legitimately doubled words (for example, "had had" as in "We had had the same discussion before," and "that that" as in "She thought that that problem had been solved."). This rule also flags a succession of articles ("the" and "a"), possessive pronouns ("my" and "his"), and similar words that must not be followed by a word of the same type.

Note This rule is not applicable in every language. Some Apply properties can only be found in specific language versions of Word Pro.

Word Pro: ApplyElision property

{button ,AL(^H_GRAMMAR_CLASS',0)} [See list of classes](#)

(Read-write) A flag that turns the Grammar Check rule on or off for checking elisions (omitting something in a word, such as a final or initial pronunciation).

Data Type

Integer

Syntax

applyelisionvalue = [objectreference].ApplyElision

[objectreference].ApplyElision = applyelisionvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Use this property when you are setting the corresponding rule for proofing the document in Grammar Check. Equivalent to choosing Edit - Check Grammar, clicking Options, and selecting the appropriate rule in the "Rule type" field on the Rules panel.

Note This rule is not applicable in every language. Some Apply properties can only be found in specific language versions of Word Pro.

Word Pro: ApplyEnglishDerived property

{button ,AL('H_GRAMMAR_CLASS',0)} [See list of classes](#)

(Read-write) A flag that turns the specific Grammar Check rule on or off.

Data Type

Integer

Syntax

applyenglishderivedvalue = [objectreference].ApplyEnglishDerived

[objectreference].ApplyEnglishDerived = applyenglishderivedvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Use this property when you are setting the corresponding rule for proofing the document in Grammar Check. Equivalent to choosing Edit - Check Grammar, clicking Options, and selecting the appropriate rule in the "Rule type" field on the Rules panel.

Note This rule is not applicable in every language. Some Apply properties can only be found in specific language versions of Word Pro.

Word Pro: ApplyEnglishWords property

{button ,AL('H_GRAMMAR_CLASS',0)} [See list of classes](#)

(Read-write) A flag that turns the specific Grammar Check rule on or off.

Data Type

Integer

Syntax

applyenglishwordsvvalue = [objectreference].ApplyEnglishWords

[objectreference].ApplyEnglishWords = applyenglishwordsvvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Use this property when you are setting the corresponding rule for proofing the document in Grammar Check. Equivalent to choosing Edit - Check Grammar, clicking Options, and selecting the appropriate rule in the "Rule type" field on the Rules panel.

Note This rule is not applicable in every language. Some Apply properties can only be found in specific language versions of Word Pro.

Word Pro: ApplyExotic property

{button ,AL(^H_GRAMMAR_CLASS',0)} [See list of classes](#)

(Read-write) A flag that turns the specific Grammar Check rule on or off.

Data Type

Integer

Syntax

applyexoticvalue = [objectreference].ApplyExotic

[objectreference].ApplyExotic = applyexoticvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Use this property when you are setting the corresponding rule for proofing the document in Grammar Check. Equivalent to choosing Edit - Check Grammar, clicking Options, and selecting the appropriate rule in the "Rule type" field on the Rules panel.

Note This rule is not applicable in every language. Some Apply properties can only be found in specific language versions of Word Pro.

Word Pro: ApplyExtraPrepositionCheck property

{button ,AL('H_GRAMMAR_CLASS',0)} [See list of classes](#)

{button ,AL('H_APPLYEXTRAPREPOSITIONCHECK_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) A flag that turns the specific Grammar Check rule on or off.

Data Type

[Integer](#)

Syntax

applyextraprepositioncheckvalue = [objectreference].ApplyExtraPrepositionCheck

[objectreference].ApplyExtraPrepositionCheck = applyextraprepositioncheckvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Use this property when you are setting the corresponding rule for proofing the document in Grammar Check. Equivalent to choosing Edit - Check Grammar, clicking Options, and selecting the appropriate rule in the "Rule type" field on the Rules panel.

Note This rule is not applicable in every language. Some Apply properties can only be found in specific language versions of Word Pro.

Word Pro: ApplyFalseFriend property

{button ,AL('H_GRAMMAR_CLASS',0)} [See list of classes](#)

(Read-write) A flag that turns the specific Grammar Check rule on or off.

Data Type

Integer

Syntax

applyfalsefriendvalue = [objectreference].ApplyFalseFriend

[objectreference].ApplyFalseFriend = applyfalsefriendvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Use this property when you are setting the corresponding rule for proofing the document in Grammar Check. Equivalent to choosing Edit - Check Grammar, clicking Options, and selecting the appropriate rule in the "Rule type" field on the Rules panel.

Note This rule is not applicable in every language. This property can only be found in specific language versions of Word Pro, for example, Spanish.

Word Pro: ApplyFemaleOccupation property

{button ,AL(^H_GRAMMAR_CLASS',0)} [See list of classes](#)

(Read-write) A flag that turns the specific Grammar Check rule on or off.

Data Type

Integer

Syntax

applyfemaleoccupationvalue = [objectreference].ApplyFemaleOccupation [objectreference].ApplyFemaleOccupation = applyfemaleoccupationvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Use this property when you are setting the corresponding rule for proofing the document in Grammar Check. Equivalent to choosing Edit - Check Grammar, clicking Options, and selecting the appropriate rule in the "Rule type" field on the Rules panel.

Note This rule is not applicable in every language. Some Apply properties can only be found in specific language versions of Word Pro.

Word Pro: ApplyFixedExpression property

{button ,AL('H_GRAMMAR_CLASS',0)} [See list of classes](#)

(Read-write) A flag that turns the specific Grammar Check rule on or off.

Data Type

Integer

Syntax

applyfixedexpressionvalue = [objectreference].ApplyFixedExpression

[objectreference].ApplyFixedExpression = applyfixedexpressionvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Use this property when you are setting the corresponding rule for proofing the document in Grammar Check. Equivalent to choosing Edit - Check Grammar, clicking Options, and selecting the appropriate rule in the "Rule type" field on the Rules panel.

Note This rule is not applicable in every language. Some Apply properties can only be found in specific language versions of Word Pro.

Word Pro: ApplyForeignWord property

{button ,AL(^H_GRAMMAR_CLASS',0)} [See list of classes](#)

(Read-write) A flag that turns the specific Grammar Check rule on or off.

Data Type

Integer

Syntax

applyforeignwordvalue = [objectreference].ApplyForeignWord

[objectreference].ApplyForeignWord = applyforeignwordvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Use this property when you are setting the corresponding rule for proofing the document in Grammar Check. Equivalent to choosing Edit - Check Grammar, clicking Options, and selecting the appropriate rule in the "Rule type" field on the Rules panel.

Note This rule is not applicable in every language. Some Apply properties can only be found in specific language versions of Word Pro.

Word Pro: ApplyFormalTerms property

{button ,AL(^H_GRAMMAR_CLASS',0)} [See list of classes](#)

(Read-write) A flag that turns the specific Grammar Check rule on or off.

Data Type

Integer

Syntax

applyformaltermsvalue = [objectreference].ApplyFormalTerms

[objectreference].ApplyFormalTerms = applyformaltermsvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Use this property when you are setting the corresponding rule for proofing the document in Grammar Check. Equivalent to choosing Edit - Check Grammar, clicking Options, and selecting the appropriate rule in the "Rule type" field on the Rules panel.

Note This rule is not applicable in every language. Some Apply properties can only be found in specific language versions of Word Pro.

Word Pro: ApplyFormatErrors property

{button ,AL('H_GRAMMAR_CLASS',0)} [See list of classes](#)

{button ,AL('H_APPLYFORMATERRORS_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) A flag that turns the Grammar Check rule on or off for checking formatting errors.

Data Type

[Integer](#)

Syntax

applyformaterrorsvalue = [objectreference].ApplyFormatErrors

[objectreference].ApplyFormatErrors = applyformaterrorsvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Use this property when you are setting the corresponding rule for proofing the document in Grammar Check.

Equivalent to choosing Edit - Check Grammar, clicking Options, and selecting "Formatting errors" in the "Rule type" field on the Rules panel.

This rule checks the format of numbers (placement of periods/commas, endings of ordinal numbers, spelling of fractions/other numbers), dates (use of cardinal/ordinal numbers), times (use of abbreviations and punctuation marks), currency/other symbols, and addresses.

Note This rule is not applicable in every language. Some Apply properties can only be found in specific language versions of Word Pro.

Word Pro: ApplyGallicisms property

{button ,AL('H_GRAMMAR_CLASS',0)} [See list of classes](#)

(Read-write) A flag that turns the specific Grammar Check rule on or off.

Data Type

Integer

Syntax

applygallicismsvalue = [objectreference].ApplyGallicisms

[objectreference].ApplyGallicisms = applygallicismsvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Use this property when you are setting the corresponding rule for proofing the document in Grammar Check. Equivalent to choosing Edit - Check Grammar, clicking Options, and selecting the appropriate rule in the "Rule type" field on the Rules panel.

Note This rule is not applicable in every language. Some Apply properties can only be found in specific language versions of Word Pro.

Word Pro: ApplyGenderExpressions property

{button ,AL('H_GRAMMAR_CLASS',0)} [See list of classes](#)

{button ,AL('H_APPLYGENDEREXPRESSIONS_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) A flag that turns the Grammar Check rule on or off for checking gender expressions.

Data Type

[Integer](#)

Syntax

applygenderexpressionsvalue = [objectreference].ApplyGenderExpressions

[objectreference].ApplyGenderExpressions = applygenderexpressionsvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Use this property when you are setting the corresponding rule for proofing the document in Grammar Check.

Equivalent to choosing Edit - Check Grammar, clicking Options, and selecting "Gender-specific expressions" in the "Rule type" field on the Rules panel.

This rule flags gender-specific terms, such as names of occupations or professions that may unnecessarily indicate a person's gender. For professions that were formerly dominated by women but now include men, gender-neutral designations are preferred. For example, one rule will flag the sentence, "The guest speaker was a popular local poetess," and suggest the word, "poet" as a substitute.

Note This rule is not applicable in every language. Some Apply properties can only be found in specific language versions of Word Pro.

Word Pro: ApplyGermanisms property

{button ,AL('H_GRAMMAR_CLASS',0)} [See list of classes](#)

(Read-write) A flag that turns the specific Grammar Check rule on or off.

Data Type

Integer

Syntax

applygermanismsvalue = [objectreference].ApplyGermanisms

[objectreference].ApplyGermanisms = applygermanismsvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Use this property when you are setting the corresponding rule for proofing the document in Grammar Check. Equivalent to choosing Edit - Check Grammar, clicking Options, and selecting the appropriate rule in the "Rule type" field on the Rules panel.

Note This rule is not applicable in every language. Some Apply properties can only be found in specific language versions of Word Pro.

Word Pro: ApplyHomoGraphs property

{button ,AL(^H_GRAMMAR_CLASS',0)} [See list of classes](#)

(Read-write) A flag that turns the Grammar Check rule on or off for checking homographic expressions (one of two or more words that have the same spelling but differ in origin, meaning, and sometimes pronunciation).

Data Type

[Integer](#)

Syntax

applyhomographsvalue = [objectreference].ApplyHomoGraphs

[objectreference].ApplyHomoGraphs = applyhomographsvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Use this property when you are setting the corresponding rule for proofing the document in Grammar Check. Equivalent to choosing Edit - Check Grammar, clicking Options, and selecting the appropriate rule in the "Rule type" field on the Rules panel.

Note This rule is not applicable in every language. This property can only be found in specific language versions of Word Pro, for example, Italian.

Word Pro: ApplyHomonymsEasy property

{button ,AL('H_GRAMMAR_CLASS',0)} [See list of classes](#)

(Read-write) A flag that turns the specific Grammar Check rule on or off.

Data Type

Integer

Syntax

applyhomonymseasyvalue = [objectreference].ApplyHomonymsEasy [objectreference].ApplyHomonymsEasy = applyhomonymseasyvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Use this property when you are setting the corresponding rule for proofing the document in Grammar Check. Equivalent to choosing Edit - Check Grammar, clicking Options, and selecting the appropriate rule in the "Rule type" field on the Rules panel.

Note This rule is not applicable in every language. Some Apply properties can only be found in specific language versions of Word Pro.

Word Pro: ApplyHomonymsHard property

{button ,AL('H_GRAMMAR_CLASS',0)} [See list of classes](#)

(Read-write) A flag that turns the specific Grammar Check rule on or off.

Data Type

Integer

Syntax

applyhomonymshardvalue = [objectreference].ApplyHomonymsHard

[objectreference].ApplyHomonymsHard = applyhomonymshardvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Use this property when you are setting the corresponding rule for proofing the document in Grammar Check. Equivalent to choosing Edit - Check Grammar, clicking Options, and selecting the appropriate rule in the "Rule type" field on the Rules panel.

Note This rule is not applicable in every language. Some Apply properties can only be found in specific language versions of Word Pro.

Word Pro: ApplyHomonyms property

{button ,AL('H_GRAMMAR_CLASS',0)} [See list of classes](#)

(Read-write) A flag that turns the Grammar Check rule on or off for checking homonyms.

Data Type

Integer

Syntax

applyhomonymsvalue = [objectreference].ApplyHomonyms

[objectreference].ApplyHomonyms = applyhomonymsvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Use this property when you are setting the corresponding rule for proofing the document in Grammar Check.

Equivalent to choosing Edit - Check Grammar, clicking Options, and selecting "Homonyms" in the "Rule type" field on the Rules panel.

This rule flags homonyms or near-homonyms, words that may be confused because they sound alike (for example, "principle" and "principal," "complacent" and "complaisant"). Note that all occurrences of these words will be flagged because there are no contextual clues to distinguish their usage. You must consult the explanations given in the error message to determine whether your usage is correct.

Note This rule is not applicable in every language. Some Apply properties can only be found in specific language versions of Word Pro.

Word Pro: ApplyHomoPhone1 property

{button ,AL(^H_GRAMMAR_CLASS',0)} [See list of classes](#)

(Read-write) A flag that turns the Grammar Check rule on or off for checking homophonic words (one or two or more words, such as "night" and "knight," that are pronounced the same but differ in meaning, origin, and sometimes spelling).

Data Type

Integer

Syntax

applyhomophone1value = [objectreference].ApplyHomoPhone1

[objectreference].ApplyHomoPhone1 = applyhomophone1value

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Use this property when you are setting the corresponding rule for proofing the document in Grammar Check. Equivalent to choosing Edit - Check Grammar, clicking Options, and selecting the appropriate rule in the "Rule type" field on the Rules panel.

Note This rule is not applicable in every language. This property can only be found in specific language versions of Word Pro, for example, Spanish.

Word Pro: ApplyHomoPhone2 property

{button ,AL('H_GRAMMAR_CLASS',0)} [See list of classes](#)

(Read-write) A flag that turns the Grammar Check rule on or off for checking homophonic words (one or two or more words, such as "night" and "knight," that are pronounced the same but differ in meaning, origin, and sometimes spelling).

Data Type

Integer

Syntax

applyhomophone2value = [objectreference].ApplyHomoPhone2

[objectreference].ApplyHomoPhone2 = applyhomophone2value

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Use this property when you are setting the corresponding rule for proofing the document in Grammar Check. Equivalent to choosing Edit - Check Grammar, clicking Options, and selecting the appropriate rule in the "Rule type" field on the Rules panel.

Note This rule is not applicable in every language. This property can only be found in specific language versions of Word Pro, for example, Spanish.

Word Pro: ApplyHomoPhones property

{button ,AL('H_GRAMMAR_CLASS',0)} [See list of classes](#)

(Read-write) A flag that turns the Grammar Check rule on or off for checking homophonic words (one or two or more words, such as "night" and "knight," that are pronounced the same but differ in meaning, origin, and sometimes spelling).

Data Type

Integer

Syntax

applyhomophonesvalue = [objectreference].ApplyHomoPhones

[objectreference].ApplyHomoPhones = applyhomophonesvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Use this property when you are setting the corresponding rule for proofing the document in Grammar Check. Equivalent to choosing Edit - Check Grammar, clicking Options, and selecting the appropriate rule in the "Rule type" field on the Rules panel.

Note This rule is not applicable in every language. This property can only be found in specific language versions of Word Pro, for example, Swedish.

Word Pro: ApplyIncorrectPlural property

{button ,AL('H_GRAMMAR_CLASS',0)} [See list of classes](#)

{button ,AL('H_APPLYINCORRECTPLURAL_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) A flag that turns the Grammar Check rule on or off for checking incorrect plurals.

Data Type

[Integer](#)

Syntax

applyincorrectpluralvalue = [objectreference].ApplyIncorrectPlural

[objectreference].ApplyIncorrectPlural = applyincorrectpluralvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Use this property when you are setting the corresponding rule for proofing the document in Grammar Check. Equivalent to choosing Edit - Check Grammar, clicking Options, and selecting the appropriate rule in the "Rule type" field on the Rules panel.

Note This rule is not applicable in every language. Some Apply properties can only be found in specific language versions of Word Pro.

Word Pro: ApplyInformalExpressions property

{button ,AL('H_GRAMMAR_CLASS',0)} [See list of classes](#)

{button ,AL('H_APPLYINFORMALEXPRESSIONS_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) A flag that turns the Grammar Check rule on or off for checking informal expressions.

Data Type

[Integer](#)

Syntax

applyinformalexpressionsvalue = [objectreference].ApplyInformalExpressions

[objectreference].ApplyInformalExpressions = applyinformalexpressionsvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Use this property when you are setting the corresponding rule for proofing the document in Grammar Check.

Equivalent to choosing Edit - Check Grammar, clicking Options, and selecting "Informal expressions" in the "Rule type" field on the Rules panel.

This rule flags words and expressions that are more appropriate in speech than in writing. This rule offers a less casual alternative or suggests rephrasing the sentence to eliminate the expression. For example, the phrase "a shot at" can be replaced by "a chance to."

Note This rule is not applicable in every language. Some Apply properties can only be found in specific language versions of Word Pro.

Word Pro: ApplyJargonWords property

{button ,AL('H_GRAMMAR_CLASS',0)} [See list of classes](#)

{button ,AL('H_APPLYJARGONWORDS_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) A flag that turns the Grammar Check rule on or off for checking jargon words and expressions.

Data Type

[Integer](#)

Syntax

applyjargonwordsvalue = [objectreference].ApplyJargonWords

[objectreference].ApplyJargonWords = applyjargonwordsvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Use this property when you are setting the corresponding rule for proofing the document in Grammar Check.

Equivalent to choosing Edit - Check Grammar, clicking Options, and selecting "Jargon" in the "Rule type" field on the Rules panel.

This rule flags words and expressions that belong to a specific technical vocabulary (medicine, science, music, and so on), but are inappropriate when used in general writing (for example, "input").

Note This rule is not applicable in every language. Some Apply properties can only be found in specific language versions of Word Pro.

Word Pro: ApplykSplitInfinitives property

{button ,AL('H_GRAMMAR_CLASS',0)} [See list of classes](#)

{button ,AL('H_APPLYKSPLITINFINITIVES_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Determines how the Grammar Check tool will behave when checking for split infinitives.

Data Type

The data type for this property is [Variant](#) which allows the value of this property to be one of the constants listed below or its numeric equivalent (in parentheses).

Syntax

[objectreference].ApplykSplitInfinitives = applyksplitinfinitivesvalue

applyksplitinfinitivesvalue = [objectreference].ApplykSplitInfinitives

Legal values

<u>Value</u>	<u>Effect</u>
0	Don't flag split infinitives.
1	Flag all split infinitives.
2	Flags sentences that have two or more words between an infinitive.
3	Flags sentences that have three or more words between an infinitive.
4	Flags sentences that have four or more words between an infinitive.

Usage

Setting this property is equivalent to making a selection under the "Flag split infinitives" option in the Grammar Options dialog box.

Word Pro: ApplyLowercaseAdjective property

{button ,AL(^H_GRAMMAR_CLASS',0)} [See list of classes](#)

(Read-write) A flag that turns the specific Grammar Check rule on or off.

Data Type

Integer

Syntax

applylowercaseadjectivevalue = [objectreference].ApplyLowercaseAdjective

[objectreference].ApplyLowercaseAdjective = applylowercaseadjectivevalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Use this property when you are setting the corresponding rule for proofing the document in Grammar Check. Equivalent to choosing Edit - Check Grammar, clicking Options, and selecting the appropriate rule in the "Rule type" field on the Rules panel.

Note This rule is not applicable in every language. This property can only be found in specific language versions of Word Pro, for example, German.

Word Pro: ApplyLowercaseColor property

{button ,AL(^H_GRAMMAR_CLASS',0)} [See list of classes](#)

(Read-write) A flag that turns the specific Grammar Check rule on or off.

Data Type

Integer

Syntax

applylowercasecolorvalue = [objectreference].ApplyLowercaseColor

[objectreference].ApplyLowercaseColor = applylowercasecolorvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Use this property when you are setting the corresponding rule for proofing the document in Grammar Check. Equivalent to choosing Edit - Check Grammar, clicking Options, and selecting the appropriate rule in the "Rule type" field on the Rules panel.

Note This rule is not applicable in every language. This property can only be found in specific language versions of Word Pro, for example, German.

Word Pro: ApplyLowercaseNumbers property

{button ,AL('H_GRAMMAR_CLASS',0)} [See list of classes](#)

(Read-write) A flag that turns the specific Grammar Check rule on or off.

Data Type

Integer

Syntax

applylowercasenumbersvalue = [objectreference].ApplyLowercaseNumbers
[objectreference].ApplyLowercaseNumbers = applylowercasenumbersvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Use this property when you are setting the corresponding rule for proofing the document in Grammar Check. Equivalent to choosing Edit - Check Grammar, clicking Options, and selecting the appropriate rule in the "Rule type" field on the Rules panel.

Note This rule is not applicable in every language. This property can only be found in specific language versions of Word Pro, for example, German.

Word Pro: ApplyLowercasePhrases property

{button ,AL('H_GRAMMAR_CLASS',0)} [See list of classes](#)

(Read-write) A flag that turns the specific Grammar Check rule on or off.

Data Type

Integer

Syntax

applylowercasephrasesvalue = [objectreference].ApplyLowercasePhrases [objectreference].ApplyLowercasePhrases
= applylowercasephrasesvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Use this property when you are setting the corresponding rule for proofing the document in Grammar Check. Equivalent to choosing Edit - Check Grammar, clicking Options, and selecting the appropriate rule in the "Rule type" field on the Rules panel.

Note This rule is not applicable in every language. This property can only be found in specific language versions of Word Pro, for example, German.

Word Pro: ApplyLowercasePronouns property

{button ,AL(^H_GRAMMAR_CLASS',0)} [See list of classes](#)

(Read-write) A flag that turns the specific Grammar Check rule on or off.

Data Type

Integer

Syntax

applylowercasepronounsvalue = [objectreference].ApplyLowercasePronouns

[objectreference].ApplyLowercasePronouns = applylowercasepronounsvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Use this property when you are setting the corresponding rule for proofing the document in Grammar Check. Equivalent to choosing Edit - Check Grammar, clicking Options, and selecting the appropriate rule in the "Rule type" field on the Rules panel.

Note This rule is not applicable in every language. This property can only be found in specific language versions of Word Pro, for example, German.

Word Pro: ApplyMassVsCount property

{button ,AL('H_GRAMMAR_CLASS',0)} [See list of classes](#)

{button ,AL('H_APPLYMASSVSCOUNT_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) A flag that turns the Grammar Check rule on or off for checking errors of mass/count agreement.

Data Type

[Integer](#)

Syntax

applymassvscountvalue = [objectreference].ApplyMassVsCount

[objectreference].ApplyMassVsCount = applymassvscountvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Use this property when you are setting the corresponding rule for proofing the document in Grammar Check. Equivalent to choosing Edit - Check Grammar, clicking Options, and selecting "Misspelled expressions" in the "Rule type" field on the Rules panel.

This rule flags errors of mass/count agreement which conflict with the number the noun represents (singular or plural) and the modifying adjectives. For example, one rule flags the sentence, "There are less mistakes in this document," because the adjective, "fewer," not "less" is the correct one to use with the plural noun, "mistakes."

Note This rule is not applicable in every language. Some Apply properties can only be found in specific language versions of Word Pro.

Word Pro: ApplyMisspelledExpressions property

{button ,AL('H_GRAMMAR_CLASS',0)} [See list of classes](#)

{button ,AL('H_APPLYMISSPELLEDEXPRESSIONS_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) A flag that turns the Grammar Check rule on or off for checking misspelled expressions.

Data Type

[Integer](#)

Syntax

applymisspelledexpressionsvalue = [objectreference].ApplyMisspelledExpressions

[objectreference].ApplyMisspelledExpressions = applymisspelledexpressionsvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Use this property when you are setting the corresponding rule for proofing the document in Grammar Check. Equivalent to choosing Edit - Check Grammar, clicking Options, and selecting "Misspelled expressions" in the "Rule type" field on the Rules panel.

Note This rule is not applicable in every language. Some Apply properties can only be found in specific language versions of Word Pro.

Word Pro: ApplyMisspelledForeignExpressions property

{button ,AL('H_GRAMMAR_CLASS',0)} [See list of classes](#)

{button ,AL('H_APPLYMISSPELLEDFOREIGNEXPRESSIONS_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) A flag that turns the Grammar Check rule on or off for checking misspelled foreign expressions.

Data Type

[Integer](#)

Syntax

applymisspelledforeignexpressionsvalue = [objectreference].ApplyMisspelledForeignExpressions

[objectreference].ApplyMisspelledForeignExpressions = applymisspelledforeignexpressionsvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Use this property when you are setting the corresponding rule for proofing the document in Grammar Check.

Equivalent to choosing Edit - Check Grammar, clicking Options, and selecting "Misspelled foreign expressions" in the "Rule type" field on the Rules panel.

This rule flags misspelled foreign expressions. It may also flag typing errors that make a word look like a foreign expression (for example, "esprit di corps" to "esprit de corps.")

Note This rule is not applicable in every language. Some Apply properties can only be found in specific language versions of Word Pro.

Word Pro: ApplyMisspelledItalian property

{button ,AL('H_GRAMMAR_CLASS',0)} [See list of classes](#)

(Read-write) A flag that turns the specific Grammar Check rule on or off.

Data Type

Integer

Syntax

applymisspelleditalianvalue = [objectreference].ApplyMisspelledItalian

[objectreference].ApplyMisspelledItalian = applymisspelleditalianvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Use this property when you are setting the corresponding rule for proofing the document in Grammar Check. Equivalent to choosing Edit - Check Grammar, clicking Options, and selecting the appropriate rule in the "Rule type" field on the Rules panel.

Note This rule is not applicable in every language. Some Apply properties can only be found in specific language versions of Word Pro.

Word Pro: ApplyMisspelledWords property

{button ,AL('H_GRAMMAR_CLASS',0)} [See list of classes](#)

(Read-write) A flag that turns the Grammar Check rule on or off for checking misspelled words.

Data Type

Integer

Syntax

applymisspelledwordsvvalue = [objectreference].ApplyMisspelledWords

[objectreference].ApplyMisspelledWords = applymisspelledwordsvvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Use this property when you are setting the corresponding rule for proofing the document in Grammar Check. Equivalent to choosing Edit - Check Grammar, clicking Options, and selecting the appropriate rule in the "Rule type" field on the Rules panel.

Note This rule is not applicable in every language. Some Apply properties can only be found in specific language versions of Word Pro.

Word Pro: ApplyMisusedWords property

{button ,AL('H_GRAMMAR_CLASS',0)} [See list of classes](#)

{button ,AL('H_APPLYMISUSEDWORDS_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) A flag that turns the Grammar Check rule on or off for checking misused words.

Data Type

[Integer](#)

Syntax

applymisusedwordsvvalue = [objectreference].ApplyMisusedWords

[objectreference].ApplyMisusedWords = applymisusedwordsvvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Use this property when you are setting the corresponding rule for proofing the document in Grammar Check.

Equivalent to choosing Edit - Check Grammar, clicking Options, and selecting "Misused words" in the "Rule type" field on the Rules panel.

This rule flags words or phrases that are often confused with similar words or phrases (for example, "elude to" instead of "allude to," "sit the books on the chair" instead of "set the books on the chair," and so on). The confused expressions should be used in different constructions.

Note This rule is not applicable in every language. Some Apply properties can only be found in specific language versions of Word Pro.

Word Pro: ApplyNonStandardExpression property

{button ,AL('H_GRAMMAR_CLASS',0)} [See list of classes](#)

{button ,AL('H_APPLYNONSTANDARDEXPRESSION_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) A flag that turns the Grammar Check rule on or off for checking non-standard expressions.

Data Type

[Integer](#)

Syntax

applynonstandardexpressionvalue = [objectreference].ApplyNonStandardExpression

[objectreference].ApplyNonStandardExpression = applynonstandardexpressionvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Use this property when you are setting the corresponding rule for proofing the document in Grammar Check.

Equivalent to choosing Edit - Check Grammar, clicking Options, and selecting "Nonstandard terms" in the "Rule type" field on the Rules panel.

Note This rule is not applicable in every language. Some Apply properties can only be found in specific language versions of Word Pro.

Word Pro: ApplyNonStandardModifiers property

{button ,AL('H_GRAMMAR_CLASS',0)} [See list of classes](#)

{button ,AL('H_APPLYNONSTANDARDMODIFIERS_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) A flag that turns the Grammar Check rule on or off for checking non-standard modifiers.

Data Type

[Integer](#)

Syntax

applynonstandardmodifiersvalue = [objectreference].ApplyNonStandardModifiers

[objectreference].ApplyNonStandardModifiers = applynonstandardmodifiersvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Use this property when you are setting the corresponding rule for proofing the document in Grammar Check.

Equivalent to choosing Edit - Check Grammar, clicking Options, and selecting "Nonstandard modifiers" in the "Rule type" field on the Rules panel.

This rule flags errors of modification, such as using adjectives rather than adverbs to modify verbs. For example, the rule will flag the sentence, "His new car really drives good," since "good" is an adjective mistakenly used in place of the adverb, "well." This rule also checks to determine whether the hyphenation of modifiers conforms to standard usage.

Note This rule is not applicable in every language. Some Apply properties can only be found in specific language versions of Word Pro.

Word Pro: ApplyNoudModifierOrderCheck property

{button ,AL('H_GRAMMAR_CLASS',0)} [See list of classes](#)

{button ,AL('H_APPLYNOUDMODIFIERORDERCHECK_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) A flag that turns the Grammar Check rule on or off for checking errors in word order.

Data Type

[Integer](#)

Syntax

applynoudmodifierordercheckvalue = [objectreference].ApplyNoudModifierOrderCheck

[objectreference].ApplyNoudModifierOrderCheck = applynoudmodifierordercheckvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Use this property when you are setting the corresponding rule for proofing the document in Grammar Check. Equivalent to choosing Edit - Check Grammar, clicking Options, and selecting "Word order errors" in the "Rule type" field on the Rules panel.

This rule flags the incorrect order of certain words that modify nouns, for example, "my both children" instead of "both my children."

Note This rule is not applicable in every language. Some Apply properties can only be found in specific language versions of Word Pro.

Word Pro: ApplyNounConsistency property

{button ,AL('H_GRAMMAR_CLASS',0)} [See list of classes](#)

{button ,AL('H_APPLYNOUNCONSISTENCY_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) A flag that turns the Grammar Check rule on or off for checking noun consistency errors.

Data Type

Data type is [Integer](#)

Syntax

applynounconsistencyvalue = [objectreference].ApplyNounConsistency [objectreference].ApplyNounConsistency = applynounconsistencyvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Use this property when you are setting the corresponding rule for proofing the document in Grammar Check. Equivalent to choosing Edit - Check Grammar, clicking Options, and selecting "Noun phrase consistency errors" in the "Rule type" field on the Rules panel.

This rule flags errors of number agreement within noun phrases. A noun phrase consists of a noun and the words that modify it, for example, "this old man," "that red bicycle," "a tall building." Certain modifiers, such as "this," "that," and "a" are singular and must be used with singular nouns. Other modifiers, such as "these," "those," "both," and "many," must be used with plural nouns. This rule flags the sentence, "These five machine are still under warranty," because "these" is plural and "machine" is singular.

Note This rule is not applicable in every language. Some Apply properties can only be found in specific language versions of Word Pro.

Word Pro: ApplyNounPhraseAgree property

{button ,AL('H_GRAMMAR_CLASS',0)} [See list of classes](#)

(Read-write) A flag that turns the specific Grammar Check rule on or off.

Data Type

Integer

Syntax

applynounphraseagreevalue = [objectreference].ApplyNounPhraseAgree

[objectreference].ApplyNounPhraseAgree = applynounphraseagreevalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Use this property when you are setting the corresponding rule for proofing the document in Grammar Check. Equivalent to choosing Edit - Check Grammar, clicking Options, and selecting the appropriate rule in the "Rule type" field on the Rules panel.

Note This rule is not applicable in every language. Some Apply properties can only be found in specific language versions of Word Pro.

Word Pro: ApplyNSAdjective property

{button ,AL('H_GRAMMAR_CLASS',0)} [See list of classes](#)

(Read-write) A flag that turns the specific Grammar Check rule on or off.

Data Type

Integer

Syntax

applynsadjectivevalue = [objectreference].ApplyNSAdjective

[objectreference].ApplyNSAdjective = applynsadjectivevalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Use this property when you are setting the corresponding rule for proofing the document in Grammar Check. Equivalent to choosing Edit - Check Grammar, clicking Options, and selecting the appropriate rule in the "Rule type" field on the Rules panel.

Note This rule is not applicable in every language. Some Apply properties can only be found in specific language versions of Word Pro.

Word Pro: ApplyNSClause property

{button ,AL('H_GRAMMAR_CLASS',0)} [See list of classes](#)

(Read-write) A flag that turns the specific Grammar Check rule on or off.

Data Type

Integer

Syntax

applynsclausevalue = [objectreference].ApplyNSClause

[objectreference].ApplyNSClause = applynsclausevalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Use this property when you are setting the corresponding rule for proofing the document in Grammar Check. Equivalent to choosing Edit - Check Grammar, clicking Options, and selecting the appropriate rule in the "Rule type" field on the Rules panel.

Note This rule is not applicable in every language. Some Apply properties can only be found in specific language versions of Word Pro.

Word Pro: ApplyNSCompare property

{button ,AL('H_GRAMMAR_CLASS',0)} [See list of classes](#)

(Read-write) A flag that turns the specific Grammar Check rule on or off.

Data Type

Integer

Syntax

applynscomparevalue = [objectreference].ApplyNSCompare

[objectreference].ApplyNSCompare = applynscomparevalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Use this property when you are setting the corresponding rule for proofing the document in Grammar Check. Equivalent to choosing Edit - Check Grammar, clicking Options, and selecting the appropriate rule in the "Rule type" field on the Rules panel.

Note This rule is not applicable in every language. Some Apply properties can only be found in specific language versions of Word Pro.

Word Pro: ApplyNSContract property

{button ,AL('H_GRAMMAR_CLASS',0)} [See list of classes](#)

(Read-write) A flag that turns the specific Grammar Check rule on or off.

Data Type

Integer

Syntax

applynscontractvalue = [objectreference].ApplyNSContract

[objectreference].ApplyNSContract = applynscontractvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Use this property when you are setting the corresponding rule for proofing the document in Grammar Check. Equivalent to choosing Edit - Check Grammar, clicking Options, and selecting the appropriate rule in the "Rule type" field on the Rules panel.

Note This rule is not applicable in every language. Some Apply properties can only be found in specific language versions of Word Pro.

Word Pro: ApplyNSGeography property

{button ,AL('H_GRAMMAR_CLASS',0)} [See list of classes](#)

(Read-write) A flag that turns the specific Grammar Check rule on or off.

Data Type

Integer

Syntax

applynsgeographyvalue = [objectreference].ApplyNSGeography

[objectreference].ApplyNSGeography = applynsgeographyvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Use this property when you are setting the corresponding rule for proofing the document in Grammar Check. Equivalent to choosing Edit - Check Grammar, clicking Options, and selecting the appropriate rule in the "Rule type" field on the Rules panel.

Note This rule is not applicable in every language. Some Apply properties can only be found in specific language versions of Word Pro.

Word Pro: ApplyNSInflection property

{button ,AL(^H_GRAMMAR_CLASS',0)} [See list of classes](#)

(Read-write) A flag that turns the specific Grammar Check rule on or off.

Data Type

Integer

Syntax

applynsinflectionvalue = [objectreference].ApplyNSInflection

[objectreference].ApplyNSInflection = applynsinflectionvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Use this property when you are setting the corresponding rule for proofing the document in Grammar Check. Equivalent to choosing Edit - Check Grammar, clicking Options, and selecting the appropriate rule in the "Rule type" field on the Rules panel.

Note This rule is not applicable in every language. Some Apply properties can only be found in specific language versions of Word Pro.

Word Pro: ApplyNSNegation property

{button ,AL('H_GRAMMAR_CLASS',0)} [See list of classes](#)

(Read-write) A flag that turns the specific Grammar Check rule on or off.

Data Type

Integer

Syntax

applynsnegationvalue = [objectreference].ApplyNSNegation

[objectreference].ApplyNSNegation = applynsnegationvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Use this property when you are setting the corresponding rule for proofing the document in Grammar Check. Equivalent to choosing Edit - Check Grammar, clicking Options, and selecting the appropriate rule in the "Rule type" field on the Rules panel.

Note This rule is not applicable in every language. Some Apply properties can only be found in specific language versions of Word Pro.

Word Pro: ApplyNSPrep property

{button ,AL('H_GRAMMAR_CLASS',0)} [See list of classes](#)

(Read-write) A flag that turns the specific Grammar Check rule on or off.

Data Type

Integer

Syntax

applynsprepvalue = [objectreference].ApplyNSPrep

[objectreference].ApplyNSPrep = applynsprepvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Use this property when you are setting the corresponding rule for proofing the document in Grammar Check. Equivalent to choosing Edit - Check Grammar, clicking Options, and selecting the appropriate rule in the "Rule type" field on the Rules panel.

Note This rule is not applicable in every language. Some Apply properties can only be found in specific language versions of Word Pro.

Word Pro: ApplyNSPronoun property

{button ,AL('H_GRAMMAR_CLASS',0)} [See list of classes](#)

(Read-write) A flag that turns the specific Grammar Check rule on or off.

Data Type

Integer

Syntax

applynspronounvalue = [objectreference].ApplyNSPronoun

[objectreference].ApplyNSPronoun = applynspronounvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Use this property when you are setting the corresponding rule for proofing the document in Grammar Check. Equivalent to choosing Edit - Check Grammar, clicking Options, and selecting the appropriate rule in the "Rule type" field on the Rules panel.

Note This rule is not applicable in every language. Some Apply properties can only be found in specific language versions of Word Pro.

Word Pro: ApplyNSSpell property

{button ,AL('H_GRAMMAR_CLASS',0)} [See list of classes](#)

(Read-write) A flag that turns the specific Grammar Check rule on or off.

Data Type

Integer

Syntax

applynspellvalue = [objectreference].ApplyNSSpell

[objectreference].ApplyNSSpell = applynspellvalu

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Use this property when you are setting the corresponding rule for proofing the document in Grammar Check. Equivalent to choosing Edit - Check Grammar, clicking Options, and selecting the appropriate rule in the "Rule type" field on the Rules panel.

Note This rule is not applicable in every language. Some Apply properties can only be found in specific language versions of Word Pro.

Word Pro: ApplyNSUsage property

{button ,AL('H_GRAMMAR_CLASS',0)} [See list of classes](#)

(Read-write) A flag that turns the specific Grammar Check rule on or off.

Data Type

Integer

Syntax

applynsusagevalue = [objectreference].ApplyNSUsage

[objectreference].ApplyNSUsage = applynsusagevalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Use this property when you are setting the corresponding rule for proofing the document in Grammar Check. Equivalent to choosing Edit - Check Grammar, clicking Options, and selecting the appropriate rule in the "Rule type" field on the Rules panel.

Note This rule is not applicable in every language. Some Apply properties can only be found in specific language versions of Word Pro.

Word Pro: ApplyNSVerbForm property

{button ,AL('H_GRAMMAR_CLASS',0)} [See list of classes](#)

(Read-write) A flag that turns the specific Grammar Check rule on or off.

Data Type

Integer

Syntax

applynsverbformvalue = [objectreference].ApplyNSVerbForm

[objectreference].ApplyNSVerbForm = applynsverbformvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Use this property when you are setting the corresponding rule for proofing the document in Grammar Check. Equivalent to choosing Edit - Check Grammar, clicking Options, and selecting the appropriate rule in the "Rule type" field on the Rules panel.

Note This rule is not applicable in every language. Some Apply properties can only be found in specific language versions of Word Pro.

Word Pro: ApplyOpenClosedSpelling property

{button ,AL('H_GRAMMAR_CLASS',0)} [See list of classes](#)

{button ,AL('H_APPLYOPENCLOSEDSPELLING_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) A flag that turns the Grammar Check rule on or off for checking incorrect use of spaces in words or phrases.

Data Type

[Integer](#)

Syntax

applyopenclosedspellingvalue = [objectreference].ApplyOpenClosedSpelling

[objectreference].ApplyOpenClosedSpelling = applyopenclosedspellingvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Use this property when you are setting the corresponding rule for proofing the document in Grammar Check.

Equivalent to choosing Edit - Check Grammar, clicking Options, and selecting "Open vs. closed spelling" in the "Rule type" field on the Rules panel.

This rule flags spelling errors that result from incorrect use of spaces. The correct spelling is offered as an alternative (for example, "in a while" to "in awhile").

Note This rule is not applicable in every language. Some Apply properties can only be found in specific language versions of Word Pro.

Word Pro: ApplyOpenUsage property

{button ,AL('H_GRAMMAR_CLASS',0)} [See list of classes](#)

{button ,AL('H_APPLYOPENUSAGE_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) A flag that turns the specific Grammar Check rule on or off.

Data Type

[Integer](#)

Syntax

applyopenusagevalue = [objectreference].ApplyOpenUsage

[objectreference].ApplyOpenUsage = applyopenusagevalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Use this property when you are setting the corresponding rule for proofing the document in Grammar Check. Equivalent to choosing Edit - Check Grammar, clicking Options, and selecting the appropriate rule in the "Rule type" field on the Rules panel.

Note This rule is not applicable in every language. Some Apply properties can only be found in specific language versions of Word Pro.

Word Pro: ApplyOverusedPhrases property

{button ,AL('H_GRAMMAR_CLASS',0)} [See list of classes](#)

{button ,AL('H_APPLYOVERUSEDPHRASES_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) A flag that turns the Grammar Check rule on or off for checking overused phrases.

Data Type

[Integer](#)

Syntax

applyoverusedphrasesvalue = [objectreference].ApplyOverusedPhrases

[objectreference].ApplyOverusedPhrases = applyoverusedphrasesvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Use this property when you are setting the corresponding rule for proofing the document in Grammar Check.

Equivalent to choosing Edit - Check Grammar, clicking Options, and selecting "Overused phrases" in the "Rule type" field on the Rules panel.

This rule flags overused expressions that have lost their original impact. In a spoken context, they might be acceptable, but in writing, the phrases are too casual and should be replaced with expressions that are less colloquial and more precise. In some cases, the sentence must be rephrased to avoid an overused expression. For example, the phrase, "blissful ignorance" might be replaced by just the word, "ignorance."

Note This rule is not applicable in every language. Some Apply properties can only be found in specific language versions of Word Pro.

Word Pro: ApplyPassiveVerbErrors property

{button ,AL(^H_GRAMMAR_CLASS',0)} [See list of classes](#)

{button ,AL(^H_APPLYPASSIVEVERBERRORS_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) A flag that turns the Grammar Check rule on or off for checking the use of passive voice.

Data Type

[Integer](#)

Syntax

applypassiveverberrorsvalue = [objectreference].ApplyPassiveVerbErrors

[objectreference].ApplyPassiveVerbErrors = applypassiveverberrorsvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Use this property when you are setting the corresponding rule for proofing the document in Grammar Check. Equivalent to choosing Edit - Check Grammar, clicking Options, and selecting "Passive voice usage" in the "Rule type" field on the Rules panel.

This rule flags use of the passive voice, where the subject is acted upon but completes no action, as in "Our proposal was accepted by the board." The counterpart in active voice is, "The board accepted our proposal."

Sentences written in the passive voice can sound weak and indirect; it is often better to rephrase them as active. Occasionally, a sentence makes sense only in the passive voice, but most sentences are more coherent when the subject completes the action.

Note This rule is not applicable in every language. Some Apply properties can only be found in specific language versions of Word Pro.

Word Pro: ApplyPostClitAgree property

{button ,AL('H_GRAMMAR_CLASS',0)} [See list of classes](#)

(Read-write) A flag that turns the specific Grammar Check rule on or off.

Data Type

Integer

Syntax

applypostclitagreevalue = [objectreference].ApplyPostClitAgree

[objectreference].ApplyPostClitAgree = applypostclitagreevalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Use this property when you are setting the corresponding rule for proofing the document in Grammar Check. Equivalent to choosing Edit - Check Grammar, clicking Options, and selecting the appropriate rule in the "Rule type" field on the Rules panel.

Note This rule is not applicable in every language. This property can only be found in specific language versions of Word Pro, for example, Italian.

Word Pro: ApplyPrepExpression property

{button ,AL('H_GRAMMAR_CLASS',0)} [See list of classes](#)

(Read-write) A flag that turns the specific Grammar Check rule on or off.

Data Type

[Integer](#)

Syntax

applyprepexpressionvalue = [objectreference].ApplyPrepExpression

[objectreference].ApplyPrepExpression = applyprepexpressionvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Use this property when you are setting the corresponding rule for proofing the document in Grammar Check. Equivalent to choosing Edit - Check Grammar, clicking Options, and selecting the appropriate rule in the "Rule type" field on the Rules panel.

Note This rule is not applicable in every language. Some Apply properties can only be found in specific language versions of Word Pro.

Word Pro: ApplyPrepositionalPhrases property

{button ,AL('H_GRAMMAR_CLASS',0)} [See list of classes](#)

{button ,AL('H_APPLYPREPOSITIONALPHRASES_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Determines whether or not the Grammar Check tool will flag prepositional phrases in a document.

Data Type

[Integer](#)

Syntax

applyprepositionalphrasesvalue = [objectreference].ApplyPrepositionalPhrases

[objectreference].ApplyPrepositionalPhrases = applyprepositionalphrasesvalue

Legal values

<u>Value</u>	<u>Effect</u>
0	Never flag prepositional phrases.
1	Flag 3 or more prepositional phrases in a row.
2	Flag 4 or more prepositional phrases in a row.
3	Flag 5 or more prepositional phrases in a row.

Usage

Equivalent to the "Flag consecutive prepositional phrases" option on the Grammatical Style panel of the Grammar Options dialog box.

This rule flags expressions that include an unnecessary preposition and suggests deleting it to make the expression more concise. For example, in the sentence, "I sat down on the lawn," the preposition "down" is superfluous since it is implied by "sat."

Note This rule is not applicable in every language. Some Apply properties can only be found in specific language versions of Word Pro.

Word Pro: ApplyPretentiousWords property

{button ,AL('H_GRAMMAR_CLASS',0)} [See list of classes](#)

{button ,AL('H_APPLYPRETENTIOUSWORDS_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) A flag that turns the Grammar Check rule on or off for checking pretentious words.

Data Type

[Integer](#)

Syntax

applypretentiouswordsvvalue = [objectreference].ApplyPretentiousWords [objectreference].ApplyPretentiousWords = applypretentiouswordsvvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Use this property when you are setting the corresponding rule for proofing the document in Grammar Check. Equivalent to choosing Edit - Check Grammar, clicking Options, and selecting "Pretentious words" in the "Rule type" field on the Rules panel.

This rule flags unnecessarily complex words and offers simple, straightforward alternatives. For example, the word, "eventuate," can usually be replaced with the expression, "takes place," or "happens."

Note This rule is not applicable in every language. Some Apply properties can only be found in specific language versions of Word Pro.

Word Pro: ApplyPronounErrors property

{button ,AL('H_GRAMMAR_CLASS',0)} [See list of classes](#)

{button ,AL('H_APPLYPRONOUNERRORS_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) A flag that turns the Grammar Check rule on or off for checking pronoun errors.

Data Type

[Integer](#)

Syntax

applypronounerrorsvalue = [objectreference].ApplyPronounErrors

[objectreference].ApplyPronounErrors = applypronounerrorsvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Use this property when you are setting the corresponding rule for proofing the document in Grammar Check.

Equivalent to choosing Edit - Check Grammar, clicking Options, and selecting "Pronoun errors" in the "Rule type" field on the Rules panel.

This rule checks pronouns for errors in case and order. For example, one rule flags the sentence, "They are baking a cake for my sister and I," and notes the pronoun "I" is incorrect. The error message explains that because "I" is the object of a preposition, it should be the objective case (me). This rule will also flag the relative pronoun, "which," when used in a restrictive clause and recommends using "that."

Note This rule is not applicable in every language. Some Apply properties can only be found in specific language versions of Word Pro.

Word Pro: ApplyPunctuationErrors property

{button ,AL('H_GRAMMAR_CLASS',0)} [See list of classes](#)

{button ,AL('H_APPLYPUNCTUATIONERRORS_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) A flag that turns the Grammar Check rule on or off for checking punctuation errors.

Data Type

[Integer](#)

Syntax

applypunctuationerrorsvalue = [objectreference].ApplyPunctuationErrors

[objectreference].ApplyPunctuationErrors = applypunctuationerrorsvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Use this property when you are setting the corresponding rule for proofing the document in Grammar Check.

Equivalent to choosing Edit - Check Grammar, clicking Options, and selecting "Punctuation errors" in the "Rule type" field on the Rules panel.

This rule flags punctuation errors, such as the incorrect placement of commas in specific expressions and inappropriate punctuation of parenthetical or quoted material. It also flags doubled punctuation.

Note This rule is not applicable in every language. Some Apply properties can only be found in specific language versions of Word Pro.

Word Pro: ApplyRedundantExpressions property

{button ,AL('H_GRAMMAR_CLASS',0)} [See list of classes](#)

{button ,AL('H_APPLYREDUNDANTEXPRESSIONS_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) A flag that turns the Grammar Check rule on or off for checking redundant expressions.

Data Type

[Integer](#)

Syntax

applyredundantexpressionsvalue = [objectreference].ApplyRedundantExpressions

[objectreference].ApplyRedundantExpressions = applyredundantexpressionsvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Use this property when you are setting the corresponding rule for proofing the document in Grammar Check. Equivalent to choosing Edit - Check Grammar, clicking Options, and selecting "Redundant expressions" in the "Rule type" field on the Rules panel.

This rule flags expressions containing multiple words that mean or imply the same thing. Redundancy can often be eliminated by deleting part of the expression. For example, "sufficient enough" might become either "sufficient" or "enough."

Note This rule is not applicable in every language. Some Apply properties can only be found in specific language versions of Word Pro.

Word Pro: ApplyRegionalExpression property

{button ,AL('H_GRAMMAR_CLASS',0)} [See list of classes](#)

(Read-write) A flag that turns the specific Grammar Check rule on or off.

Data Type

Integer

Syntax

applyregionalexpressionvalue = [objectreference].ApplyRegionalExpression

[objectreference].ApplyRegionalExpression = applyregionalexpressionvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Use this property when you are setting the corresponding rule for proofing the document in Grammar Check. Equivalent to choosing Edit - Check Grammar, clicking Options, and selecting the appropriate rule in the "Rule type" field on the Rules panel.

Note This rule is not applicable in every language. Some Apply properties can only be found in specific language versions of Word Pro.

Word Pro: ApplyRelatedWord property

{button ,AL(^H_GRAMMAR_CLASS',0)} [See list of classes](#)

(Read-write) A flag that turns the specific Grammar Check rule on or off.

Data Type

Integer

Syntax

applyrelatedwordvalue = [objectreference].ApplyRelatedWord

[objectreference].ApplyRelatedWord = applyrelatedwordvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Use this property when you are setting the corresponding rule for proofing the document in Grammar Check. Equivalent to choosing Edit - Check Grammar, clicking Options, and selecting the appropriate rule in the "Rule type" field on the Rules panel.

Note This rule is not applicable in every language. Some Apply properties can only be found in specific language versions of Word Pro.

Word Pro: Action property

{button ,AL('H_CLICKHERE_CLASS',0)} [See list of classes](#)

{button ,AL('H_CLICKHERE_ACTION_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Indicates how a ClickHere block behaves.

Data Type

Data type is Integer.

Syntax

actionvalue = [objectreference].Action

[objectreference].Action = actionvalue

Legal values

Must be an integer from 1 to 10.

<u>Value</u>	<u>Effect</u>
1 = Text	Plain text block
2 = Table	Create table dialog
3 = Picture	Import Picture dialog
4 = OleObject	Insert OLE Object dialog
5 = Chart	Creates a chart
6 = Drawing	Creates a drawing
7 = Import/Export	Import/Export Dialog
8 = Glossary	Insert glossary Item dialog
9 = Equation	Creates an equation frame
10 = InternetLink	Stores a URL which is opened when you click on the ClickHere block
11 = Keyword List	Select Keyword dialog
12 = Insert Date/Time	Insert Date/Time dialog
13 = Insert Symbol	Insert Symbol bar
14 = Doc Fields	Document Fields dialog
15 = Insert Page Number	Insert Page Number dialog

Usage

OLE is not supported in OS/2.

Word Pro: Action property

{button ,AL(^H_MENUITEM_CLASS',0)} [See list of classes](#)

(Read-write) The name of a script function or a WMCommand ID that is executed when a menu item is selected.

Data Type

Data type is String.

Syntax

actionvalue = [objectreference].Action

[objectreference].Action = actionvalue

Legal values

Usage

When you select a menu item, Word Pro executes the Action property for the item. The Action property can be the name of a script function that you create, or it can be the value of a Word Pro WMCommand. WMCommands are unique IDs that Word Pro uses to differentiate each menu item.

In order to specify an external subroutine or function in the Action property, use the following syntax:

```
"filename!subroutine"
```

For example, if you have an external LotusScript file in the current directory called *exsubs.lss*, and it contains a subroutine called *mysub*, you can assign that subroutine to a MenuItem object's Action property in the following way:

```
mymenuitem.action = "exsubs.lss!mysub"
```

If the subroutine *mysub* is contained within the current document, you can assign it to a MenuItem object's Action property in the following way:

```
mymenuitem.action = "!mysub"
```

In order to assign a standard Word Pro menu command to a menu item, you can assign the WMCommand ID to the MenuItem object's Action property in the following way:

```
mymenuitem.action = "101"
```

Note that the WMCommand ID must be assigned as a String value. For more information about WMCommand IDs and a complete listing of the appropriate values, see the WMCommand method topic.

Word Pro: AbortURLDownload method

{button ,AL(^H_OLEOBJECT_CLASS',0)} [See list of classes](#)

Aborts the download of the OLE object from which the method is called. Equivalent to clicking the stop button (a red X) on the Web Tools bar.

Syntax

[objectreference].AbortURLDownload()

Parameters

none

Return value

Always returns a value of -1.

Usage

Use this method on documents opened from the internet. This method allows you to abort the download of OLE objects such as graphics and controls. By using a Forall statement, you can iterate over the entire collection of OLEObject objects, aborting the download of each one.

```
Forall objs in .Division.Foundry.OleObjects
  objs.AbortUrlDownload
End Forall
```

Note You may have to experiment with the timing of this method. If your internet connection is particularly fast, the OLE objects may finish downloading before you can run the script.

Word Pro: AcquireTWAINImage method

{button ,AL('H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL('H_ACQUIRETWAINIMAGE_METHOD_EXSCRIPT',1)} [See example](#)

This language element has not yet been defined. You can check for a newer Help file on the Lotus web site.

Syntax

[objectreference].AcquireTWAINImage()

Parameters

none

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Word Pro: AddContainer method

{button ,AL(^H_WPAPPLICATION_CLASS;H_BASECONTAINER_CLASS;H_CELLCONTAINER_CLASS;H_FRAME_CONTAINER_CLASS;H_NOTECONTAINER_CLASS;H_RUBYCONTAINER_CLASS;H_DROPDOWNCONTAINER_CLASS;H_PAGECONTAINER_CLASS;H_SUBPAGECONTAINER_CLASS;H_SUPERPAGECONTAINER_CLASS;H_ROWCONTAINER_CLASS;H_SUPERTABLECONTAINER_CLASS;H_TABLECONTAINER_CLASS;H_PARALLELCONTAINER_CLASS;H_TABLEONLYCONTAINER_CLASS;';0)} [See list of classes](#)

This language element has not yet been defined. You can check for a newer Help file on the Lotus web site.

Syntax

[objectreference].AddContainer(String LayoutName)

Parameters

LayoutName

The data type for this parameter is String. There is no default value. You must provide a value for this parameter.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Word Pro: AddIndexAllEntry method

{button ,AL('H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL('H_ADDINDEXALLENTY_METHOD_EXSCRIPT',1)} [See example](#)

This language element has not yet been defined. You can check for a newer Help file on the Lotus web site.

Syntax

[objectreference].AddIndexAllEntry(String WordToMark, String IndexEntry, String ViceVersaEntry)

Parameters

WordToMark

The data type for this parameter is String. There is no default value. You must provide a value for this parameter.

IndexEntry

The data type for this parameter is String. There is no default value. You must provide a value for this parameter.

ViceVersaEntry

The data type for this parameter is String. There is no default value. You must provide a value for this parameter.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Word Pro: CanRepeatClickHere method

{button ,AL('H_CLICKHERE_CLASS;H_TEXTMARKER_CLASS;H_TEXT_CLASS',0)} [See list of classes](#)

{button ,AL('H_CANREPEATCLICKHERE_METHOD_EXSCRIPT',1)} [See example](#)

Determines if you can legally create a repeating Click Here Block at the insertion point, which will repeat the Click Here specified in the method's argument.

Syntax

[objectreference].CanRepeatClickHere(String TargetClickHere)

Parameters

TargetClickHere

String. There is no default value. You must provide a value for this parameter.

Return value

This method returns a value of -1 (True) if you can create the Click Here Block, or 0 (False) if you cannot.

Usage

Equivalent to choosing Create - Create Click Here Block and selecting "Repeat Existing Click Here Block."

Word Pro: ClearSpellHints method

{button ,AL(^H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

This language element has not yet been defined. You can check for a newer Help file on the Lotus web site.

Syntax

[objectreference].ClearSpellHints()

Parameters

none

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Word Pro: CopyDivision method

{button ,AL('H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL('H_COPYDIVISION_METHOD_EXSCRIPT',1)} [See example](#)

Copies the specified division to the Clipboard. If you don't specify a division, this method copies the currently active division.

Syntax

[objectreference].CopyDivision([DivisionName])

Parameters

DivisionName

The [internal name](#) of the division you want to copy. The data type for this parameter is String. There is no default value. This is an optional parameter. If you don't provide a value for this parameter, Word Pro copies the currently active division.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Word Pro: CreateOleFromFile method

{button ,AL('H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL('H_CREATEOLEFROMFILE_METHOD_EXSCRIPT',1)} [See example](#)

This language element has not yet been defined. You can check for a newer Help file on the Lotus web site.

Syntax

[objectreference].CreateOleFromFile(String FilePath, Intbool Link, Long IconMetaFilePictHandle=0)

Parameters

FilePath

The data type for this parameter is String. There is no default value. You must provide a value for this parameter.

Link

The data type for this parameter is Integer. The legal values for this parameter are -1 and 0 but you may use the LotusScript constants True (-1) and False (0) instead of the integer values. This is an optional parameter. There is no default value. You must provide a value for this parameter.

IconMetaFilePictHandle

The data type for this parameter is Long. The default value for this parameter is 0. This is an optional parameter.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Word Pro: CreateTextEntryField method

{button ,AL('H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL('H_CREATETEXTENTRYFIELD_METHOD_EXSCRIPT',1)} [See example](#)

This language element has not yet been defined. You can check for a newer Help file on the Lotus web site.

Syntax

[objectreference].CreateTextEntryField(Longtwips FieldWidth, BorderPattern BorderPattern, Longtwips BorderWidth, Long BorderColor)

Parameters

FieldWidth

Specifies the length of the Text Entry Field in Twips. There are 1440 Twips per inch. The data type of this parameter is Long. There is no default value. You must provide a value for this parameter.

BorderPattern

The data type for this parameter is Variant which allows the value of this parameter to be one of the constants listed below or its numeric equivalent (in parentheses). There is no default value. You must provide a value for this parameter.

- \$LtsBorderPatternBorderDot (1056964663)
- \$LtsBorderPatternDashDot (1056964659)
- \$LtsBorderPatternDashDotDot (1056964660)
- \$LtsBorderPatternDashed (1056964662)
- \$LtsBorderPatternDot (2498)
- \$LtsBorderPatternDouble (1056964666)
- \$LtsBorderPatternLongDash (1056964661)
- \$LtsBorderPatternNone (1056964657)
- \$LtsBorderPatternSolid (1056964658)
- \$LwpBorderPattern13space (36)
- \$LwpBorderPattern31space (37)
- \$LwpBorderPatternButttdown (35)
- \$LwpBorderPatternButtonup (34)
- \$LwpBorderPatternCircle (41)
- \$LwpBorderPatternDblThick (51)
- \$LwpBorderPatternDblWavy (56)
- \$LwpBorderPatternDeco1 (44)
- \$LwpBorderPatternDeco2 (45)
- \$LwpBorderPatternDeco3 (50)
- \$LwpBorderPatternDiagonal (38)
- \$LwpBorderPatternGirder (2571)
- \$LwpBorderPatternPin (47)
- \$LwpBorderPatternRain (46)
- \$LwpBorderPatternRope (43)
- \$LwpBorderPatternRose (48)
- \$LwpBorderPatternStar (42)
- \$LwpBorderPatternSunf (49)
- \$LwpBorderPatternTaro (39)
- \$LwpBorderPatternThickDblwavy (58)
- \$LwpBorderPatternThickThin (53)
- \$LwpBorderPatternThickWavy (57)
- \$LwpBorderPatternThinThick (54)

\$LwpBorderPatternThinThickThin (52)

\$LwpBorderPatternWarning (2379)

\$LwpBorderPatternWavy (55)

\$LwpLtsBorderPatternDot (40)

BorderWidth

Specifies the width of the border line in Twips. There are 1440 Twips per inch. The data type of this parameter is Long. There is no default value. You must provide a value for this parameter.

BorderColor

The data type for this parameter is Long. There is no default value. You must provide a value for this parameter.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Word Pro: CutDivision method

{button ,AL('H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL('H_CUTDIVISION_METHOD_EXSCRIPT',1)} [See example](#)

Deletes the specified division from the document and places it in the Clipboard. If you don't specify a division, this method cuts the currently active division.

Syntax

[objectreference].CutDivision([DivisionName])

Parameters

DivisionName

The [internal name](#) of the division you want to cut. The data type for this parameter is String. There is no default value.

This is an optional parameter. If you don't provide a value for this parameter, Word Pro cuts the currently active division.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Word Pro: DestroyPowerFields method

{button ,AL('H_FOUNDRY_CLASS',0)} [See list of classes](#)

This language element has not yet been defined. You can check for a newer Help file on the Lotus web site.

Syntax

[objectreference].DestroyPowerFields(PfType Type=FldNone)

Parameters

Type

Data type is Variant, which allows the value of this parameter to be one of the constants listed below or its numeric equivalent (in parentheses). Default is FldNone. Optional parameter.

<u>Value</u>	<u>Effect</u>
\$LwpPFTypeBookmark (2003)	
\$LwpPFTypeButton (2007)	
\$LwpPFTypeDde (2002)	
\$LwpPFTypeDocvar (2011)	
\$LwpPFTypeField (2004)	
\$LwpPFTypeIndex (2009)	
\$LwpPFTypeMarker (2010)	
\$LwpPFTypeMergevar (2013)	
\$LwpPFTypeNone (2517)	
\$LwpPFTypePrtescape (2008)	
\$LwpPFTypeSeq (2005)	
\$LwpPFTypeSet (2006)	
\$LwpPFTypeToc (2012)	

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Word Pro: DoesMultiCellPaste method

```
{button ,AL(^H_BASECONTAINER_CLASS;H_CELLCONTAINER_CLASS;H_FRAMECONTAINER_CLASS;H_NOTE  
CONTAINER_CLASS;H_RUBYCONTAINER_CLASS;H_DROPDOWNCONTAINER_CLASS;H_PAGECONTAINER_  
CLASS;H_SUBPAGECONTAINER_CLASS;H_SUPERPAGECONTAINER_CLASS;H_ROWCONTAINER_CLASS;  
H_SUPERTABLECONTAINER_CLASS;H_TABLECONTAINER_CLASS;H_PARALLELCOLSCONTAINER_CLASS  
;H_TABLEONLYCONT_CLASS;H_GRAPHICOLEBJECT_CLASS;H_GRAPHIC_CLASS;H_OLEOBJECT_CLAS  
S;H_CLICKHERE_CLASS;H_TEXTMARKER_CLASS;H_TEXT_CLASS',0)} See list of classes
```

Indicates whether or not the container which is uppermost in the focus is capable of handling a multi-cell paste operation. While this method is inherited by many containers, it is only pertinent to a few.

When called from Graphic, GraphicOleObject, OleObject:

Do not call this method from the Graphic, GraphicOleObject, or OleObject classes.

Syntax

[objectreference].DoesMultiCellPaste()

Parameters

None.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Word Pro: DoVerb method

{button ,AL('H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL('H_DOVERB_METHOD_EXSCRIPT',1)} [See example](#)

This language element has not yet been defined. You can check for a newer Help file on the Lotus web site.

Syntax

[objectreference].DoVerb(Long Verb=0)

Parameters

Verb

The data type for this parameter is Long. The default value for this parameter is 0. This is an optional parameter.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Word Pro: EditWrap method

```
{button ,AL('H_BASECONTAINER_CLASS;H_CELLCONTAINER_CLASS;H_FRAMECONTAINER_CLASS;H_NOTE  
CONTAINER_CLASS;H_RUBYCONTAINER_CLASS;H_DROPCAPCONTAINER_CLASS;H_PAGECONTAINER_  
CLASS;H_SUBPAGECONTAINER_CLASS;H_SUPERPAGECONTAINER_CLASS;H_ROWCONTAINER_CLASS;  
H_SUPERTABLECONTAINER_CLASS;H_TABLECONTAINER_CLASS;H_PARALLELCOLSCONTAINER_CLASS  
;H_TABLEONLYCONT_CLASS;';0)} See list of classes
```

```
{button ,AL('H_EDITWRAP_METHOD_EXSCRIPT',1)} See example
```

This language element has not yet been defined. You can check for a newer Help file on the Lotus web site.

Syntax

```
[objectreference].EditWrap(Intbool On=True)
```

Parameters

On

The data type for this parameter is Integer. The legal values for this parameter are -1 and 0 but you may use the LotusScript constants True (-1) and False (0) instead of the integer values. This is an optional parameter. The default value for this parameter is True. This is an optional parameter.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Word Pro: GetPageNumFromDisplayablePageNum method

{button ,AL(^H_DOCWINDOW_CLASS',0)} [See list of classes](#)

Allows the user to determine the total number of pages in a document including external, OLE, hidden, and parent divisions.

Syntax

[objectreference].GetPageNumFromDisplayablePageNum(Integer Page)

Parameters

Page

The data type for this parameter is Integer. There is no default value. You must provide a value for this parameter.

Return value

Integer

Usage

Word Pro: GetTabAlignChar method

{button ,AL('H_TABRACK_CLASS',0)} [See list of classes](#)

{button ,AL('H_GETTABALIGNCHAR_METHOD_EXSCRIPT',1)} [See example](#)

This language element has not yet been defined. You can check for a newer Help file on the Lotus web site.

Syntax

[objectreference].GetTabAlignChar(Integer TabIndex)

Parameters

TabIndex

The data type for this parameter is Integer. There is no default value. You must provide a value for this parameter.

Return value

The data type for this method's return value is Integer.

Usage

Word Pro: GetTabLeaderType method

{button ,AL('H_TABRACK_CLASS',0)} [See list of classes](#)

{button ,AL('H_GETTABLEADERTYPE_METHOD_EXSCRIPT',1)} [See example](#)

This language element has not yet been defined. You can check for a newer Help file on the Lotus web site.

Syntax

[objectreference].GetTabLeaderType(Integer TabIndex)

Parameters

TabIndex

The data type for this parameter is Integer. There is no default value. You must provide a value for this parameter.

Return value

The data type for this method's return value is TabLeader.

Usage

Word Pro: GetTabPosition method

{button ,AL('H_TABRACK_CLASS',0)} [See list of classes](#)

{button ,AL('H_GETTABPOSITION_METHOD_EXSCRIPT',1)} [See example](#)

This language element has not yet been defined. You can check for a newer Help file on the Lotus web site.

Syntax

[objectreference].GetTabPosition(Integer TabIndex)

Parameters

TabIndex

The data type for this parameter is Integer. There is no default value. You must provide a value for this parameter.

Return value

The data type for this method's return value is Longtwips.

Usage

Word Pro: GetTabRelativeType method

{button ,AL('H_TABRACK_CLASS',0)} [See list of classes](#)

{button ,AL('H_GETTABRELATIVETYPE_METHOD_EXSCRIPT',1)} [See example](#)

This language element has not yet been defined. You can check for a newer Help file on the Lotus web site.

Syntax

[objectreference].GetTabRelativeType(Integer TabIndex)

Parameters

TabIndex

The data type for this parameter is Integer. There is no default value. You must provide a value for this parameter.

Return value

The data type for this method's return value is TabRelative.

Usage

Word Pro: GetTabType method

{button ,AL('H_TABRACK_CLASS',0)} [See list of classes](#)

{button ,AL('H_GETTABTYPE_METHOD_EXSCRIPT',1)} [See example](#)

This language element has not yet been defined. You can check for a newer Help file on the Lotus web site.

Syntax

[objectreference].GetTabType(Integer TabIndex)

Parameters

TabIndex

The data type for this parameter is Integer. There is no default value. You must provide a value for this parameter.

Return value

The data type for this method's return value is TabType.

Usage

Word Pro: GoToDivision method

{button ,AL('H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL('H_GOTODIVISION_METHOD_EXSCRIPT',1)} [See example](#)

This language element has not yet been defined. You can check for a newer Help file on the Lotus web site.

Syntax

[objectreference].GoToDivision(String Name)

Parameters

Name

The data type for this parameter is String. There is no default value. You must provide a value for this parameter.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Word Pro: ImageProcess method

{button ,AL(^H_GRAPHICOLEBJECT_CLASS;H_GRAPHIC_CLASS;H_OLEOBJECT_CLASS',0)} [See list of classes](#)

{button ,AL(^H_IMAGEPROCESS_METHOD_EXSCRIPT',1)} [See example](#)

Opens the Image Processing dialog box for the OLE object from which the method is called.

Syntax

[objectreference].ImageProcess()

Parameters

none

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

This method only works on OLE objects for which Image Processing dialog box would normally be available. Bitmap and TIFF images can make use of the Image Processing features. Other types of images must first be converted.

Word Pro: InternetFileExists method

{button ,AL(^H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

This language element has not yet been defined. You can check for a newer Help file on the Lotus web site.

Syntax

[objectreference].InternetFileExists(String URL)

Parameters

URL

The data type for this parameter is String. There is no default value. You must provide a value for this parameter.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Word Pro: LinkFrameContents method

{button ,AL(^H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

This language element has not yet been defined. You can check for a newer Help file on the Lotus web site.

Syntax

[objectreference].LinkFrameContents()

Parameters

none

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Word Pro: NewInfoBusLink method

{button ,AL(^H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

This language element has not yet been defined. You can check for a newer Help file on the Lotus web site.

Syntax

[objectreference].NewInfoBusLink(String InfoBusDataMoniker)

Parameters

InfoBusDataMoniker

The data type for this parameter is String. There is no default value. You must provide a value for this parameter.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Word Pro: PasteDivision method

{button ,AL('H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL('H_PASTEDIVISION_METHOD_EXSCRIPT',1)} [See example](#)

Pastes the division in the Clipboard into the active document after the specified division. If you don't specify a division, Word Pro pastes the division after the currently active division. Equivalent to right-clicking a divider tab and choosing Paste Division.

Syntax

[objectreference].PasteDivision([DivisionName])

Parameters

DivisionName

The [internal name](#) of the division after which you want to paste the division from the clipboard. The data type for this parameter is String. There is no default value. This is an optional parameter. If you don't provide a value for this parameter, Word Pro pastes the Clipboard division after the currently active division.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Note If there is not a valid Word Pro division in the Clipboard, this method returns False and has no effect on the document.

Usage

Before calling this method, a valid Word Pro division must be cut or copied to the Clipboard. You can place a division on the Clipboard by calling the CutDivision or CopyDivision method. A user can add a division to the Clipboard by right-clicking a division's divider tab and choosing Cut Division or Paste Division.

Word Pro: RemoveHyperlink method

{button ,AL(^H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL(^H_REMOVEHYPERLINK_METHOD_EXSCRIPT',1)} [See example](#)

Removes the Hyperlink code from the text at the insertion point.

Syntax

[objectreference].RemoveHyperlink()

Parameters

none

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

This method removes only the link code, leaving the text intact. If the text at the insertion point is not marked as a Hyperlink, this method does nothing and returns a zero.

Word Pro: SaveInternetFile method

{button ,AL('H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL('H_SAVEINTERNETFILE_METHOD_EXSCRIPT',1)} [See example](#)

This language element has not yet been defined. You can check for a newer Help file on the Lotus web site.

Syntax

[objectreference].SaveInternetFile(String URL, String Type, String UserID=, String Password=, Intbool Passive=False, String Proxy=, Integer ProxyPort=21)

Parameters

URL

The data type for this parameter is String. There is no default value. You must provide a value for this parameter.

Type

The data type for this parameter is String. There is no default value. You must provide a value for this parameter.

UserID

The data type for this parameter is String. There is no default value for this parameter. This is an optional parameter.

Password

The data type for this parameter is String. There is no default value for this parameter. This is an optional parameter.

Passive

The data type for this parameter is Integer. The legal values for this parameter are -1 and 0 but you may use the LotusScript constants True (-1) and False (0) instead of the integer values. This is an optional parameter. The default value for this parameter is False. This is an optional parameter.

Proxy

The data type for this parameter is String. There is no default value for this parameter. This is an optional parameter.

ProxyPort

The data type for this parameter is Integer. The default value for this parameter is 21. This is an optional parameter.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Word Pro: SavePreviewFile method

{button ,AL(^H_DIVISION_CLASS;H_TEXTDOCUMENT_CLASS',0)} [See list of classes](#)

This language element has not yet been defined. You can check for a newer Help file on the Lotus web site.

Syntax

[objectreference].SavePreviewFile(String filename, Integer pagenumber, Integer width, Integer height)

Parameters

filename

The data type for this parameter is String. There is no default value. You must provide a value for this parameter.

pagenumber

The data type for this parameter is Integer. There is no default value. You must provide a value for this parameter.

width

The data type for this parameter is Integer. There is no default value. You must provide a value for this parameter.

height

The data type for this parameter is Integer. There is no default value. You must provide a value for this parameter.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Word Pro: Search method

{button ,AL(^H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

This language element has not yet been defined. You can check for a newer Help file on the Lotus web site.

Syntax

[objectreference].Search()

Parameters

none

Return value

The data type for this method's return value is Search.

Usage

Word Pro: SelectTWAINSource method

{button ,AL(^H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

This language element has not yet been defined. You can check for a newer Help file on the Lotus web site.

Syntax

[objectreference].SelectTWAINSource()

Parameters

none

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Word Pro: SetButtonPercent method

{button ,AL(^H_STATUSBARBUTTON_CLASS',0)} [See list of classes](#)

This language element has not yet been defined. You can check for a newer Help file on the Lotus web site.

Syntax

[objectreference].SetButtonPercent(Integer Percentage, Long IsHilite)

Parameters

Percentage

The data type for this parameter is Integer. There is no default value. You must provide a value for this parameter.

IsHilite

The data type for this parameter is Long. There is no default value. You must provide a value for this parameter.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Word Pro: SetCursorPosition method

{button ,AL(^H_CLICKHERE_CLASS;H_TEXTMARKER_CLASS;H_TEXT_CLASS',0)} [See list of classes](#)

This language element has not yet been defined. You can check for a newer Help file on the Lotus web site.

Syntax

[objectreference].SetCursorPosition(TabType Tab)

Parameters

Tab

Data type is Variant, which allows the value of this parameter to be one of the constants listed below or its numeric equivalent (in parentheses). There is no default value. You must provide a value for this parameter.

<u>Value</u>	<u>Effect</u>
\$LwpTabTypeCenter (1864)	
\$LwpTabTypeLeft (1863)	
\$LwpTabTypeNumeric (1866)	
\$LwpTabTypeRight (1865)	

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Word Pro: SetMultiCellPaste method

{button ,AL(^H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

This language element has not yet been defined. You can check for a newer Help file on the Lotus web site.

Syntax

[objectreference].SetMultiCellPaste(String MultiCellPaste)

Parameters

MultiCellPaste

The data type for this parameter is String. There is no default value. You must provide a value for this parameter.

Return value

The data type for this method's return value is String.

Usage

Word Pro: SetPaperNameAndUpdateSize method

```
{button ,AL(^H_LAYOUT_CLASS;H_CELLGROUPLAYOUT_CLASS;H_CELLLAYOUT_CLASS;H_CONNECTEDLAY  
OUT_CLASS;H_COLUMNGROUPLAYOUT_CLASS;H_FOOTERLAYOUT_CLASS;H_FRAMELAYOUT_CLASS;H  
_FRAMEGROUPLAYOUT_CLASS;H_DROP_CAPLAYOUT_CLASS;H_GROUPLAYOUT_CLASS;H_HEADERLAY  
OUT_CLASS;H_NOTELAYOUT_CLASS;H_PAGELAYOUT_CLASS;H_ROWGROUPLAYOUT_CLASS;H_ROWLA  
YOUT_CLASS;H_RUBYLAYOUT_CLASS;H_SUPERTABLEGROUPLAYOUT_CLASS;H_SUPERTABLELAYOUT  
_CLASS;H_TABLELAYOUT_CLASS;H_ENDNOTELAYOUT_CLASS;H_FOOTNOTELAYOUT_CLASS;H_TABLE  
HEADINGLAYOUT_CLASS;H_TOCSUPERTABLELAYOUT_CLASS',0)} See list of classes
```

This language element has not yet been defined. You can check for a newer Help file on the Lotus web site.

Syntax

```
[objectreference].SetPaperNameAndUpdateSize(String PaperSizeName)
```

Parameters

PaperSizeName

The data type for this parameter is String. There is no default value. You must provide a value for this parameter.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Word Pro: SetupForCropping method

```
{button ,AL('H_LAYOUT_CLASS;H_CELLGROUPLAYOUT_CLASS;H_CELLLAYOUT_CLASS;H_CONNECTEDLAY  
OUT_CLASS;H_COLUMNGROUPLAYOUT_CLASS;H_FOOTERLAYOUT_CLASS;H_FRAMELAYOUT_CLASS;H  
_FRAMEGROUPLAYOUT_CLASS;H_DROP_CAPLAYOUT_CLASS;H_GROUPLAYOUT_CLASS;H_HEADERLAY  
OUT_CLASS;H_NOTELAYOUT_CLASS;H_PAGELAYOUT_CLASS;H_ROWGROUPLAYOUT_CLASS;H_ROWLA  
YOUT_CLASS;H_RUBYLAYOUT_CLASS;H_SUPERTABLEGROUPLAYOUT_CLASS;H_SUPERTABLELAYOUT  
_CLASS;H_TABLELAYOUT_CLASS;H_ENDNOTELAYOUT_CLASS;H_FOOTNOTELAYOUT_CLASS;H_TABLE  
HEADINGLAYOUT_CLASS;H_TOCSUPERTABLELAYOUT_CLASS;',0)} See list of classes
```

```
{button ,AL('H_SETUPFORCROPPING_METHOD_EXSCRIPT',1)} See example
```

This language element has not yet been defined. You can check for a newer Help file on the Lotus web site.

Syntax

[objectreference].SetupForCropping()

Parameters

none

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Word Pro: SortParagraphs method

{button ,AL(^H_WPAPPLICATION_CLASS;H_CLICKHERE_CLASS;H_TEXTMARKER_CLASS;H_TEXT_CLASS',0)}

[See list of classes](#)

{button ,AL(^H_SORTPARAGRAPHS_METHOD_EXSCRIPT',1)} [See example](#)

Sorts the currently active text object, selection or entire table, based on the settings in the SortOptions object.

Syntax

[objectreference].SortParagraphs()

Parameters

None.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

This method behaves the same way as the Sort function found on the Word Pro Text menu.

Word Pro: SpellAddToUserDict method

{button ,AL(^H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL(^H_SPELLADDTouserDict_METHOD_EXSCRIPT',1)} [See example](#)

Adds the selected word to the user dictionary.

Syntax

[objectreference].SpellAddToUserDict()

Parameters

None.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Word Pro: SpellClearSkippedWords method

{button ,AL(^H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

Removes the Skip flag from words you have instructed Word Pro to skip.

Syntax

[objectreference].SpellClearSkippedWords()

Parameters

None.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Word Pro: SpellMarkSkippedWords method

{button ,AL('H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL('H_SPELLMARKSKIPPEDWORDS_METHOD_EXSCRIPT',1)} [See example](#)

Marks the selected word as skipped.

Note This method also removes the flags for misspelled or double words.

Syntax

[objectreference].SpellMarkSkippedWords()

Parameters

None.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Word Pro: SpellSkipAll method

{button ,AL(`H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL(`H_SPELLSKIPALL_METHOD_EXSCRIPT',1)} [See example](#)

Adds the selected word to Spell Check's skip all list.

Syntax

[objectreference].SpellSkipAll()

Parameters

None.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Word Pro: SpellWord method

{button ,AL(^H_CLICKHERE_CLASS;H_TEXTMARKER_CLASS;H_TEXT_CLASS',0)} [See list of classes](#)

Syntax

[objectreference].SpellWord()

Parameters

Return value

Usage

Word Pro: SplitParagraph method

{button ,AL('H_CLICKHERE_CLASS;H_TEXTMARKER_CLASS;H_TEXT_CLASS',0)} [See list of classes](#)

{button ,AL('H_SPLITPARAGRAPH_METHOD_EXSCRIPT',1)} [See example](#)

Splits the current paragraph into two separate paragraphs.

Syntax

[objectreference].SplitParagraph([PropagateAttributes])

Parameters

PropagateAttributes

Allows you to carry over any local paragraph attributes from the existing paragraph to the new paragraph. Data type is Integer. The legal values for this parameter are -1 and 0 but you may use the LotusScript constants True (-1) and False (0) instead of the integer values. Optional parameter. Default is False which passes on only the paragraph style attributes.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Word Pro: Split method

{button ,AL(^H_BASERABLE_CLASS;H_FOOTNOTETABLE_CLASS;H_PARALLELCOLUMNS_CLASS;H_GLOSSARY_CLASS;H_TABLEHEADING_CLASS;H_TABLE_CLASS',0)} [See list of classes](#)

{button ,AL(^H_SPLIT_METHOD_EXSCRIPT',1)} [See example](#)

Splits a cell into two or more rows or columns.

Syntax

[objectreference].Split(SplitType, [NumRows,] [NumCols])

Parameters

SplitType

This Variant parameter must be one of the string constants below or its numeric equivalent.

<u>Value</u>	<u>Effect</u>
\$LwpTableSplitTypeCell (1904)	Equivalent to choosing Table - Split Cell. Splits the current cell into the number of cells specified by the NumRows and NumCols parameter values.
\$LwpTableSplitTypeTable (1905)	Equivalent to choosing Table - Split Entire Table. Values in the NumRows and NumCols parameters have no effect.

NumRows

An optional Integer parameter that determines how many rows into which a cell will be split.

NumCols

An optional Integer parameter that determines how many columns into which a cell will be split.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Word Pro: SRReplace method

{button ,AL(^H_CLICKHERE_CLASS;H_TEXTMARKER_CLASS;H_TEXT_CLASS',0)} [See list of classes](#)

Syntax

[objectreference].SRReplace([IsTemporary])

Parameters

IsTemporary

Data type is Boolean. Optional parameter. Default is False.

Return value

Usage

Word Pro: StartEditMergeData method

{button ,AL(^H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL(^H_STARTEDITMERGEDATA_METHOD_EXSCRIPT',1)} [See example](#)

Opens the data file for the currently active merge document. The data file remains hidden from the user but available for editing from the merge document. Equivalent to choosing Text - Merge and clicking Edit Data File.

Syntax

[objectreference].StartEditMergeData()

Parameters

None.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Word Pro: StartEnvelopeDiv method

{button ,AL('H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL('H_STARTENVELOPEDIV_METHOD_EXSCRIPT',1)} [See example](#)

Creates an envelope division in the currently active document.

Syntax

[objectreference].StartEnvelopeDiv()

Parameters

None.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

This method performs the following tasks in the course of creating the envelope division:

- Creates frames for the return address and delivery address.
- Checks for selected text in the active document.
- If text is selected, Word Pro copies that text into the delivery address frame.
- If no text is selected, Word Pro looks at the beginning of the document for 3 to 6 lines which have fewer than 60 characters each.
- If Word Pro finds lines which meet these criteria, Word Pro assumes those lines are the delivery address and copies them into the delivery address frame.

Word Pro: StartFieldInsert method

{button ,AL('H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL('H_STARTFIELDINSERT_METHOD_EXSCRIPT',1)} [See example](#)

Opens the Merge bar so the user can insert fields for merging. This is the same Merge bar that appears during Step 2 of the automated merge process.

Syntax

[objectreference].StartFieldInsert()

Parameters

None.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Word Pro: Start method

```
{button ,AL('H_BASECONTAINER_CLASS;H_CELLCONTAINER_CLASS;H_FRAMECONTAINER_CLASS;H_NOTE  
CONTAINER_CLASS;H_RUBYCONTAINER_CLASS;H_DROPCAPCONTAINER_CLASS;H_PAGECONTAINER_  
CLASS;H_SUBPAGECONTAINER_CLASS;H_SUPERPAGECONTAINER_CLASS;H_ROWCONTAINER_CLASS;  
H_SUPERTABLECONTAINER_CLASS;H_TABLECONTAINER_CLASS;H_PARALLELCOLSCONTAINER_CLASS  
;H_TABLEONLYCONT_CLASS',0)} See list of classes
```

```
{button ,AL('H_START_METHOD_EXSCRIPT',1)} See example
```

In any container, this method moves the insertion point from its current position to the beginning of the document.

Syntax

[objectreference].Start(OfWhat)

Parameters

OfWhat

Data type is Variant. Value must be \$LwpDocumentObjectTypeDocument, or its numeric equivalent (216).

Return value

The return value for this method will always be -1 or 0. When testing the return value, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Word Pro: StoreInternetFile method

{button ,AL('H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

Uses FTP to transfer a file from your local machine to an Internet site. You can transfer any file including Word Pro documents and HTML files.

Syntax

[objectreference].StoreInternetFile(LocalFile, URL, [UserID], [Password], [Passive], [Proxy], [ProxyPort])

Parameters

LocalFile

A String expression specifying the name and path of the file which you want to send to an Internet site.

URL

A String expression specifying the URL for the FTP server that will receive the file. This string value must include the name of the directory in which you want to store the file.

UserID

A String expression representing the name of the user who has an account with the FTP server.

Password

A String expression representing the password for the user named in UserID.

Passive

Set this value to True when you want to initiate the file transfer. Set it to False to allow the server to respond to your request when it is ready. Some FTP servers do not support this feature. The value of this parameter is usually False. Data type is Integer but the value is always 0 (False) or -1 (True). You can use the LotusScript constants of True and False.

Proxy

A String expression specifying the DNS (for example, *screen.companyname.com*) or IP address (for example, *123.456.78.912*). Do not include "http://" in front of the the proxy value.

ProxyPort

An Integer which specifies the port number for the proxy server. The value of this parameter is usually 21 for FTP, but you should check with your Internet service provider for your settings.

Return value

Integer.

Usage

This method will not work unless your machine is configured for Internet access. A standard Internet access configuration includes a WINSOCK compliant DLL.

Word Pro: StrikeThru method

{button ,AL('H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL('H_STRIKETHRU_METHOD_EXSCRIPT',1)} [See example](#)

Sets the strikethrough attribute for selected text, or all following text, if no text is selected. Acts as a toggle, turning the attribute off if it is on and on if it is off. Equivalent to choosing Text - Attributes - Other, and then choosing "Strikethrough" from the Attributes box.

Syntax

[objectreference].StrikeThru()

Parameters

None.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Word Pro: SubScript method

{button ,AL('H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL('H_SUBSCRIPT_METHOD_EXSCRIPT',1)} [See example](#)

Sets the subscript attribute for selected text, or all following text, if no text is selected. Acts as a toggle, turning the attribute off if it is on and on if it is off. Equivalent to choosing Text - Attributes - Other, and then choosing "Subscript" from the Attributes box.

Syntax

[objectreference].SubScript()

Parameters

None.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Word Pro: Sum method

{button ,AL('H_TABLE_CLASS',0)} [See list of classes](#)

{button ,AL('H_SUM_METHOD_EXSCRIPT',1)} [See example](#)

Adds the contents of a range of cells within a table.

Syntax

[objectreference].Sum(TableSumScope)

Parameters

TableSumScope

Adds the contents in a range of columns or rows. The value of this Variant parameter must be one of the strings below or its code equivalent.

<u>Value</u>	<u>Effect</u>
\$LwpTableSumScopeColumn (1907)	Adds the contents of a range of columns within a table.
\$LwpTableSumScopeRow (1906)	Adds the contents of a range of rows within a table.

Return value

This method returns an Integer value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Equivalent to inserting a SmartSum formula.

Word Pro: SuperScript method

{button ,AL('H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL('H_SUPERSCRIPT_METHOD_EXSCRIPT',1)} [See example](#)

Sets the Superscript attribute for selected text, or all following text, if no text is selected. Acts as a toggle, turning the attribute off if it is on and on if it is off. Equivalent to choosing Text - Attributes - Other, and then choosing "Superscript" from the Attributes box.

Syntax

[objectreference].SuperScript()

Parameters

None.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Word Pro: TeamMail method

{button ,AL(`H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL(`H_TEAMMAIL_METHOD_EXSCRIPT',1)} [See example](#)

Opens and displays the TeamMail dialog box. Equivalent to choosing File - TeamMail.

Syntax

[objectreference].TeamMail()

Parameters

None.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Word Pro: TileWindowHorz method

{button ,AL(^H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL(^H_TILEWINDOWHORZ_METHOD_EXSCRIPT',1)} [See example](#)

Resizes and arranges all active document windows so they appear side by side in the Word Pro application window. Equivalent to choosing Window - Tile Left-Right.

Syntax

[objectreference].TileWindowHorz()

Parameters

None

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

If more than two windows are open, Word Pro may arrange the windows in rows so the first windows are tiled left to right in the top row, and the remaining windows are tiled in more rows beneath.

Word Pro: TileWindowVert method

{button ,AL('H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL('H_TILEWINDOWVERT_METHOD_EXSCRIPT',1)} [See example](#)

Resizes and arranges all active document windows so they appear one above the other in the Word Pro application window. Equivalent to choosing Window - Tile Top-Bottom.

Syntax

[objectreference].TileWindowVert()

Parameters

None.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

If more than two windows are open, Word Pro may arrange the windows in columns so the first windows are tiled top to bottom on the left, and the remaining windows are tiled in columns to the right.

Word Pro: Tile method

{button ,AL('H_APPLICATIONWINDOW_CLASS',0)} [See list of classes](#)

{button ,AL('H_TILE_METHOD_EXSCRIPT',1)} [See example](#)

Tiles the document windows within the Word Pro application window.

Syntax

[objectreference].Tile()

Parameters

None.

Return value

The return value for this method will always be -1 or 0. When testing the return value, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Uses the Tile command in the Windows application.

Word Pro: TimedSave method

{button ,AL('H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL('H_TIMEDSAVE_METHOD_EXSCRIPT',1)} [See example](#)

Performs a timed save of all open documents.

Syntax

[objectreference].TimedSave()

Parameters

None.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Timed save is the procedure which Word Pro performs automatically if you choose File - User Setup - Word Pro Preferences, and select "Automatically time save every." This method performs a timed save, regardless of whether a document's automatic save option is selected.

In a timed save, Word Pro creates a .~TS file for the document. The .~TS file is a copy of the document, located in the same folder as the original document.

Word Pro updates the .~TS file each time it autosaves the document. When you save a document, Word Pro saves the .~TS file to the original document and deletes the .~TS file. When you close a document without saving it, Word Pro deletes the .~TS file, without saving it to the original document.

If you exit Word Pro abnormally, the .~TS files are not deleted. The next time you start Word Pro or open the original document, Word Pro prompts you about opening the .~TS file. If you do not open the .~TS file, Word Pro deletes it.

Word Pro: ToggleCleanScreen method

{button ,AL(^H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL(^H_TOGGLECLEANSCREEN_METHOD_EXSCRIPT',1)} [See example](#)

Syntax

[objectreference].ToggleCleanScreen()

Parameters

None.

Return value

None.

Usage

Word Pro: ToggleIconBar method

{button ,AL(`H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL(`H_TOGGLEICONBAR_METHOD_EXSCRIPT',1)} [See example](#)

Shows or hides the currently active set of SmartIcons. Equivalent to choosing View - Show/Hide - SmartIcons.

Syntax

[objectreference].ToggleIconBar()

Parameters

None.

Return value

None.

Usage

Word Pro: Type method

{button ,AL('H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL('H_TYPE_METHOD_EXSCRIPT',1)} [See example](#)

Performs the specified keystrokes in a document. Available keystrokes include insertion point movement and function keys as outlined below in Parameters.

Note When you use multiple Type methods in a single script, you must also use the BeginChange and EndChange methods.

Syntax

[objectreference].Type(Keystroke)

Parameters

Keystroke

A String expression which represents the characters you want Word Pro to type in the document. To type a double quote mark, you must use two double quotes, so that LotusScript can distinguish between the double quote you want to type and those which surround the rest of the string.

For example, this statement:

```
.Type("a double quote "" in a document")
```

would type this in your document:

```
a double quote " in a document
```

You can also include an insertion point movement or function key. To type a key, surround its name with square braces. The following key names can be used:

[Home] - Home key

[End] - End key

[PgUp] - Page Up key

[PgDn] - Page Down key

[Ins] - Insert key

[Del] - Delete key

[Backspace] - Backspace key

[Enter] - Enter or Return key

[Tab] - Tab key

[ESC] - Escape key

[Up] - Up Arrow key

[Down] - Down Arrow key

[Left] - Left Arrow key

[Right] - Right Arrow key

[F1] - [F12] - Function keys F1 through F12

You can also add the standard modifiers (CTRL, SHIFT and ALT) to these keys by appending "CTRL", "SHIFT" or "ALT," in any combination, to the front of the key name. For example, "CTRLDown" is the same as holding down the CTRL key, while pressing the Down Arrow key; "CTRLSHIFTDown" is the same as holding down the CTRL and SHIFT keys, while pressing the Down Arrow key.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

If you use multiple Type methods in succession within your script, be sure to surround the sequence of Type methods with the BeginChange and EndChange methods. For example:

```
.BeginChange  
.Type "12"  
.Type "34"  
.Type "56"
```

.EndChange

Failure to use the BeginChange and EndChange methods may yield unpredictable results, particularly within table cells.

Word Pro: Underline method

{button ,AL(`H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL(`H_UNDERLINE_METHOD_EXSCRIPT',1)} [See example](#)

Sets the underline attribute for selected text, or all following text, if no text is selected. Acts as a toggle, turning the attribute off if it is on and on if it is off. Equivalent to choosing Text - Attributes - Other, and then choosing "Underline" from the Attributes box.

Syntax

[objectreference].Underline()

Parameters

None.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Word Pro: Undo method

{button ,AL(^H_WPAPPLICATION_CLASS;H_GRAPHICOLEBJECT_CLASS;H_GRAPHIC_CLASS;H_OLEOBJECT_CLASS;H_TEXTDOCUMENT_CLASS',0)} [See list of classes](#)

{button ,AL(^H_UNDO_METHOD_EXSCRIPT',1)} [See example](#)

Reverses (undoes) the previous editing function. Equivalent to choosing Edit - Undo.

Syntax

When called from WPAApplication or a Graphic object:

```
[Objectreference].Undo()
```

When called from a TextDocument object:

```
[Objectreference].Undo([Count])
```

Parameters

Count

Specifies the number of recent edits to undo. Optional parameter. Only available when this method is called from a TextDocument object. Data type is Integer. Default is 1.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Word Pro: UnlinkFrameContents method

{button ,AL(`H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL(`H_UNLINKFRAMECONTENTS_METHOD_EXSCRIPT',1)} [See example](#)

This language element has not yet been defined. You can check for a newer Help file on the Lotus web site.

Syntax

[objectreference].UnlinkFrameContents()

Parameters

none

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Word Pro: UnregisterWPDataSet method

```
{button ,AL(^H_WPAPPLICATION_CLASS;H_CHARACTERSTYLE_CLASS;H_DIVISION_CLASS;H_LAYOUT_CLASSES;H_CELLGROUP_LAYOUT_CLASS;H_CELL_LAYOUT_CLASS;H_CONNECTED_LAYOUT_CLASS;H_COLUMN_GROUP_LAYOUT_CLASS;H_FOOTER_LAYOUT_CLASS;H_FRAME_LAYOUT_CLASS;H_FRAME_GROUP_LAYOUT_CLASS;H_DROP_CAP_LAYOUT_CLASS;H_GROUP_LAYOUT_CLASS;H_HEADER_LAYOUT_CLASS;H_NOTE_LAYOUT_CLASS;H_PAGE_LAYOUT_CLASS;H_ROW_GROUP_LAYOUT_CLASS;H_ROW_LAYOUT_CLASS;H_RUBY_LAYOUT_CLASS;H_SUPER_TABLE_GROUP_LAYOUT_CLASS;H_SUPER_TABLE_LAYOUT_CLASS;H_TABLE_LAYOUT_CLASS;H_END_NOTE_LAYOUT_CLASS;H_FOOT_NOTE_LAYOUT_CLASS;H_TABLE_HEADING_LAYOUT_CLASS;H_TOCSUPER_TABLE_LAYOUT_CLASS;H_MARKER_CLASS;H_POWERFIELD_CLASS;H_CLICKHERE_CLASS;H_TEXTMARKER_CLASS;H_RUBYMARKER_CLASS;H_TABLEMARKER_CLASS;H_PARAGRAPHSTYLE_CLASS;H_TEXT_CLASS;H_TEXTDOCUMENT_CLASS',0)} See list of classes
```

```
{button ,AL(^H_UNREGISTERWPDATASET_METHOD_EXSCRIPT',1)} See example
```

Deletes a WPDataSet from the object from which you call this method.

Syntax

```
[objectreference].UnregisterWPDataSet(Group Name)
```

Parameters

GroupName

A String expression representing the name given to the WPDataSet.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

WPDataSet objects are useful tools that store data with a document. When you close a document that has one or more data sets attached to it, Word Pro saves the data set(s) with the document. Any time the document is open, you have access to the data sets created for that document.

When you register or unregister a WPDataSet on a Text object, that WPDataSet is assigned to the currently active paragraph.

This method deletes a WPDataSet object. The deleted data set cannot be restored, once it has been unregistered.

Word Pro: UpdateAllMenus method

{button ,AL('H_APPLICATIONWINDOW_CLASS',0)} [See list of classes](#)

This language element has not yet been defined. You can check for a newer Help file on the Lotus web site.

Syntax

[objectreference].UpdateAllMenus()

Parameters

none

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Word Pro: UpdateFootersText method

{button ,AL('H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL('H_UPDATEFOOTERSTEXT_METHOD_EXSCRIPT',1)} [See example](#)

Updates all the footers in a document to match the footer content in the currently active division.

Syntax

[objectreference].UpdateFootersText()

Parameters

None.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Word Pro: UpdateHeadersText method

{button ,AL('H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL('H_UPDATEHEADERSTEXT_METHOD_EXSCRIPT',1)} [See example](#)

Updates all the headers in a document to match the header content in the currently active division.

Syntax

[objectreference].UpdateHeadersText()

Parameters

None.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Word Pro: UpdateIndexSection method

{button ,AL(^H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

Updates the specified index section to reflect any changes made to the associated division or document.

Syntax

[objectreference].UpdateIndexSection(DivisionName, SectionName)

Parameters

DivisionName

Data type is String.

SectionName

Data type is String.

Return value

Usage

Word Pro: UpdateLink method

{button ,AL('H_GRAPHIC_CLASS;H_OLEOBJECT_CLASS',0)} [See list of classes](#)

Note This method is not implemented for the OLEObject class within OS/2.

Syntax

[objectreference].UpdateLink(ShowErrorMessage)

Parameters

ShowErrorMessage

Data type is Integer. The legal values for this parameter are -1 and 0 but you may use the LotusScript constants True (-1) and False (0).

Return value

The return value for this method will always be -1 or 0. When testing the return value, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Word Pro: UpdateOle method

{button ,AL(^H_WPAPPLICATION_CLASS;H_TEXTDOCUMENT_CLASS',0)} [See list of classes](#)

When the Word Pro document is OLE embedded in another container, this method updates the Word Pro document which is embedded.

Note OLE is not supported under OS/2.

Syntax

[objectreference].UpdateOle()

Parameters

None

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Word Pro: UpdatePageSizeChange method

{button ,AL('H_PRINTMANAGER_CLASS',0)} [See list of classes](#)

{button ,AL('H_UPDATEPAGESIZECHANGE_METHOD_EXSCRIPT',1)} [See example](#)

Updates any changes to the size of the paper used to print the document.

Syntax

[objectreference].UpdatePageSizeChange()

Parameters

Return value

Usage

Word Pro: UpdatePowerFields method

{button ,AL('H_DIVISION_CLASS;H_TEXTDOCUMENT_CLASS',0)} [See list of classes](#)

{button ,AL('H_UPDATEPOWERFIELDS_METHOD_EXSCRIPT',1)} [See example](#)

Updates power fields in the division from which you call this method.

Syntax

[objectreference].UpdatePowerFields([Reset] [, JustDates])

Parameters

Reset

Data type is Integer. The legal values for this parameter are -1 and 0 but you may use the LotusScript constants True (-1) and False (0) instead of the integer values. Optional parameter. Default is False (0).

JustDates

Data type is Integer. The legal values for this parameter are -1 and 0 but you may use the LotusScript constants True (-1) and False (0) instead of the integer values. Optional parameter. Default is False (0).

Return value

Usage

Word Pro: UpdatePrinterChanges method

{button ,AL(^H_PRINTMANAGER_CLASS',0)} [See list of classes](#)

Updates any changes to power fields in a document, power fields used to create a table of contents, and power fields used to create indexes. Prints the results at the location of the power field.

Syntax

[objectreference].UpdatePrinterChanges()

Parameters

Return value

Usage

Word Pro: UpdateSelectedFields method

{button ,AL('H_DOCINFO_CLASS',0)} [See list of classes](#)

{button ,AL('H_UPDATESELECTEDFIELDS_METHOD_EXSCRIPT',1)} [See example](#)

Updates selected DocInfo fields in a document.

Syntax

[objectreference].UpdateSelectedFields([FieldUpdateSelect])

Parameters

FieldUpdateSelect

Data type is Variant. Default is \$LwpFieldUpdateWordsPagesFilesize. The value of this parameter must be one of the strings below or its numeric equivalent (in parentheses):

<u>Value</u>	<u>Effect</u>
\$LwpFieldUpdateChars (257)	Changes and displays within a DocInfo field the current number of characters in a document.
\$LwpFieldUpdatePages (256)	Changes and displays within a DocInfo field the current number of pages in a document.
\$LwpFieldUpdateSize (258)	Changes and displays within a DocInfo field the current kilobyte size of a document.
\$LwpFieldUpdateWords (255)	Changes and displays within a DocInfo field the current number of words in a document.
\$LwpFieldUpdateWordsPagesFilesize (254)	Changes and displays within a DocInfo field the current number of pages, the number of words, and the file size of a document.

Return value

The return value for this method will always be -1 or 0. When testing the return value, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

This method updates the internal values of the specified DocInfo fields. To make the refreshed values appear in your document, update the display of the active document window. For example, to update the DocInfo fields that display the file number of words, you can use the following code:

```
.ActiveDocument.DocInfo.UpdateSelectedFields $LwpFieldUpdateWords  
.ActiveDocWindow.Update
```

Word Pro: UpdateTOC method

{button ,AL(^H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

Updates any changes to a table of contents. Equivalent to choosing Create - Other Document Part - Table of Contents, and clicking Update to display the Update Table of Contents dialog box.

Syntax

[objectreference].UpdateTOC(TOCName)

Parameters

TOCName

Data type is String.

Return value

Usage

Word Pro: Update method

```
{button ,AL(^H_WINDOW_CLASS;H_APPLICATIONWINDOW_CLASS;H_STATUSBAR_CLASS;H_DOCWINDOW_CLASS;H_CHARACTERSTYLE_CLASS;H_POWERFIELD_CLASS;H_LAYOUT_CLASS;H_CELLGROUPLAYOUT_CLASS;H_CELLLAYOUT_CLASS;H_CONNECTEDLAYOUT_CLASS;H_COLUMNGROUPLAYOUT_CLASS;H_FOOTERLAYOUT_CLASS;H_FRAMELAYOUT_CLASS;H_FRAMEGROUPLAYOUT_CLASS;H_DROPPLAYOUT_CLASS;H_GROUPLAYOUT_CLASS;H_HEADERLAYOUT_CLASS;H_NOTELAYOUT_CLASS;H_PAGELAYOUT_CLASS;H_ROWGROUPLAYOUT_CLASS;H_ROWLAYOUT_CLASS;H_RUBYLAYOUT_CLASS;H_SUPER TABLEGROUPLAYOUT_CLASS;H_SUPERTABLELAYOUT_CLASS;H_TABLELAYOUT_CLASS;H_ENDNOTELAYOUT_CLASS;H_FOOTNOTELAYOUT_CLASS;H_TABLEHEADINGLAYOUT_CLASS;H_TOCSUPERTABLELAYOUT_CLASS;H_PARAGRAPHSTYLE_CLASS',0)} See list of classes
```

[CharacterStyle]

Takes the referenced character style and propagates that character style to all divisions. Creates the character style where one does not exist and updates existing character styles.

[Layout]

Takes the referenced layout style and propagates that layout style to all the divisions. When this method is called, it updates an existing layout style or creates a new layout style if one doesn't already exist.

[ParagraphStyle]

[PowerField]

[SectionTabs]

Causes section tabs to verify that all displayed information is correct, and then causes a repaint of the document window.

[Window]

[ApplicationWindow]

Forces a repaint of the application window.

Syntax

```
[objectreference].Update()
```

Parameters

None

Return value

[Layout]

No return value.

[ApplicationWindow]

No return value.

[SectionTabs]

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

This method is not valid for StatusBar.

Word Pro: UpperCase method

{button ,AL('H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL('H_UPPERCASE_METHOD_EXSCRIPT',1)} [See example](#)

Sets the Upper Case attribute for selected text, or all following text, if no text is selected. It acts as a toggle, turning the attribute off if it is on and on if it is off. Equivalent to choosing Text - Attributes - Other, and then choosing "Upper Case" from the Attributes box.

Syntax

[objectreference].UpperCase()

Parameters

None

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Word Pro: UseLSX method

{button ,AL(^H_APPLICATION_CLASS;H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

This language element has not yet been defined. You can check for a newer Help file on the Lotus web site.

Syntax

[objectreference].UseLSX(String LSXName)

Parameters

LSXName

The data type for this parameter is String. There is no default value. You must provide a value for this parameter.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Word Pro: ValidateValue method

{button ,AL(^H_CELLLAYOUT_CLASS;H_CONNECTEDLAYOUT_CLASS',0)} [See list of classes](#)

Used to notify the Word Pro cell engine that the content of a cell has been modified, and that the content of the cell should now be evaluated by the cell engine.

Syntax

[objectreference].ValidateValue()

Parameters**Return value**

This method always returns -1.

Usage

This method does not evaluate or validate any value contained within a table cell.

Word Pro: WMCommand method

{button ,AL('H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL('H_WMCOMMAND_METHOD_EXSCRIPT',1)} [See example](#)

Issues the standard Word Pro menu command specified in the CommandID parameter. The complete list of available menu commands and their corresponding CommandID values is included below.

Note To use these menu commands, you must have the WPBITMSK.LSS file in the Word Pro directory.

Syntax

[objectreference].WMCommand(CommandID)

Parameters

CommandID

A numeric expression of type Integer which specifies the menu command you want to issue. The values in the WPBITMSK.LSS file are expressed as hexadecimal numbers.

Command	ID	Command	ID
LwpMenuFilemenu	100	LwpMenuMIFielddoauto	226
LwpMenuMfNew	101	LwpMenuMIFieldremove	227
LwpMenuMfOpen	102	LwpMenuMIExechand	574
LwpMenuMfSave	103	LwpMenuMIIndexopts	228
LwpMenuMfSaveas	104	LwpMenuHelpmenu	550
LwpMenuMfRevert	105	LwpMenuMhldxhelp	551
LwpMenuMfAppendtext	106	LwpMenuMhAbout	552
LwpMenuMfImport	107	LwpMenuMhMacrohelp	553
LwpMenuMfFilemanage	108	LwpMenuMhCompatible	555
LwpMenuMfDocdesc	109	LwpMenuMhUsingHelp	830
LwpMenuMfCreatedatafile	110	LwpMenuMhKeyboard	831
LwpMenuMfPrint	111	LwpMenuMhHowDol	832
LwpMenuMfChgprinter	112	LwpMenuMhDoHelp	833
LwpMenuMfExit	113	LwpMenuMhForUpgraders	834
LwpMenuMfPrintreset	114	LwpMenuMhTutorial	837
LwpMenuMfDde	115	LwpMenuMhSearch	879
LwpMenuMfMergeaction	116	LwpMenuMsControl	601
LwpMenuMfJustprint	117	LwpMenuMsFileman	602
LwpMenuMfStandalonep	327	LwpMenuMtTbllayout	626
LwpMenuMfAllfidsrename	118	LwpMenuMtInscolorw	627
LwpMenuMfPrintwrongpaper	119	LwpMenuMtDelcolorw	628
LwpMenuMfPrintcancel	120	LwpMenuMtEditfrmla	629
LwpMenuMfDummymrgsel	121	LwpMenuMtTbladd	630
LwpMenuMfDummymrgext	122	LwpMenuMtHeading	631
LwpMenuMfAsciiopts	123	LwpMenuMtTableinfobox	633
LwpMenuMfCanprint	124	LwpMenuMtSizecolorw	634
LwpMenuMfFldrename	125	LwpMenuMtConnect	635
LwpMenuMfPrintopt	126	LwpMenuMtSavefrmla	637
LwpMenuMfSaveasnw	127	LwpMenuMtQuickaddrow	643
LwpMenuMfImportnw	128	LwpMenuMtQuickaddcol	644
LwpMenuMfDocdescnw	129	LwpMenuMtDelcolumn	645
LwpMenuMfMasternw	130	LwpMenuMtDelrow	646
LwpMenuMfNwimport	131	LwpMenuMtInsertrow	647

LwpMenuMfMergeviewprint	132	LwpMenuMtInsertcol	648
LwpMenuMfLastopen1	133	LwpMenuMtInsertaccel	649
LwpMenuMfLastopen2	134	LwpMenuMtDeleteaccel	650
LwpMenuMfLastopen3	135	LwpMenuMtDeltable	651
LwpMenuMfLastopen4	136	LwpMenuMtSelectcolumn	652
LwpMenuMfLastopen5	137	LwpMenuMtSelectrow	653
LwpMenuMfOpennostyle	138	LwpMenuMtSelecttable	654
LwpMenuMfMergelabels	139	LwpMenuMtChgiconsiz	655
LwpMenuMfOtherflds	140	LwpMenuMtDisconnect	656
LwpMenuMfImportpicture	141	LwpMenuMtSplitcells	657
LwpMenuMfExport	142	LwpMenuMtSplittable	658
LwpMenuMfAttribnw	143	LwpMenuMtPcolinscolrow	659
LwpMenuMfCanmergeprint	146	LwpMenuMtPcolinsertrow	660
LwpMenuMfImporttext	152	LwpMenuMtPcolinsertcol	661
LwpMenuMfMergeprintopts	565	LwpMenuMtPcoldeltable	662
LwpMenuMfClose	144	LwpMenuMtPcoldelcolumn	663
LwpMenuMfCloseall	731	LwpMenuMtPcoldelrow	664
LwpMenuMfCloseallreplacelast	732	LwpMenuMtPcolconnect	665
LwpMenuMfPassword	145	LwpMenuMtPcoldisconnect	666
LwpMenuMfPassword2	290	LwpMenuMtPcolsplitcells	667
LwpMenuMfMasterdocument	736	LwpMenuMtPcolselectcolumn	668
LwpMenuMfClosefile	147	LwpMenuMtPcolselectrow	669
LwpMenuMfLock	148	LwpMenuMtPcolselecttable	670
LwpMenuMfNextmdiwindow	727	LwpMenuMtPcoldelcolrow	671
LwpMenuMfMail	229	LwpMenuMtPcolheading	672
LwpMenuMfMailNew	737	LwpMenuMtCellinfolbox	673
LwpMenuMfMailread	230	LwpMenuMtShowtablelineinfolbox	674
LwpMenuMfQuickopen	231	LwpMenuMtShowtablesizeinfolbox	675
LwpMenuMfFilesep	235	LwpMenuMtCreatetablewithgrid	676
LwpMenuMfSavecopyas	292	LwpMenuMtShowpcollineinfolbox	677
LwpMenuMfUpdateobject	293	LwpMenuMtShowpcolsizeinfolbox	678
LwpMenuMfEditorgreet	733	LwpMenuMtSplitpcol	679
LwpMenuMfNewdivision	734	LwpMenuMtPcolselectentiretable	680
LwpMenuMfOpendivision	735	LwpMenuMtSelectentiretable	681
LwpMenuMfSaContinueRoute	738	LwpMenuMtSelectentirecellrange	682
LwpMenuMfSaRoute	739	LwpMenuMtSelectentirepcolcellrange	683
LwpMenuMfSeldatafile	780	LwpMenuMtTabletopalign	684
LwpMenuMfMrgdelimit	780	LwpMenuMtTablecenteralign	685
LwpMenuMfMrgletter	781	LwpMenuMtTablebottomalign	686
LwpMenuMfMrgenvelope	786	LwpMenuMtConnectrow	687
LwpMenuMfMrggotorecord	782	LwpMenuMtPcolconnectrow	688
LwpMenuMfCreatedesc	783	LwpMenuMtGotonextblock	689
LwpMenuMfMrgsort	784	LwpMenuMtCellinfolboxmouse	690
LwpMenuMfCreatemerge	785	LwpMenuMtTableinfolboxmouse	691
LwpMenuMfEditdatafile	787	LwpMenuMxRunrexx	699

LwpMenuMfUsecurasmrgdoc	788	LwpMenuMxRunontime	700
LwpMenuMfOpenandeditdatafile	789	LwpMenuMxHourglass	720
LwpMenuMfOpenpreview	790	LwpMenuMxMacpause	721
LwpMenuMfNewpreview	791	LwpMenuMxMacresume	722
LwpMenuMxMergemacro	792	LwpMenuMxTimedelay	723
LwpMenuMfMergeprintall	793	LwpMenuMxTestforcancel	724
LwpMenuMfMergeopen	794	LwpMenuMxMacmessages	725
LwpMenuMIsRunscript	795	LwpMenuMxPreanswermsgbox	726
LwpMenuMIsCreatescript	863	LwpMenuMxMaccancel	728
LwpMenuMIsCreatedialog	864	LwpMenuMxMaccontinue	729
LwpMenuMIsCompile	869	LwpMenuMxMacrovariables	730
LwpMenuMIsAutoapp	873	LwpMenuMxRunautonew	798
LwpMenuMIsIdeStepIn	960	LwpMenuMxRunautoopen	799
LwpMenuMIsIdeStepOver	961	LwpMenuOutlinemenu	750
LwpMenuMIsIdeStepExit	962	LwpMenuMolPromote	751
LwpMenuMIsIdeContinue	963	LwpMenuMolDemote	752
LwpMenuMIsIdeStop	964	LwpMenuMolMoveup	753
LwpMenuMIsIdeSetBp	965	LwpMenuMolMovedown	754
LwpMenuMIsIdeDisBp	966	LwpMenuMolExpand	755
LwpMenuMIsIdeClearAll	967	LwpMenuMolContract	756
LwpMenuMIsIdeDisAll	968	LwpMenuMolShowlevels	757
LwpMenuMIsIdeBreakpoints	969	LwpMenuMolExpandsingle	758
LwpMenuMIsIdeBrowser	970	LwpMenuMolContractsingle	759
LwpMenuMIsIdeOutput	971	LwpMenuMolUseoutlinestyle	760
LwpMenuMIsIdeVariables	972	LwpMenuMolSetshowlevel1	761
LwpMenuMIsIdeCheckScript	973	LwpMenuMolSetshowlevel2	762
LwpMenuMIsIdeNewSub	974	LwpMenuMolSetshowlevel3	763
LwpMenuMIsIdeNewFun	975	LwpMenuMolSetshowlevel4	764
LwpMenuMIsIdeDebug	976	LwpMenuMolSetshowlevel5	765
LwpMenuMIsIdeRedo	977	LwpMenuMolSetshowlevel6	766
LwpMenuMIsShowide	978	LwpMenuMolSetshowlevel7	767
LwpMenuMIsIdeImport	979	LwpMenuMolSetshowlevel8	768
LwpMenuMIsIdeExport	980	LwpMenuMolSetshowlevel9	769
LwpMenuMIsIdeScriptPref	981	LwpMenuMolSetshowlevel10	770
LwpMenuMIsIdeCloseWin	982	LwpMenuMolOutstylesequence	749
LwpMenuMIsIdeToggle	983	LwpMenuMtTabkey	800
LwpMenuMIsIdePrevSub	984	LwpMenuMeDefpaths nw	804
LwpMenuMIsIdeNextSub	985	LwpMenuMvHidepicts	805
LwpMenuMIsIdeTogglebp	986	LwpMenuMvHidemarks	806
LwpMenuMIsIdeCodeWindow	987	LwpMenuMvHidetrack	807
LwpMenuMIsIdeCheckAll	988	LwpMenuMvColguide	808
LwpMenuMIsIdeSaveLso	989	LwpMenuMvShowcleanscreen	840
LwpMenuMIsIdeShowicon	992	LwpMenuMvFixediconbar	842
LwpMenuMIsIdeRuncursub	993	LwpMenuMvViewlevel	843
LwpMenuMIsIdeHelpLs	994	LwpMenuMtbChange	810

LwpMenuMlsideHelpWp	995	LwpMenuMtbTabbox	811
LwpMenuMlsideHelpSe	996	LwpMenuMvVertruler	814
LwpMenuMlsideHelpAbout	997	LwpMenuMvShownotes	816
LwpMenuMlsideInsertLso	998	LwpMenuMvShowtabsandreturns	818
LwpMenuMIRenumnotes	149	LwpMenuMvOut2lay	820
LwpMenuEditmenu	150	LwpMenuMfImportdraw	821
LwpMenuMeUndo	151	LwpMenuMIMacroendrecord	822
LwpMenuMeCut	153	LwpMenuMvLay2out	823
LwpMenuMeCopy	154	LwpMenuMvSetpreferences	824
LwpMenuMePaste	155	LwpMenuMrCreatelastframe	825
LwpMenuMeUndoredo	156	LwpMenuMIQuickendrecord	826
LwpMenuMePastelinks	157	LwpMenuMrUngroupframes	827
LwpMenuMeInsvariable	158	LwpMenuMIREvmarkingoff	828
LwpMenuMeFootnotes	160	LwpMenuMIFieldhide	829
LwpMenuMeNotes	161	LwpMenuMcExtracton	838
LwpMenuMeNotes20	282	LwpMenuMdivDivisionproperties	844
LwpMenuMeDefaults	162	LwpMenuMdivNewdivision	845
LwpMenuMeTypeover	163	LwpMenuMdivCombinedivisions	846
LwpMenuMeNbDefaults	164	LwpMenuMdivDeletedivision	847
LwpMenuMeInsertdate	165	LwpMenuMdivExpanddivision	848
LwpMenuMeAddmergevar	166	LwpMenuMdivGroupdivision	854
LwpMenuMeAdddocdescvar	167	LwpMenuMsecSectionproperties	849
LwpMenuMxMdiswitch	168	LwpMenuMsecNewsection	850
LwpMenuMeSeclevaction	169	LwpMenuMsecCombinesections	851
LwpMenuMeDefpaths	170	LwpMenuMsecSelectsection	852
LwpMenuMeFootopts	171	LwpMenuMsecExpandsection	853
LwpMenuMeRedo	172	LwpMenuMtabSettabs	855
LwpMenuMeDocoptions	173	LwpMenuMtabCleartabs	856
LwpMenuMeDefpath	174	LwpMenuMtabLefttab	857
LwpMenuMeDefsty	175	LwpMenuMtabRighttab	858
LwpMenuMeDefback	176	LwpMenuMtabCentertab	859
LwpMenuMeFootnoteinsert	177	LwpMenuMtabNumerictab	862
LwpMenuMeInsertnote	178	LwpMenuMtabDefaulttabs	865
LwpMenuMeDelete	179	LwpMenuMtabRemoveruler	866
LwpMenuMeIndexentry	180	LwpMenuMtabHideruler	867
LwpMenuMeAddnote	181	LwpMenuMtabQuickright	990
LwpMenuMeMergevar	182	LwpMenuMtabQuickcenter	991
LwpMenuMeDescvar	183	LwpMenuMrSetmarginsinfox	870
LwpMenuMeFormatcheckbar	184	LwpMenuMfToggleinfox	871
LwpMenuMePageup	185	LwpMenuMfCreateversion	872
LwpMenuMePagedown	186	LwpMenuMfCreateversionDivision	905
LwpMenuMeRevinsertion	187	LwpMenuMvViewversion	880
LwpMenuMeProtected	188	LwpMenuMafidHelp	900
LwpMenuMeNohyphen	189	LwpMenuMafidNotfound	901
LwpMenuMeGetglosname	198	LwpMenuMfMailNextstop	902

LwpMenuMeMacpath	286	LwpMenuMfMailEditroute	903
LwpMenuMeTocentry	835	LwpMenuMergeScriptNotify	910
LwpMenuMeBullet	836	LwpMenuFirstpopupid	925
LwpMenuMeMergedatafile	562	LwpMenuRenderformat	860
LwpMenuMeMergeextdesc	564	LwpMenuRenderallformats	861
LwpMenuMeMergevarsdatafile	566	LwpMenuMcCommenttoolsOn	889
LwpMenuMePastesimal	588	LwpMenuMnDeletenote	890
LwpMenuMeInsertobject	589	LwpMenuMnClosenote	891
LwpMenuMeChangelink	590	LwpMenuMnNextnote	892
LwpMenuMeUseworkpath	591	LwpMenuMnPrevnote	893
LwpMenuMeSelectword	595	LwpMenuMcCommenttools	894
LwpMenuMeSelectsentence	596	LwpMenuMcHighnote	895
LwpMenuMeSelectparagraph	597	LwpMenuMnCloseall	896
LwpMenuMeSelectfile	598	LwpMenuMnOpenall	897
LwpMenuMeRevisionmode	599	LwpMenuMnDeleteall	898
LwpMenuMIDoccontrol	600	LwpMenuMnShowinitials	899
LwpMenuMfSharedoccontrol	602	LwpMenuMnNumbernotes	900
LwpMenuMIDoccontrolEditorpanel	603	LwpMenuMfQuickprint	946
LwpMenuMIDoccontrolExternal	604	LwpMenuMdDebugger	1000
LwpMenuMvViewversionExternal	605	LwpMenuMdDebugDoc	1001
LwpMenuMfDocdescExternal	606	LwpMenuMdDebugPres	1002
LwpMenuMfMrglabel	607	LwpMenuMdDebugDivinfo	1003
LwpMenuMfMrgletterssetup	608	LwpMenuMdDebugCont	1004
LwpMenuMfMrgenvelopesetup	609	LwpMenuMdDebugCol	1005
LwpMenuMfMrgviewprint	612	LwpMenuMdDebugLine	1006
LwpMenuNwobjects	190	LwpMenuMdDebugVobj	1007
LwpMenuMoOpennw	191	LwpMenuMdDebugHeap	1008
LwpMenuMoCreatenw	192	LwpMenuMdDebugMain	1009
LwpMenuMoListnw	193	LwpMenuMdDebugProf	1010
LwpMenuMoLinksnw	194	LwpMenuMdDebugVcnt	1011
LwpMenuMoAttribnw	195	LwpMenuMdDebugDivision	1012
LwpMenuViewmenu	200	LwpMenuMdDebugLay	1013
LwpMenuMvFit	201	LwpMenuMdDebugVermgr	1014
LwpMenuMvWorking	202	LwpMenuMdDebugVerctrl	1015
LwpMenuMv75	796	LwpMenuMdDebugDoclay	1016
LwpMenuMv150	797	LwpMenuMdDebugPara	1018
LwpMenuMv100	203	LwpMenuMdDebugInfo	1019
LwpMenuMv200	204	LwpMenuMdDebugAtom	1020
LwpMenuMvSwitchiconstate	205	LwpMenuMdDebugWrdarg	1021
LwpMenuMvPagesort	771	LwpMenuMdDebugDwrdarg	1022
LwpMenuMvZoomoptions	772	LwpMenuMdDebugStrarg	1023
LwpMenuMvComment	774	LwpMenuMdDebugSline	1024
LwpMenuMvShowheadfoot	775	LwpMenuMdDebugPhint	1025
LwpMenuMvCustom	773	LwpMenuMdDebugChint	1026
LwpMenuMvFacingpages	776	LwpMenuMdDebugMsgtrace	1027

LwpMenuMvMultiplepages	777	LwpMenuMdDebugFoundry	1028
LwpMenuMvPagemargin	778	LwpMenuMdDebugCounts	1029
LwpMenuMvPagewidth	779	LwpMenuMdDebugPresinfo	1030
LwpMenuMvSpecialview	803	LwpMenuMdDebugMeter	1031
LwpMenuMvClearspecialview	881	LwpMenuMdDebugStack	1032
LwpMenuMvSynchronizeviews	868	LwpMenuMdDebugPtrholder	1033
LwpMenuMvClearhighlight	874	LwpMenuMdDebugMemory	1034
LwpMenuMvPreferences	206	LwpMenuMdDebugMemorydump	1035
LwpMenuMvTabrack	207	LwpMenuMdDebugContent	1036
LwpMenuMvFunctionbar	208	LwpMenuMdDebugSilver	1037
LwpMenuMvChgsidebar	209	LwpMenuMdDebugVernode	1038
LwpMenuMvLayout	210	LwpMenuMdDebugCheckpoint	1039
LwpMenuMvDraft	211	LwpMenuMdDebugSocket	1040
LwpMenuMvFacing	212	LwpMenuMdDebugSfb	1041
LwpMenuMvFacecancel	214	LwpMenuMdDebugDemandload	1042
LwpMenuMvShowpicts	215	LwpMenuMdDebugNumberhint	1043
LwpMenuMvShowmarks	216	LwpMenuMdDebugDialog	1044
LwpMenuMvPreferdefaults	217	LwpMenuMdDebugBack	1045
LwpMenuMvOutline	218	LwpMenuMdDebugHeadtailholder	1046
LwpMenuMvFloatingicon	219	LwpMenuMdDebugNotifypersistent	1047
LwpMenuMvLefticon	220	LwpMenuMdDebugObjectholder	1048
LwpMenuMvRighticon	221	LwpMenuMdDebugEnumobjhead	1049
LwpMenuMvTopicon	222	LwpMenuMdDebugNotify	1050
LwpMenuMvBottomicon	223	LwpMenuMdDebugNamedobjectholder	1051
LwpMenuMvSmarticons	232	LwpMenuMdDebugSection	1052
LwpMenuMvFullscreen	233	LwpMenuMdDebugFontmgr	1053
LwpMenuMvFullscreenoptions	234	LwpMenuMdDebugSanity	1054
LwpMenuMvEditicon	236	LwpMenuMdDebugAtomWatch	1055
LwpMenuMfPrintenvelope	237	LwpMenuMdDebugAtomCount	1056
LwpMenuMvNexticonset	239	LwpMenuMdDebugPieceCount	1057
LwpMenuMvSaveicon	240	LwpMenuMdDebugObjectSizes	1058
LwpMenuMvCreateicon	241	LwpMenuMdDebugTextstyle	1059
LwpMenuMvFullscreenstartup	242	LwpMenuMdDebugEnd	1099
LwpMenuMvSectionontabs	244	LwpMenuMkPageup	1100
LwpMenuCharmenu	250	LwpMenuMkPagedown	1101
LwpMenuMcTextinfobox	283	LwpMenuMcDooledragdrop	1102
LwpMenuMcTextinfoboxmouse	610	LwpMenuMsMplMargins	1300
LwpMenuMcDrawinfoboxmouse	611	LwpMenuMsMplPage	1301
LwpMenuMcFont	251	LwpMenuMsMplFooter	1302
LwpMenuMcNormal	253	LwpMenuMsMplHeader	1303
LwpMenuMcBold	254	LwpMenuMsMplLines	1304
LwpMenuMcItalic	255	LwpMenuMrMfiType	1400
LwpMenuMcUnderline	256	LwpMenuMrMfiSizepos	1401
LwpMenuMcWordunder	257	LwpMenuMrMfiLines	1402
LwpMenuMcLeft	259	LwpMenuMrMfiColumnstabs	1403

LwpMenuMcCenter	260	LwpMenuMISetmacrofile	1500
LwpMenuMcRight	261	LwpMenuMICompilemacro	1501
LwpMenuMcJustify	262	LwpMenuMIMacrorun	1502
LwpMenuMcAlignment	263	LwpMenuMIMacroappend	1503
LwpMenuMcCaps	264	LwpMenuMIPrototype	1504
LwpMenuMcIndentall	247	LwpMenuMeSmartcorrect	1505
LwpMenuMcIndentfirst	248	LwpMenuMrChart	1506
LwpMenuMcIndentrest	249	LwpMenuMrCharttodraw	1507
LwpMenuMcExtract	246	LwpMenuMmCascades	1600
LwpMenuMcMacrofastformat	238	LwpMenuSubmenuInsert	1600
LwpMenuMcFontrevert	243	LwpMenuMmFields	1601
LwpMenuMcEnlargetext	252	LwpMenuSubmenuMarktextas	1602
LwpMenuMcReducetext	245	LwpMenuSubmenuAlignment	1603
LwpMenuMcIndent	258	LwpMenuMmCaps	1604
LwpMenuMcOutdent	284	LwpMenuMmMacros	1607
LwpMenuMcBulletpanel	285	LwpMenuSubmenuInsertsmartsum	1609
LwpMenuMcNextfacename	950	LwpMenuMMailmemo	1610
LwpMenuMcNextalignment	951	LwpMenuMPrinttofile	1611
LwpMenuMcNextattribute	952	LwpMenuSubmenuUsersetup	1612
LwpMenuMcNextbullet	953	LwpMenuSubmenuScripts	1613
LwpMenuMcNextnumbering	954	LwpMenuMmShowoptions	1614
LwpMenuMcNextstyle	955	LwpMenuSubmenuQuickdemo	1615
LwpMenuMcDoubleunder	270	LwpMenuSubmenuOtherhelpstuff	1616
LwpMenuMcSuperscript	271	LwpMenuMmOtherframe	1617
LwpMenuMcSubscript	272	LwpMenuMmOthertable	1618
LwpMenuMcAllcaps	273	LwpMenuSubmenuOthertextstuff	1619
LwpMenuMfEnvelopeprint	274	LwpMenuSubmenuPriority	1620
LwpMenuMcAlllower	287	LwpMenuMmOtherstuff	1621
LwpMenuMcInitcaps	288	LwpMenuSubmenuSelect	1622
LwpMenuMcSmallcaps	289	LwpMenuSubmenuOtherdocpart	1623
LwpMenuMcJustfont	278	LwpMenuSubmenuAttributes	1624
LwpMenuMcJustpointsize	279	LwpMenuSubmenuOutline	1625
LwpMenuMcStrikethru	280	LwpMenuSubmenuCollapse	1626
LwpMenuMcFastformat	291	LwpMenuSubmenuEdit	1627
LwpMenuMcHighlight	294	LwpMenuSubmenuDebug	1628
LwpMenuStylemenu	300	LwpMenuSubmenuDocprops	1629
LwpMenuMsCreate	301	LwpMenuSubmenuZoomto	1630
LwpMenuMsStylemanage	303	LwpMenuSubmenuShowhide	1631
LwpMenuMsChgstyle	304	LwpMenuSubmenuOtherviewstuff	1632
LwpMenuMsSavestyle	305	LwpMenuSubmenuNamedstyles	1633
LwpMenuMsSelectstyle	306	LwpMenuSubmenuDelete	1634
LwpMenuMsOutlinestyle	320	LwpMenuSubmenuFile	1635
LwpMenuMsStylemanageinit	321	LwpMenuSubmenuView	1636
LwpMenuMsStylemanageaction	322	LwpMenuSubmenuTeammail	1637
LwpMenuMsChangestylename	323	LwpMenuSubmenuEditmodemenu	1638

LwpMenuMsDefinestyle	324	LwpMenuSubmenuDebugmodemenu	1639
LwpMenuMsModcreatestyle	325	LwpMenuSubmenuHelp	1640
LwpMenuMsStylemanagefinish	326	LwpMenuCreatemenu	1650
LwpMenuMsRedefinestyle	336	LwpMenuParallelstylemenu	1658
LwpMenuMsMargins	328	LwpMenuEnvAddress	1661
LwpMenuMsPagesettings	329	LwpMenuEnvSave	1665
LwpMenuMsApply	330	LwpMenuEnvPrint	1666
LwpMenuMsStylemru1	331	LwpMenuMfCreatetabdivision	1670
LwpMenuMsStylemru2	332	LwpMenuMfCreatedivision	1667
LwpMenuMsStylemru3	333	LwpMenuMeTocassist	1668
LwpMenuMsStylemru4	334	LwpMenuMpNewsection	1669
LwpMenuMrManualtable	335	LwpMenuMtInsertclickhere	1671
LwpMenuMrFrameanchorinfolbox	391	LwpMenuMcTextlinepanel	1672
LwpMenuMrFramelineinfolbox	392	LwpMenuMvOtherviewstuff	1673
LwpMenuMrFramebackinfolbox	393	LwpMenuMoApplystyle	1674
LwpMenuMrFramegfxinfolbox	394	LwpMenuMsFramecreate	1675
LwpMenuMrFrametopalign	395	LwpMenuMsFrameredefinestyle	1676
LwpMenuMrFramecenteralign	396	LwpMenuMsTablecreate	1677
LwpMenuMrFramebottomalign	397	LwpMenuMsTableredefinestyle	1678
LwpMenuMrDeleteframe	399	LwpMenuMsTablecellcreate	1679
LwpMenuPagemenu	350	LwpMenuMsTablecelledefinestyle	1680
LwpMenuMpCreatelayout	351	LwpMenuMvShowmisspelled	1681
LwpMenuMpModifylayout	352	LwpMenuMtDelrowacc	1682
LwpMenuMpTabs	355	LwpMenuMtCtrlretacc	1683
LwpMenuMpPagenum	356	LwpMenuMsQuickredefinestyle	1684
LwpMenuMpPagebreak	358	LwpMenuMtInsertrowacc	1685
LwpMenuMpLinenummer	359	LwpMenuMrOleconvert	1686
LwpMenuMpHeadfoot	360	LwpMenuMrOleconvertandactivate	1687
LwpMenuMpModlayoutinit	361	LwpMenuMrFrameleft	1688
LwpMenuMpModlaypagesettings	362	LwpMenuMrFramecenter	1689
LwpMenuMpModlayrightfacepage	363	LwpMenuMrFrameright	1690
LwpMenuMpModlayrightheadfacepage	364	LwpMenuMrFramehorzspan	1691
LwpMenuMpModlayrightfooterfacepage	365	LwpMenuMrFramevertspan	1692
LwpMenuMpModlayrightlines	366	LwpMenuMfNewanytab	1693
LwpMenuMpModlayleftfacepage	367	LwpMenuMfOdmaopen	1694
LwpMenuMpModlayleftheaderfacepage	368	LwpMenuMfOdmainert	1695
LwpMenuMpModlayleftfooterfacepage	369	LwpMenuMfOdmaconvert	1696
LwpMenuMpModlayleftlines	370	LwpMenuMfOdmasaveas	1697
LwpMenuMpDellayout	373	LwpMenuMfOdmaexternalize	1698
LwpMenuMpRevertlayout	374	LwpMenuMfOdmaexport	1699
LwpMenuMpInslayout	375	LwpMenuMrOleverb	1700
LwpMenuMpModlayoutfinish	376	LwpMenuMrOleverbmax	1799
LwpMenuMpHeaderfootergoto	377	LwpMenuMrOledivtabverb	1800
LwpMenuMpMacinslayout	378	LwpMenuMrOledivtabverbmax	1899
LwpMenuMpParallel	379	LwpMenuMtSelectcell	1900

LwpMenuMpPcolinfox	380	LwpMenuMtPcolselectcell	1901
LwpMenuMpPageproperties	381	LwpMenuMpColumnbreak	1902
LwpMenuMpCreate	382	LwpMenuMeViewtablegrid	1903
LwpMenuMpNewpagestyle	383	LwpMenuMeViewtableheading	1904
LwpMenuMpHeaderproperties	384	LwpMenuMeViewpcolguides	1905
LwpMenuMpFooterproperties	385	LwpMenuMeViewmarginguides	1906
LwpMenuMpPagepropertiesmouse	386	LwpMenuMeViewpagegauge	1907
LwpMenuMpFooterpropertiesmouse	387	LwpMenuMeViewshowallmarks	1908
LwpMenuMpHeaderpropertiesmouse	388	LwpMenuMeViewhideallmarks	1909
LwpMenuMpPcolinfoxmouse	389	LwpMenuMtTableleft	1910
LwpMenuFramemenu	400	LwpMenuMtTablecenter	1911
LwpMenuMrFrametext	402	LwpMenuMtTableright	1912
LwpMenuMrScale	403	LwpMenuMtTablehorzspan	1913
LwpMenuMrGroupframes	404	LwpMenuMtGotoinfox	1914
LwpMenuMrFrameinfox	405	LwpMenuMhTour	1915
LwpMenuMrBringtofront	407	LwpMenuMfCreateoledivision	1917
LwpMenuMrSendtoback	408	LwpMenuMfToggleclickhereprompts	1918
LwpMenuMrAdd	409	LwpMenuMcToggleskipbullet	1919
LwpMenuMrSendbackone	410	LwpMenuMISkipmisspelledword	1920
LwpMenuMrDrawingicon	411	LwpMenuMISkipallmisspelledword	1921
LwpMenuMrChartingicon	412	LwpMenuMIAddtouserdictionary	1922
LwpMenuMrCreateframe	413	LwpMenuMITogglebubblehelp	1923
LwpMenuMrAddframewithdlg	414	LwpMenuMfPlainnew	1924
LwpMenuMrFramedefaults	415	LwpMenuMvToggleshowanchors	1925
LwpMenuMrFramelines	416	LwpMenuMvNextnewestversion	1926
LwpMenuMrFrametype	417	LwpMenuMvMostrecentversion	1927
LwpMenuMrFrameborders	418	LwpMenuMvOldestversion	1928
LwpMenuMrFramecolumns	419	LwpMenuMcJustifyall	1929
LwpMenuMrModframereflow	420	LwpMenuMcClosegraphicole	1930
LwpMenuMrModframeinit	421	LwpMenuMfFramerevert	1931
LwpMenuMrBringfrontone	390	LwpMenuMtCellrevert	1932
LwpMenuMrSelectframe	398	LwpMenuMfOpentabdivision	1933
LwpMenuMrEquationsicon	423	LwpMenuMfNewtabdivision	1934
LwpMenuMrImageprocessing	424	LwpMenuMtInsertclickherelink	1935
LwpMenuMrFrameinfoxmouse	620	LwpMenuMtEditclickherelink	1936
LwpMenuMrFramelineinfoxmouse	621	LwpMenuMsecDeletesection	1949
LwpMenuMrFramebackinfoxmouse	622	LwpMenuMfExternalfileexit	1950
LwpMenuMrFramegfxinfoxmouse	623	LwpMenuMfExternalfilesavedivision	1951
LwpMenuMrFrameanchorinfoxmouse	624	LwpMenuMfFtpopen	1952
LwpMenuMrExpert	911	LwpMenuMfFtpinsert	1953
LwpMenuMrExpertmax	924	LwpMenuMfFtpconvert	1954
LwpMenuMrWppack	349	LwpMenuMfFtpsaveas	1955
LwpMenuMrAfid	425	LwpMenuMfFtpexternalize	1956
LwpMenuMrAfidmax	449	LwpMenuSubmenuInternet	1957
LwpMenuToolmenu	450	LwpMenuMfFtpoptions	1958

LwpMenuMISearch	451	LwpMenuMfOdmashowattributes	1959
LwpMenuMIGoto	452	LwpMenuMrRuby	1960
LwpMenuMISpellopts	453	LwpMenuMfFtpopendivision	1961
LwpMenuMIUserdict	454	LwpMenuMfFtpopenurl	1962
LwpMenuMIThesaurus	455	LwpMenuIbShowfrommenu	1963
LwpMenuMISort	456	LwpMenuMoHideinfobox	1964
LwpMenuMITables	457	LwpMenuMfWelcome	1965
LwpMenuMIGotolast	458	LwpMenuMsQuickredefineframestyle	1969
LwpMenuMISpellopts2	459	LwpMenuMsQuickredefinetablestyle	1970
LwpMenuMIRemarkoptions	461	LwpMenuMsQuickredefinecellstyle	1971
LwpMenuMIToc	462	LwpMenuMsQuickredefinestyle	1972
LwpMenuMIAssignmacro	463	LwpMenuMeInsertsymbol	1973
LwpMenuMIMacros	464	LwpMenuMeWordcount	1974
LwpMenuMIMacroaccel	422	LwpMenuMeDropcaps	1975
LwpMenuMIGeneratetoc	465	LwpMenuMfOdmimportpicture	1976
LwpMenuMIIndexnextfile	281	LwpMenuMrOleobjectinfobox	1977
LwpMenuMIBookmark	466	LwpMenuMcInternettoolson	1978
LwpMenuMIGlossary	467	LwpMenuMcInternettools	1979
LwpMenuMIGlosset	468	LwpMenuMhLotushomepage	1980
LwpMenuMIMacpause	469	LwpMenuMhLotussupport	1981
LwpMenuMIGenerateindex	470	LwpMenuMhLotusftpsite	1982
LwpMenuMISpelladdict	471	LwpMenuMrLinkframes	1983
LwpMenuMISpellskipall	472	LwpMenuMrUnlinkframes	1984
LwpMenuMISpellreplace	474	LwpMenuMfMrgwelcome	1985
LwpMenuMISpell	475	LwpMenuMfHtmloptions	1986
LwpMenuMISpellinit	476	LwpMenuMcUpdateindex	1987
LwpMenuMISpellcheck	477	LwpMenuMcUpdatetoc	1988
LwpMenuMISpellcancel	478	LwpMenuMvDesignmode	1989
LwpMenuMISpellskip	479	LwpMenuMfNosopen	1990
LwpMenuMIReplace	480	LwpMenuMfNossaveas	1991
LwpMenuMIChglang	481	LwpMenuMfMailSendmessage	1992
LwpMenuMIRenumseq	482	LwpMenuMcInsertdefnumber	1993
LwpMenuMISRattrs	485	LwpMenuMcInsertdefbullet	1994
LwpMenuMIRepopts	486	LwpMenuMeTogglesmartselect	1995
LwpMenuMISropts	487	LwpMenuMcCreateobjectviewer	1996
LwpMenuMISearchonly	488	LwpMenuMcCreateobjectcomment	1997
LwpMenuMIMacplay	489	LwpMenuMcCreateobjectdraw	1998
LwpMenuMIMacassign	490	LwpMenuMcCreateobjectschedule	1999
LwpMenuMIMacdde	491	LwpMenuMcCreateobjectspreadsheet	2000
LwpMenuMIMactype	492	LwpMenuMcCreateobjectchart	2001
LwpMenuMICall	493	LwpMenuMhInternetsearch	2002
LwpMenuMISammy2	494	LwpMenuMhLotussmartsuite	2003
LwpMenuMITocopts	495	LwpMenuMcClickherekeyword	2004
LwpMenuMISRfindmatch	496	LwpMenuMfPublishweb	2005
LwpMenuMISRreplacethenfind	497	LwpMenuMICrossreference	2006

LwpMenuMISrcancel	498	LwpMenuMtabHidevertruler	2007
LwpMenuMISrinit	499	LwpMenuMpUpdateheaders	2008
LwpMenuMIDoccompare	500	LwpMenuMpUpdatefooters	2009
LwpMenuMIMacroedit	501	LwpMenuMtEditclickhere	2010
LwpMenuMIMacroplay	502	LwpMenuMfShowwebauthor	2011
LwpMenuMIMacrorecord	503	LwpMenuMfHidewebauthor	2012
LwpMenuMIMacrooptions	504	LwpMenuMcWebauthorbegin	2013
LwpMenuMIQuickrec	505	LwpMenuMcWebcreatelink	2013
LwpMenuMIQuickplay	506	LwpMenuMcWebcreatehr	2014
LwpMenuMIGrammar	507	LwpMenuMcWebcreatewallpaper	2015
LwpMenuMIGrammaropts	508	LwpMenuMcWebcreateformcontrol	2016
LwpMenuMIGrammarstart	509	LwpMenuMcWebauthorend	2024
LwpMenuMIGrammarinit	510	LwpMenuMrRubyAbove	3000
LwpMenuMIModcreaterule	511	LwpMenuMrRubyBelow	3001
LwpMenuMIGrammarnext	512	LwpMenuMrRubyDelete	3002
LwpMenuMIGrammarcancel	513	LwpMenuMxMinmacromenu	9000
LwpMenuMIGrammarsugtext	514	LwpMenuMxMaxmacromenu	9099
LwpMenuMIGrammaroptionstext	515	LwpMenuMrAfidmenu	10000
LwpMenuMIGrammarchange	516	LwpMenuMrAfidmenumax	16000
LwpMenuMIGrammarstats	517	LwpMenuMaNotesflow	16001
LwpMenuMIIconpath	518	LwpMenuMaNotesflowmax	16200
LwpMenuMICyclekeysetup	519	LwpMenuFcsTextmenu	0
LwpMenuMISmartfill	520	LwpMenuFcsFramemenu	1
LwpMenuMINewspell	521	LwpMenuFcsFrametextmenu	2
LwpMenuMIRevisionbar	522	LwpMenuFcsTabletextmenu	3
LwpMenuMIToa	523	LwpMenuFcsFramegraphicmenu	4
LwpMenuWindowmenu	4	LwpMenuFcsPowerfieldmenu	5
LwpMenuMwNewwindow	525	LwpMenuFcsParallelcoltextmenu	6
LwpMenuMwTilewindow	526	LwpMenuFcsDivisionmenu	7
LwpMenuMwCascadewindow	527	LwpMenuFcsSectionmenu	8
LwpMenuMwStartoffiles	528	LwpMenuFcsTablemenu	9
LwpMenuMwTilewindowhorz	582	LwpMenuFcsParallelcolmenu	10
LwpMenuMwSplitvertwindow	593	LwpMenuFcsTablegraphicmenu	11
LwpMenuMwSplithorzwindow	594	LwpMenuFcsParallelcolgraphicmenu	12
LwpMenuMwEndoffiles	548	LwpMenuFcsHeadermenu	13
LwpMenuMwFilesseparator	549	LwpMenuFcsFootermenu	14
LwpMenuMIMacresume	554	LwpMenuFcsRulermenu	15
LwpMenuMIReviewrevs	557	LwpMenuFcsNotemenu	16
LwpMenuMIRevacceptall	875	LwpMenuFcsOutlinemenu	17
LwpMenuMIRevcancelall	876	LwpMenuFcsOledivisionmenu	18
LwpMenuMIRevaccept	877	LwpMenuFcsFramechartmenu	19
LwpMenuMIRevcancel	878	LwpMenuFcsTablechartmenu	20
LwpMenuMIField	568	LwpMenuFcsParallelcolchartmenu	21
LwpMenuMIFieldnext	569	LwpMenuFcsTablecornermenu	22
LwpMenuMIFieldprev	570	LwpMenuFcsTablerowmenu	23

LwpMenuMIFieldeval	571	LwpMenuFcsTablecolumnmenu	24
LwpMenuMIUpdatedate	575	LwpMenuFcsVertrulermenu	25
LwpMenuMIUpdatealldates	576	LwpMenuFcsRubymenu	26
LwpMenuMIFieldadd	572	LwpMenuFcsFootnotetextmenu	40
LwpMenuMIFieldupdate	573	LwpMenuFcsIndextextmenu	41
LwpMenuMIFieldtog	266	LwpMenuFcsToctextmenu	42
LwpMenuMIFieldupall	267	LwpMenuFcsTablecornermenutoc	43
LwpMenuMIFieldlock	268	LwpMenuFcsTabletextmenutoc	44
LwpMenuMIFieldsave	269	LwpMenuFcsFrametextmenudropcap	45
LwpMenuMIFieldauto	224	LwpMenuFcsFramemenudropcap	46
LwpMenuMIFieldshowrit	225	LwpMenuFcsTextdatemenu	47

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Word Pro: WordUnderline method

{button ,AL('H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL('H_WORDUNDERLINE_METHOD_EXSCRIPT',1)} [See example](#)

Sets the WordUnderline attribute for selected text, or all following text, if no text is selected. Acts as a toggle, turning the attribute off if it is on, and on if it is off. Equivalent to choosing Text - Attributes - Other, and then choosing "Word Underline" from the Attributes box.

Syntax

[objectreference].WordUnderline()

Parameters

None.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Word Pro: WriteProfileString method

{button ,AL('H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

Writes a new profile string in the specified INI file or INI entry. You can write a profile string in any of the standard Word Pro INI files, or you can specify another INI file.

Syntax

[objectreference].WriteProfileString(Section, Key, NewString[, IniFileType][, WhichIniLocation][, IniName])

Parameters

Section

A String expression that specifies a name of a section in the INI. Word Pro searches only the section you name in this parameter. If the named section does not match a section in the specified INI, this method fails. If you use an empty string (""), Word Pro assumes you are writing to the LWPUSER.INI file (IniFileType parameter = \$LwplniUserPrefs) and looks for the "WordProUser" section. Most INIs have more than one section. The section name you provide in this parameter must match the section name in the INI exactly.

Key

A String expression that specifies the key name in the section you are writing.

NewString

A String expression that Word Pro writes as the value for the key.

IniFileType

Specifies the INI in which you want to write the new profile string. You can choose one of the standard Word Pro INI files or choose \$LwplniCustomFile to search another INI file. Data type is Variant, which allows the value of this parameter to be one of the constants listed below or its numeric equivalent (in parentheses). You do not have to provide a value for this parameter. Default is \$LwplniUserPrefs.

<u>Value</u>	<u>Effect</u>
\$LwplniUserPrefs (2101)	Default; the INI file used to store Word Pro's user preference information (lwpuser.ini).
\$LwplniConfigPrefs (2102)	The INI file used to store Word Pro's configuration preference information.
\$LwplniEnvelopeAndMerge (2105)	The INI file used to store Word Pro's envelope and merge information.
\$LwplniLanguages (2107)	The INI file used to store some of Word Pro's language information.
\$LwplniSharedLotusInfo (2103)	The INI file used to store shared information between Word Pro and other Lotus products.
\$LwplniSmartcorrect (2106)	The INI file used to store Word Pro's SmartCorrect information.
\$LwplniSmartfill (2104)	The INI file used to store Word Pro's SmartFill lists.
\$LwplniCustomfile (2100)	Allows you to write a profile string in an INI file which is not one of the standard Word Pro INI files. If you use this value, you must use the IniName parameter to specify the name of the INI file (Windows 3.1 or OS/2) or INI entry (Windows 95) in which the profile string is located.

WhichIniLocation

Tells Word Pro whether to look on the network or the local machine for the specified INI file. Data type is Variant, which allows the value of this parameter to be one of the constants listed below or its numeric equivalent (in parentheses). You do not have to provide a value for this parameter. Default is \$LwpUserIniLocation.

<u>Value</u>	<u>Effect</u>
\$LwpNetworkIniLocation (2171)	Searches directory for network INI files.
\$LwpUserIniLocation (2172)	Searches directory for user INI files.

Note For Windows 95, in the registry, the user location is HKEY_CURRENT_USER or HKEY_USERS. The network location is HKEY_LOCAL_MACHINE. Within either of these locations, there is a directory path which reflects the version of Word Pro which is running the script. For example, a user running Word Pro 97 sees Software\Lotus\WordPro\97.0 while a user running Word Pro Release 9 sees Software\Lotus\WordPro\98.0.

IniName

An optional String expression that identifies the INI to which you want to write. Use this parameter only if you used \$LwpIniCustomFile as the value of the IniFileType parameter. This INI must be stored in the same directory as the standard Word Pro INIs.

Note If you are using Windows 3.1 or OS/2, this value is an INI file name. If you are using Windows 95, this value is an INI entry as seen in the Windows Registry application (REGEDIT.EXE).

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Word Pro: Write method

{button ,AL(^H_BAG_CLASS',0)} [See list of classes](#)

{button ,AL(^H_WRITE_METHOD_EXSCRIPT',1)} [See example](#)

Syntax

[objectreference].Write(Data, Length)

Parameters

Data

Data type is String.

Length

Data type is Long.

Return value

Usage

Word Pro: Accelerators class

A short-cut key assignment for any Word Pro script or Ami Pro macro. With the methods in this class, you can add or remove any accelerator key assignment.

Base Classes

BaseObject

Derived Classes

None.

Contained by

ApplicationWindow in the Accelerators Property

Usage

Note that any accelerators you create and assign expire when you end the session of Word Pro in which you created those accelerators. To keep an accelerator in memory between Word Pro sessions, you must set the accelerator's IsTemporaryUse parameter to "0."

Word Pro: Alignment class

The alignment settings for an object.

Base Classes

BaseObject

Derived Classes

None.

Contained by

[ClickHere](#) in the [Alignment](#) Property

[Graphic](#) in the [Alignment](#) Property

[ParagraphStyle](#) in the [Alignment](#) Property

[Text](#) in the [Alignment](#) Property

[TextMarker](#) in the [Alignment](#) Property

Usage

Alignment objects are stored in the Alignment property of other objects. Use the syntax shown in the Alignment property description to make use of this class' properties and methods.

Word Pro: Amikake class

The text background object for the Asian language versions of Word Pro. If you are using an English language version of Word Pro, this class is not available.

Base Classes

BaseObject

Derived Classes

None.

Contained by

CharacterStyle in the Amikake Property

ClickHere in the Amikake Property

ParagraphStyle in the Amikake Property

Text in the Amikake Property

TextMarker in the Amikake Property

Usage

Word Pro: ApplicationWindow class

The Word Pro window which acts as the container for all of your document windows. Also known as the application workspace, this is the window that remains after you close all of your documents and leave Word Pro running.

Base Classes

BaseObjectWindow

Derived Classes

None.

Contained by

Application in the ApplicationWindow Property

WPApplication in the ApplicationWindow Property

Usage

This class is shared by all Lotus applications. Each Lotus application creates a single ApplicationWindow object when you launch the application. Word Pro's ApplicationWindow object is contained by the WPApplication class in the ApplicationWindow property. WPApplication inherits the ApplicationWindow property from the Application class.

Only one ApplicationWindow object is instantiated at any given time. The ApplicationWindow object allows you to access all Word Pro features which are available independently of the documents you create. For example, your preference settings, accelerator keys, SmartIcons bars, menus, the status bar, and other features are available, regardless of the document you have open. Many of these global features are accessed through the ApplicationWindow property in WPApplication.

Word Pro: Application class

An abstract class that acts as a template for the WPAApplication class. To access and manipulate the Word Pro application, use the Word Pro subclass named WPAApplication.

Base Classes

BaseObject

Derived Classes

WPAApplication

Contained by

None

Usage

Each Lotus application is represented in LotusScript by its own subclass of the Application class. By sharing a common parent class, each Lotus application inherits a common set of properties, methods, and events. This makes it possible for you to interact with each Lotus application in much the same way. You should also note that, as in all LotusScript classes, the Application class is itself a subclass of the BaseObject class.

The Application class lays the groundwork for a number of abilities and attributes which are common to all Lotus applications. These are inherited by each application's subclass and enhanced for that application's unique needs. For example, Application maintains application-wide settings and user information. Application also manages and creates documents. There is a single window associated with each application. WPAApplication inherits each of these traits and enhances them to meet the needs of the Word Pro application.

There is only one application object per running application instance. You cannot instantiate an object from the Application class.

Word Pro: AppViewPrefs class

The view preferences for a session of Word Pro.

Base Classes

BaseObject

Derived Classes

None.

Contained by

WPApplication in the AppViewPrefs Property

Usage

This class defines Word Pro's margin color, window pane color, selection border color, and spelling error colors.

Word Pro: Attributes class

The attributes of an object.

Base Classes

BaseObject

Derived Classes

None.

Contained by

[CharacterStyle](#) in the [TextAttributes](#) Property

[ClickHere](#) in the [Attributes](#) Property

[Editor](#) in the [TextAttributes](#) Property

[FindAndReplace](#) in the [ReplaceAttributes](#) Property

[FindAndReplace](#) in the [SearchAttributes](#) Property

[ParagraphStyle](#) in the [TextAttributes](#) Property

[RevisionDisplay](#) in the [TextAttributes](#) Property

[Text](#) in the [Attributes](#) Property

[TextMarker](#) in the [Attributes](#) Property

Usage

Word Pro: AutoRunMacro class

Contains the names of macros which run automatically each time a document is created, opened, or closed.

Base Classes

BaseObject

Derived Classes

None.

Contained by

TextDocument in the AutoRunMacro Property

Usage

Word Pro: Background class

The background of an object. Three properties correspond to the three elements of an object's background: the pattern, the pattern color, and the color of the null space behind the pattern.

Base Classes

BaseObject

Derived Classes

None.

Contained by

[Amikake](#) in the [Background](#) Property

[Layout](#) in the [Background](#) Property

[TableFill](#) in the [Background](#) Property

Usage

Setting the properties of a Background object is equivalent to setting the InfoBox options labeled "Pattern," "Pattern color," and "Background color" for a Layout or TableFill object.

Use the Pattern property to choose a pattern for the Background object. Use the Color property to choose a color for that pattern. Use the BackColor property to choose a color for the null space behind the pattern.

Word Pro: BagCollection class

A collection of Bag objects in the foundry of a specific division.

Base Classes

BaseObject\BaseCollection

Derived Classes

None.

Contained by

Foundry in the Bags Property

Usage

Word Pro: Bag class

Stores custom data for a division.

Base Classes

BaseObject

Derived Classes

None.

Contained by**Usage**

Word Pro: BaseCollection class

A virtual class which provides the basic members for all collection classes.

Base Classes

BaseObject

Derived Classes

[BagCollection](#)

[BookmarkCollection](#)

[CellCollection](#)

[CellLayoutCollection](#)

[CharacterStyleCollection](#)

[ClickHereCollection](#)

[ConnectedLayoutCollection](#)

[ContentCollection](#)

[DdeLinkCollection](#)

[DivisionCollection](#)

[DocInfoFieldCollection](#)

[Documents](#)

[DocWindowCollection](#)

[EditorCollection](#)

[EndnoteLayoutCollection](#)

[FooterLayoutCollection](#)

[FootnoteCollection](#)

[FootnoteLayoutCollection](#)

[FrameLayoutCollection](#)

[GlossaryCollection](#)

[GraphicCollection](#)

[GraphicOleObjectCollection](#)

[GroupLayoutCollection](#)

[HeaderLayoutCollection](#)

[IconBarCollection](#)

[LayoutCollection](#)

[MarkerCollection](#)

[MenuItemCollection](#)

[NoteLayoutCollection](#)

[OleObjectCollection](#)

[OutlineSeqCollection](#)

[OutlineSeqItemCollection](#)

[PageLayoutCollection](#)

[ParagraphStyleCollection](#)

[ParallelColsCollection](#)

[PowerFieldCollection](#)

[RowLayoutCollection](#)

[RubyLayoutCollection](#)

[SectionCollection](#)

[SilverBulletCollection](#)

[StatusBarButtonCollection](#)

[StringCollection](#)

[SuperTableCollection](#)

[SuperTableLayoutCollection](#)

[TableCollection](#)

[TableHeadingCollection](#)

[TableHeadingLayoutCollection](#)

[TableLayoutCollection](#)

[TableMarkerCollection](#)

[TableOnlyCollection](#)

[TextCollection](#)

[TextMarkerCollection](#)

[TextStyleCollection](#)

[VersionCollection](#)

[WPDataSetCollection](#)

Contained by

None

Usage

The BaseCollection class is an abstract class. That is, you cannot create an instance of BaseCollection, and there are no BaseCollection objects within Word Pro. This class is used to provide a foundation for a number of derived classes, such as FrameLayoutCollection, Documents, and MenuItemCollection. For more information on Abstract classes, see [Overview: Word Pro LotusScript Abstract Classes](#).

Word Pro: BaseContainer class

An abstract class which defines properties and methods that are common to all Word Pro container objects. An explanation of container objects is provided below under Usage. This information applies to all container objects in Word Pro, but each container object may exhibit minor differences. These differences are noted in the documentation of each specific container class.

Base Classes

BaseObject

Derived Classes

[CellContainer](#)

[FrameContainer](#)

[NoteContainer](#)

[PageContainer](#)

[ParallelColsContainer](#)

[RowContainer](#)

[RubyContainer](#)

[SubPageContainer](#)

[SuperPageContainer](#)

[SuperTableContainer](#)

[TableContainer](#)

[TableOnlyCont](#)

Contained by

[WPApplication](#) in the [Container](#) Property

Usage

Word Pro creates container objects as a means of giving you quick and easy access to a group of related objects. However, that access is provided only when those related objects have the focus. For example, a table cell is comprised of a group of related objects, including a CellLayout object, a Background object, a Borderlines object, and all the attributes of those objects. When you move the insertion point into that cell, we say that cell has the focus. When a cell gets the focus, Word Pro creates a container object from the CellContainer class to hold all the objects related to that cell.

Container objects are temporary and exist only as long as a group of related objects has the focus. If you move the insertion point to another object or group of objects, Word Pro destroys the container object, leaving the group of related objects intact. Only a handful of objects have related objects that can be pulled together in a container. They include pages, tables, parallel columns, super tables, cells, and frames. Each of these objects has related objects that get pulled into a container when you give that object the focus (for example, when you place the insertion point in that object). A container object is always stored in the same property on the WPApplication object. You can never have more than one kind of container object at any time.

The container object properties on the WPApplication object include:

Cell

Frame

Page

SuperTableContainer

TableContainer

TableOnlyContainer

You may also notice the Container property, which always contains the topmost container object in the focus. Container uses the abstract class, BaseContainer, as its data type. This allows the Container property to store any kind of container object.

Container properties remain empty, unless the focus includes a cell, a frame, a page, a table, or a combination of these objects. As you move the focus around in a document and different groups of related objects come into the focus, Word Pro creates temporary container objects and stores them in these container properties to give you easy

access to each group of related objects. While Word Pro never creates more than one of each type of container object at any given time, it is not unusual to have one of each kind of container object stored in the WPApplication properties.

An Example of Container Objects

Here's an example of how Word Pro manages container objects in a document with a table on the first page, a table and a frame on the second page, and a set of parallel columns on the third page:

When you place your insertion point on the first page, Word Pro creates a PageContainer object and stores it in the Page property. When you move your insertion point to a table cell, Word Pro creates CellContainer, TableOnlyCont, and SuperTableContainer objects, and places them in their respective container properties. Word Pro leaves the PageContainer object for page one intact because you never moved your insertion point (and thus the focus) off page one.

When you move the insertion point to page two, Word Pro destroys all the container objects from the first page and creates a new PageContainer object for the second page. When you move the insertion point to a cell in the table on the second page, Word Pro creates container objects for that table, its super table, and the cell in which you placed the insertion point. When you move the insertion point to the frame on the second page, Word Pro destroys the table, super table, and cell container objects, and creates a FrameContainer object. Word Pro leaves the PageContainer object for page two intact because your insertion point never left page two.

When you move the insertion point to page three, Word Pro destroys all the container objects from the second page. Page three has a set of parallel columns. Word Pro sees parallel columns as a special kind of table. In fact, the Table class and ParallelColumns class are derived from the same parent class (BaseTable), and the container classes (TableOnlyCont and ParallelColsContainer) for these objects are derived from the same parent class (TableContainer). You may notice that there is a property for the TableOnlyCont object, but no property set aside for the ParallelColsContainer object. Word Pro provides a place for the ParallelColsContainer object in the TableContainer property by specifying the data type of the TableContainer property as TableContainer (the parent class for ParallelColsContainer). Using the parent class as the data type allows Word Pro to store either a ParallelColsContainer or a TableOnlyCont object in the TableContainer property. Therefore, when you move the insertion point onto page three, Word Pro creates a PageContainer object for the Page property and a ParallelColsContainer object for the TableContainer property.

Using Containers To Access The Appropriate Layout

In most cases, you will use container objects as a means of selecting a single Layout object when the focus encompasses several layout objects. In the example above, if your focus was on a table cell and you wanted to access the PageLayout object, you cannot use the statement:

```
.Layout.layoutpropertyname
```

This statement returns the layout of the cell. Instead, you need a way of specifying which layout object you want. The container objects make this easy because you can always use a statement, such as:

```
Page.Layout
```

or

```
TableOnlyContainer.Layout
```

Variables and Container Objects

Remember that a container object is temporary and changes with the focus. So if the insertion point was in cell (1,1) of a table, and you assign the container object to a variable (myCellContainer), such as:

```
set myCellContainer = .Cell
```

You can call the variable and get cell (1,1), as long as the focus remains on cell (1,1). However, if you move the focus to cell (2,2) and try to use that same variable as in the statement below:

```
myCellContainer.Layout.Content.InsertText "Hello"
```

Word Pro inserts the text, "Hello," into cell (2,2) because the variable, myCellContainer, contains the CellContainer object and that object changed from cell (1,1) to cell (2,2) when you moved the focus. Container objects always refer to the current context, and the variables that contain container objects also change with the focus.

Finally, if you move the focus out of the table entirely so that the insertion point is not within a cell at all, then this statement results in an error:

```
myCellContainer.Layout.Content.InsertText "Hello"
```

This is because the myCellContainer variable must contain a container object for the statement to work, but the object is destroyed when you move the focus out of the table.

Word Pro: BaseObject class

The BaseObject class is shared among all Lotus applications and is a true virtual class. No instances of the BaseObject class are ever created. BaseObject exists to provide a basic set of properties to all Word Pro classes. It is important to remember that, while the properties in the BaseObject class are inherited by every Word Pro class, not every Word Pro class actually implements each property. Check the descriptions for the BaseObject properties to see which are implemented by Word Pro classes.

Base Classes

BaseObject

Derived Classes

All classes, except [OLEControl](#), are derived from BaseObject.

Contained by

[BaseObject](#) in the [Parent](#) Property

Usage

BaseObject is one of the shared Lotus Object Interface (LOI) classes. The BaseObject class exists solely as a means of providing a common set of class members to all other classes in Word Pro and other Lotus applications. Word Pro never instantiates an object from the BaseObject class alone, and you cannot instantiate an object from the BaseObject class. The properties defined in the BaseObject class are inherited, directly or indirectly, by every class in the Word Pro object model. However, not every class implements these inherited properties. For example, every Word Pro class inherits the Description property, but only the DocInfo class implements the property for you to use in your scripts.

Another feature of deriving Word Pro classes from the BaseObject class is the ability to store any Word Pro object in a variable of type BaseObject. When you store an object in this manner, you can access only those class members which the object inherited from the BaseObject class and implemented for your use. For Word Pro, the BaseObject class defines only six class members, all of them properties. With this basic set of properties, you can get basic information about any object in the Word Pro object model, including the object's base application, parent, version, and whether or not the object is available for your use in a specific context. For more information on which classes implement the BaseObject properties, see the individual property descriptions.

Word Pro: BaseTable class

An abstract class acting as the parent class for several types of tables, glossaries, and parallel columns. This class is used as the data type for the BaseTable property so that property can contain any object created from one of BaseTable's derived classes.

Base Classes

BaseObject\Content

Derived Classes

FootnoteTable

Glossary

ParallelColumns

Table

TableHeading

Contained by

WPAApplication in the BaseTable Property

Usage

This abstract class describes the behavior that is common to FootnoteTables, Tables, TableHeadings, Glossaries, and ParallelColumns, all of which are derived from BaseTable. Note that the BaseTable class is derived from the Content class. This allows the objects created from classes derived from BaseTable to provide the content for other objects. These objects typically contain RowLayouts, ColumnLayouts, and CellLayouts. Each CellLayout has its own content which may be any of the following Content types: Text, Graphic, OleObject.

Word Pro: BookmarkCollection class

A collection of bookmark objects in the BookmarkManager class.

Base Classes

BaseObject\BaseCollection

Derived Classes

None

Contained by

BookmarkManager in the Bookmarks Property

BookmarkManager in the BookmarksByMarkerName Property

Usage

Use this collection to access any of the bookmark objects in the BookmarkManager class.

Word Pro automatically creates and maintains one BookmarkCollection object for each division of a document. The BookmarkCollection object and its contents are stored with the document.

Word Pro: BookmarkManager class

A tool for managing bookmarks in a document. Keeps and manages the list of bookmarks in the document.

Base Classes

BaseObject

Derived Classes

None

Contained by

Division in the BookmarkManager Property

TextDocument in the BookmarkManager Property

Usage

Used in conjunction with Bookmark and BookmarkCollection objects. You can use the BookmarkManager to find, add, or remove bookmarks.

Note BookmarkManager objects are only valid if they are accessed through a Division object.

Word Pro: Bookmark class

A bookmark in a Word Pro document. Word Pro instantiates (creates an instance of) a Bookmark object each time you create a bookmark in a document. Once created, the bookmark name displays in the Bookmarks dialog box.

Base Classes

BaseObject

Derived Classes

None.

Contained by**Usage**

Bookmarks are stored in BookmarkCollection objects. Each division in a document contains a BookmarkManager object, which in turn stores the division's BookmarkCollection.

In order to access all bookmarks in a document, you must iterate the collection of divisions with in the document to access each division's BookmarkManager object.

Word Pro: BorderLines class

The lines which comprise the border of an object.

Base Classes

BaseObject

Derived Classes

Gutter

Contained by

CharacterBorder in the BorderLines Property

FootnoteSepOpt in the BorderLines Property

Layout in the BorderLines Property

ParagraphBorder in the BorderLines Property

TableLine in the BorderLines Property

TableLine in the OutlineBorderLines Property

Usage

Most of the properties in BorderLines contain instances of the Border class. You can use the contents of these properties to make changes to the lines around an object. The complete syntax depends on the containing object but always follow this standard:

```
.objectname.BorderLines.AllBorders.property/method
```

The same syntax applies to BottomBorder, LeftBorder, RightBorder, and TopBorder. Each of these properties has a data type of Border, and therefore contains a Border object which has its own set of properties that you must use to achieve your desired results.

Word Pro: Border class

The attributes of a border, including the border's width, color, and pattern.

Base Classes

BaseObject

Derived Classes

None.

Contained by

BorderLines in the AllBorders Property

BorderLines in the BottomBorder Property

BorderLines in the RightBorder Property

BorderLines in the TopBorder Property

BorderLines in the LeftBorder Property

Usage

Nearly every object in Word Pro has a border of some sort. You can use this object in any object which contains an instance of the BorderLines class. The complete syntax depends on the containing object but always follow this standard:

```
.objectname.BorderLines.AllBorders.Color.Red=0  
.objectname.BorderLines.AllBorders.Color.Green=0  
.objectname.BorderLines.AllBorders.Color.Blue=255  
.objectname.BorderLines.AllBorders.Pattern=1  
.objectname.BorderLines.AllBorders.Width=200
```

The same syntax applies to BottomBorder, LeftBorder, RightBorder, and TopBorder. Each of these properties has a data type of Border and contains a Border object which comprises the top, bottom, left, right, and all border properties of the larger BorderLines object.

Word Pro: Breaks class

Break options as seen in the Advanced panel of the Text Properties dialog box.

Base Classes

BaseObject

Derived Classes

None.

Contained by

[ClickHere](#) in the [Breaks](#) Property

[ParagraphStyle](#) in the [Breaks](#) Property

[Text](#) in the [Breaks](#) Property

[TextMarker](#) in the [Breaks](#) Property

Usage

Use the properties of this class to set the Break options for an object. Use the RevertBreaksToStyle method to return all the break options to the options selected in the object's style.

Word Pro: Bullet class

Bullet options as seen in the Bullet panel of the Text Properties dialog box.

Base Classes

BaseObject

Derived Classes

None.

Contained by

[ClickHere](#) in the [Bullet](#) Property

[ParagraphStyle](#) in the [Bullet](#) Property

[Text](#) in the [Bullet](#) Property

[TextMarker](#) in the [Bullet](#) Property

Usage

Use the properties of this class to set the Bullet options for an object. Use the RevertBulletToStyle method to return all the bullet options to the options selected in the object's style.

Word Pro: CellCollection class

A collection of cell objects in the foundry of a specific division.

Base Classes

BaseObject\BaseCollection

Derived Classes

None.

Contained by

Foundry in the CellEngines Property

Usage

Use this collection to access any of the cell objects in the foundry of a specific division.

Word Pro: CellContainer class

The container object for table cells. This object only exists for one table cell at a time, and only when there is a table cell with in the focus. When a CellContainer object is present, it is stored in the Cell property on the WPApplication object.

Base Classes

BaseObject\BaseContainer

Derived Classes

None.

Contained by

WPApplication in the Cell Property

Usage

The primary use for a CellContainer object is to provide quick and easy access to the CellLayout object for the currently active cell. A CellContainer object always represents the cell that currently has the focus. Therefore, if you assign a CellContainer object to a variable, you can use that variable to access the currently active cell. However, you must remember that the cell referenced by the variable changes as the focus moves from one cell to another. This is because the variable references the CellContainer object, and the CellContainer object always represents the cell that has the focus. If there is no cell with in the focus, there is no CellContainer object. Therefore, a variable that stores a CellContainer object has a null value whenever the focus does not contain a cell. There is never more than one CellContainer object at any time.

For more information about container objects, see BaseContainer.

Word Pro: CellEngine class

This class allows you to access formulas within a table.

Base Classes

BaseObject

Derived Classes

None.

Contained by

Table in the CellEngine Property

Usage

Word Pro: CellGroupLayout class

The cell group layout for a cell group.

Base Classes

BaseObject\Layout

Derived Classes

None.

Contained by

WPApplication in the CurrentCell property

Usage

The CellGroupLayout class provides you with a way to access and modify the format and appearance of CellGroupLayout objects within your document. When more than one table cell is selected, the combined layout object is a CellGroupLayout object.

Since the CellGroupLayout class is derived from the Layout class, CellGroupLayout objects can be stored within properties of the Layout type. For example, the Layout property with in the BaseContainer class is of the Layout type. However, this property often stores objects of the CellGroupLayout type. The Layout property is implemented in this way, so that objects of other derived layout class types can be stored there as well. The Layout property with in the FrameContainer class, for example, may also contain objects of the NoteLayout type.

At many locations within your document, multiple layouts are available. For example, you may have multiple cells selected within a table. In this case, the cells and the table both have associated layout objects. These layout objects may be combined with other objects into related groups known as containers. For more information on containers and their associated layouts, see the Help topic titled Word Pro: BaseContainer class.

Word Pro: CellLayoutCollection class

A collection of cell layout objects in the foundry of a specific division.

Base Classes

BaseObject\BaseCollection

Derived Classes

None.

Contained by

Foundry in the CellLayouts Property

Foundry in the CellLayoutStyles Property

Usage

Use this collection to access any of the cell layout objects in the foundry of a specific division.

Word Pro: CellLayout class

The cell layout for a CellContainer object. This class inherits most of its members from the Layout class.

Base Classes

BaseObjectLayout

Derived Classes

ConnectedLayout

Contained by

BaseTable in the CurrentCell Property

CellContainer in the CellLayout Property

WPApplication in the CurrentCell Property

Usage

The layout object of a single cell is a CellLayout object. The layout object of a connected cell is a CellGroupLayout object.

Word Pro: CharacterBorder class

The border around a character in a document.

Base Classes

BaseObject

Derived Classes

None.

Contained by

[CharacterStyle](#) in the [CharacterBorder](#) Property

[ClickHere](#) in the [CharacterBorder](#) Property

[ParagraphStyle](#) in the [CharacterBorder](#) Property

[Text](#) in the [CharacterBorder](#) Property

[TextMarker](#) in the [CharacterBorder](#) Property

Usage

Word Pro: CharacterSet class

A set of characters used in Find and Replace.

Base Classes

BaseObject

Derived Classes

None.

Contained by

FindAndReplace in the CharacterSet Property

Preferences in the CharacterSet Property

Usage

Word Pro: CharacterStyleCollection class

A collection of character style objects in the foundry of a specific division.

Base Classes

BaseObject\BaseCollection

Derived Classes

None.

Contained by

Foundry in the CharacterStyles Property

Usage

Use this collection to access any of the character style objects in the foundry of a specific division.

Word Pro: CharacterStyle class

Contains the style used to create a character in a division.

Base Classes

BaseObject

Derived Classes

None.

Contained by

[ClickHere](#) in the [CharacterStyle](#) Property

[Text](#) in the [CharacterStyle](#) Property

[TextMarker](#) in the [CharacterStyle](#) Property

Usage

Word Pro: ClickHereCollection class

A collection of ClickHere objects in the foundry of a specific division.

Base Classes

BaseObject\BaseCollection

Derived Classes

None.

Contained by

Foundry in the ClickHeres Property

Usage

Use this collection to access any of the ClickHere objects in the foundry of a specific division application.

Word Pro: ClickHere class

A ClickHere block in a document.

Base Classes

BaseObject\Marker

Derived Classes

None.

Contained by**Usage**

ClickHere objects have many of the same properties and methods as Text objects. Note, however, that the events available on a ClickHere object are limited to EnterClickHere and ExitClickHere. Use these events to trigger scripts written for a ClickHere object.

Word Pro: Color class

Defines the color that is applied to a specific object.

Base Classes

BaseObject

Derived Classes

None.

Contained by

[AppViewPrefs](#) in the [MarginColor](#) Property

[AppViewPrefs](#) in the [SelectionBorderColor1](#) Property

[AppViewPrefs](#) in the [SelectionBorderColor2](#) Property

[AppViewPrefs](#) in the [SelectionBorderColor3](#) Property

[AppViewPrefs](#) in the [SpellColor](#) Property

[AppViewPrefs](#) in the [SpellFocusedColor](#) Property

[Background](#) in the [BackColor](#) Property

[Background](#) in the [Color](#) Property

[Border](#) in the [Color](#) Property

[DivisionInfo](#) in the [Color](#) Property

[Editor](#) in the [HiLiteColor](#) Property

[Font](#) in the [FontColor](#) Property

[Font](#) in the [BackColor](#) Property

[NoteLayout](#) in the [Color](#) Property

[NumericFormatSubset](#) in the [Color](#) Property

[Preferences](#) in the [HiLiteColor](#) Property

[Section](#) in the [Color](#) Property

[Shadow](#) in the [Color](#) Property

Usage

You can set a color for an object that has a color property. For example, objects, such as text, shadows, frames, and lines have color contexts associated with them. You can select a color context associated with an object and use that context to change the object's color.

You can set a color for an object in three ways:

- Use the SetRGB method to simultaneously assign the red, green, and blue color components to an object.
- Set the red, green, and blue color component values independently. Each one of the RGB values is a property of the Color class.
- Set the Override property to use predefined Word Pro colors (red, green, blue, black, white, light gray, dark gray, and transparent). When you use the Override property value, Word Pro does not recognize previously set RGB values.

Word Pro: ColumnGroupLayout class

The layout for a column group in a table object.

Base Classes

BaseObject

Derived Classes

None.

Contained by**Usage**

When multiple cells are selected within a table, the CurrentColumn property of the table object contains a ColumnGroupLayout object.

Word Pro: ConnectedLayoutCollection class

A collection of connected layout objects in the foundry of a specific division.

Base Classes

BaseObject\BaseCollection

Derived Classes

None.

Contained by

Foundry in the ConnectedLayouts Property

Usage

When two or more cells are connected, the resulting layout is a ConnectedLayout object. Use the ConnectedLayoutCollection to access any of the ConnectedLayout objects in the foundry of a specific division.

ConnectedLayout objects must be accessed through a ConnectedLayoutCollection object.

Word Pro: ConnectedLayout class

The connected layout for a connected cell object.

Base Classes

BaseObject\Layout\CellLayout

Derived Classes

None.

Contained by**Usage**

The layout object that results from connecting two or more cells is a ConnectedLayout object. A ConnectedLayout object can only be accessed through the collection of ConnectedLayout objects. For more information on the collection of ConnectedLayout objects, see the [ConnectedLayouts](#) property.

Word Pro: ContentCollection class

A collection of content objects in the foundry of a specific division. Content can include FootnoteTable objects, Formula objects, Graphic objects, GraphicOleObjects, OleObjects, ParallelColumns objects, SuperTable objects, BaseTable objects, TableHeading objects, Table objects, and Text objects.

Base Classes

BaseObject\BaseCollection

Derived Classes

None.

Contained by

Foundry in the Contents Property

Usage

Use this collection to access any of the content objects in the foundry of a specific division.

Word Pro: Content class

Content is an abstract class which provides the basic functionality that is common to all content-related objects. Each content object represents the contents of a particular type of object. For example, a Text object might represent the contents of a page or the prompt in a ClickHere block; a Graphic object might represent the contents of a frame.

In essence, while each content object is tailored to represent the contents of a particular type of object, all content objects share a few common traits. The Content class defines those common traits and each class derived from the Content class inherits those traits as properties.

Base Classes

[BaseObject](#)

Derived Classes

[BaseTable](#)

[FootnoteTable](#)

[Glossary](#)

[Graphic](#)

[GraphicOleObject](#)

[OleObject](#)

[ParallelColumns](#)

[SuperTable](#)

[Table](#)

[TableHeading](#)

[Text](#)

Contained by

[Footnote](#) in the [Content](#) Property

[WPApplication](#) in the [Content](#) Property

Usage

While no object is ever instantiated from this class, two properties use this class as their data type. By using the abstract Content class as the data type for a property, Word Pro can store any content object in that property. Note that WPApplication provides a current context property, called Content, which uses the abstract class as its data type. This allows Word Pro to give you access to the content object for whatever object has the focus. By using the same technique in the Footnote class, Word Pro ensures that you can place any type of content object in your footnotes.

Word Pro: ContextMenuOptions class

Controls the display of context sensitive menus in text, cells, frames, graphics, and parallel columns.

Base Classes

None

Derived Classes

None.

Contained by

ApplicationWindow in the ContextMenuOptions Property

Usage

Allows you to control whether or not context menu items display when the focus is within text, a frame, a table cell, a graphic, or a parallel column. For example, setting the `IsFrameMenuEnabled` property to `False` prevents Word Pro from displaying the Frame menu, when the cursor is within a frame. If this property is set to `False`, you cannot modify a frame by using the InfoBox. Setting this property to `True` allows Word Pro to display context sensitive menus, such as the Frame menu.

Word Pro: DdeLinkCollection class

A collection of DdeLink objects in the DdeLinkManager class.

Base Classes

BaseObject\BaseCollection

Derived Classes

None.

Contained by

DdeLinkManager in the DdeLinks Property

Usage

Use this collection to access any of the DdeLink objects in the DdeLinkManager class.

Word Pro: DdeLinkManager class

A tool that creates and maintains DDE links in a document. DDE links can be created in script via the DDELinkManager. There are also DDE functions (outside the WordPro object model) for maintaining DDE links. The DDELinkManager maintains a collection of DDE link objects.

Base Classes

BaseObject

Derived Classes

None.

Contained by

Division in the DdeLinkManager Property

TextDocument in the DdeLinkManager Property

Usage

Used in conjunction with DdeLink and DdeLinkCollection objects. DdeLinkManager allows you to perform a variety of tasks, such as finding, adding, and removing DDE links between parts of a document. You can use the Paste Special dialog box to paste data that uses different formats. DDE link is used only if OLE fails or is not available.

Note OLE is not supported under OS/2.

Word Pro: DdeLink class

Represents a DDE link in a document. DDE links can be created in script via the DDELinkManager. There are also DDE functions (outside the WordPro object model) for maintaining DDE links. The DDELinkManager maintains a collection of DDE link objects.

Base Classes

BaseObject

Derived Classes

None.

Contained by

Usage

You can link objects to a document in two ways: by DDE link or OLE link. DDE link is used only if OLE fails or is not available. The DdeLink class contains information about a DDE link, including its name, format, server, topic, and so on.

Note OLE is not supported under OS/2.

Word Pro: DivisionCollection class

A collection of division objects in a document or division.

Base Classes

BaseObject\BaseCollection

Derived Classes

None.

Contained by

Division in the Divisions Property

TextDocument in the Divisions Property

WPApplication in the Divisions Property

Usage

Use this collection to access any of the division objects in the TextDocument or Division class.

Word Pro: DivisionInfo class

Information about a division in a document.

Base Classes

BaseObject

Derived Classes

None.

Contained by

BaseContainer in the DivisionInfo Property

Division in the DivisionInfo Property

Marker in the DivisionInfo Property

TextDocument in the DivisionInfo Property

Usage

Word Pro: DivisionOptions class

Division options displayed in the Division Properties dialog box.

Base Classes

BaseObject

Derived Classes

None.

Contained by

Division in the DivisionOptions Property

TextDocument in the DivisionOptions Property

Usage

Word Pro: Division class

A division in a document. A division can contain text, frames, text marked as sections, other divisions with different properties from each other, external files linked to a document, or OLE objects.

Note OLE is not supported under OS/2.

Base Classes

BaseObject

Derived Classes

None.

Contained by

WPApplication in the Division Property

Usage

Word Pro: DocControl class

This class allows you to access a document, assign editing rights, enable password protection, select or change colors that show editor markups, make insertions and deletions, and enable document protection in a division.

Base Classes

BaseObject

Derived Classes

None.

Contained by

Division in the DocControl Property

TextDocument in the DocControl Property

Usage

Setting the properties and methods of this class is equivalent to choosing File - TeamSecurity and doing one of the following: opening a specific file, assigning editing rights, verifying editing rights, creating a password, disabling version review, disabling Notes/FX fields, editing ClickHere Blocks, initiating startup scripts, revealing hidden text, editing protected text, protecting frames and table cells, or displaying all division tabs.

Word Pro: DocInfoFieldCollection class

A collection of DocInfoField objects in the DocInfoFieldManager class.

Base Classes

BaseObject\BaseCollection

Derived Classes

None.

Contained by

DocInfoFieldManager in the Fields Property

Usage

Use this collection to access any of the DocInfoField objects in the DocInfoFieldManager class.

Fields stored in this collection are denoted by a field type value of "Doc. Field". Choose File - Document Properties - Document and click the Fields tab to determine the type of a specific field. Word Pro's default document fields, such as Filename and Path, are not stored in this collection.

Word Pro: DocInfoFieldManager class

A tool for managing DocInfo fields in a document.

Base Classes

BaseObject

Derived Classes

None.

Contained by

DocInfo in the FieldManager Property

Usage

Use this class to access, add, or delete DocInfo fields in a document.

Word Pro: DocInfoField class

Represents the information in a specific DocInfo field.

Base Classes

BaseObject

Derived Classes

None.

Contained by**Usage**

Use this class to access information about a DocInfo field, such as its name or its contents.

Word Pro: DocInfo class

The class that holds all the statistics associated with a document, such as document/version editing information or field name descriptions.

Base Classes

BaseObject

Derived Classes

None.

Contained by

Division in the DocInfo Property

TextDocument in the DocInfo Property

Usage

This class is the container for all DocInfo fields. As a result, you can obtain any general information about a document by using this class. Setting the properties and methods of this class is equivalent to choosing File - Document Properties, choosing Document, and clicking the Fields panel.

Word Pro: Documents class

A collection of text document objects in the Word Pro application.

Base Classes

BaseObject\BaseCollection

Derived Classes

None.

Contained by

Application in the Documents Property

Usage

Use this collection to access any open documents in a WordPro session.

Word Pro: Document class

An abstract class that describes the basic top-level container for data in a Lotus application. Each Lotus application defines its own subclass for Document. In Word Pro, the subclass is TextDocument. You should use TextDocument and its class members when working with Word Pro documents.

Base Classes

BaseObject

Derived Classes

TextDocument

Contained by

None

Usage

In Word Pro, you can open documents and create new documents, using the OpenDocument and CreateDocument methods found in the WPAApplication class.

Word Pro: DocWindowCollection class

A collection of DocWindow objects in the ApplicationWindow class.

Base Classes

BaseObject\BaseCollection

Derived Classes

None.

Contained by

ApplicationWindow in the DocWindows Property

Usage

Use this collection to access any of the DocWindow objects in the ApplicationWindow class.

Word Pro: DocWindow class

DocWindow is the class of the document window.

Base Classes

BaseObjectWindow

Derived Classes

None.

Contained by

Application in the ActiveDocWindow Property

Usage

Word Pro: DropCapContainer class

This language element has not yet been defined. You can check for a newer Help file on the Lotus web site.

Base Classes

FrameContainer

Derived Classes

None.

Contained by

None.

Usage

Word Pro: DropCapLayoutCollection class

This language element is not yet defined.

Base Classes

Unknown

Derived Classes

None.

Contained by

None.

Usage

Word Pro: DropCapLayout class

The drop cap layout for a DropCapContainer object.

Base Classes

BaseObject\Layout\FrameLayout

Derived Classes

Contained by

DropCapContainer in the Layout Property

Usage

Word Pro: EditorCollection class

A collection of editor objects in the EditorManager class.

Base Classes

BaseObject\BaseCollection

Derived Classes

None.

Contained by

EditorManager in the Editors Property

Usage

Use this collection to access any of the editor objects in the EditorManager class.

Word Pro: EditorManager class

A tool for managing editor objects in a document.

Base Classes

BaseObject

Derived Classes

None.

Contained by

Division in the EditorManager Property

TextDocument in the EditorManager Property

Usage

Allows you to add, remove, or access editors in a document.

Word Pro: Editor class

Represents the information associated with a specific editor of a document.

Base Classes

BaseObject

Derived Classes

None.

Contained by

EditorManager in the CurrentEditor Property

Usage

While writing a script, be careful how you change editing options for yourself as the current editor. Carelessly changing access options for the current editor could cause you to accidentally lock yourself out of the document.

Word Pro: EndnoteDivisionGroupNum class

Controls the way endnote numbers display at the end of a division group.

Base Classes

BaseObject\FooterNumOpt

Derived Classes

None.

Contained by

FootnoteOptions in the EndnoteDivisionGroupNum Property

Usage

You can set endnote numbers at the end of a division group by assigning the Reset \$OptionEachDivisiongroup value to the ResetWhen property. This value increases endnote numbers through a division group and resets with the first endnote in the next division group. For more information, see the ResetWhen property.

Word Pro: EndnoteDivisionNum class

Controls the way endnote numbers display at the end of a current division.

Base Classes

BaseObject\FooterNumOpt

Derived Classes

None.

Contained by

FootnoteOptions in the EndnoteDivisionNum Property

Usage

You can set endnote numbers at the end of a current division by assigning the \$ResetWhenOptionEachDivision value to the ResetWhen property. This value increases endnote numbers throughout the division and resets with the first endnote in the next division. For more information, see the ResetWhen property.

Word Pro: EndnoteDocNum class

Controls the way endnote numbers display at the end of a document.

Base Classes

BaseObject\FooterNumOpt

Derived Classes

None.

Contained by

[FootnoteOptions](#) in the [EndnoteDocNum](#) Property

Usage

You can set endnote numbers at the end of a document by assigning the \$ResetWhenOptionEachDoc value to the ResetWhen property. This value increases endnote numbers each time you add a new endnote and continues increasing throughout the document. For more information, see the ResetWhen property.

Word Pro: EndnoteLayoutCollection class

A collection of endnote layouts in the foundry of a specific division, document, or application.

Base Classes

BaseObject\BaseCollection

Derived Classes

None.

Contained by

Foundry in the Endnotes Property

Usage

Use this collection to access any of the endnote objects in the Foundry of a specific division, document, or application.

Word Pro: EndnoteLayout class

The layout for an endnote object.

Base Classes

BaseObject\Layout\TableLayout

Derived Classes

[FootnoteLayout](#)

Contained by

Usage

The default property settings should remain as they are. Changing any endnote layout properties can cause a script not to work properly.

Word Pro: FilterHelper class

Helps a filter convert non-Word Pro file formats to Word Pro file formats.

Base Classes

BaseObject

Derived Classes

None.

Contained by

Filter in the FilterHelper Property

Usage

Word Pro: Filter class

Converts non-Word Pro file formats to a Word Pro file format.

Base Classes

BaseObject

Derived Classes

None.

Contained by

ApplicationWindow in the Filter Property

Usage

Word Pro: FindAndReplace class

The Find & Replace tool in the Word Pro application.

Base Classes

BaseObject

Derived Classes

None.

Contained by

WPAApplication in the FindAndReplace Property

WPAApplication in the TempFindAndReplace Property

Usage

Use the Find & Replace feature to find and replace text, paragraph styles, and special characters in a document. Always runs in a default state in the Word Pro application.

Word Pro: FontMetrics class

Base Classes

BaseObject

Derived Classes

None.

Contained by

Font in the FontMetrics Property

Usage

Word Pro: Font class

All of the font and text style properties associated with an object that has text.

Base Classes

BaseObject

Derived Classes

None.

Contained by

[CharacterStyle](#) in the [Font](#) Property

[ClickHere](#) in the [Font](#) Property

[Editor](#) in the [InsertFont](#) Property

[Editor](#) in the [DeleteFont](#) Property

[FindAndReplace](#) in the [FindFont](#) Property

[FindAndReplace](#) in the [ReplaceFont](#) Property

[Graphic](#) in the [Font](#) Property

[ParagraphStyle](#) in the [Font](#) Property

[RevisionDisplay](#) in the [InsertFont](#) Property

[RevisionDisplay](#) in the [DeleteFont](#) Property

[Text](#) in the [Font](#) Property

[TextMarker](#) in the [Font](#) Property

Usage

Word Pro: FooterLayoutCollection class

A collection of footer layout objects in the foundry of a specific division.

Base Classes

BaseObject\BaseCollection

Derived Classes

None.

Contained by

Foundry in the Footers Property

Usage

Use this collection to access any of the footer layout objects in the foundry of a specific division.

Word Pro: FooterLayout class

The layout for a footer object.

Base Classes

BaseObjectLayout

Derived Classes

None.

Contained by

Usage

Word Pro: FootnoteCollection class

A collection of footnote objects in the foundry of a specific division.

Base Classes

BaseObject\BaseCollection

Derived Classes

None.

Contained by

Foundry in the Footnotes Property

Usage

Use this collection to access any of the footnote objects in the foundry of a specific division.

Word Pro: Footnote class

A footnote object in a document.

Base Classes

BaseObject

Derived Classes

None.

Contained by**Usage**

Word Pro: Accelerators class members

Properties

Application AS WPAApplication class

Description

IsValid

Name

Parent AS BaseObject class

VersionID

Methods

AddAccelerators

DeleteMacroAccelerator

RemovePersistentAccelerators

Events

None

Word Pro: Alignment class members

Properties

AlignmentChar

AlignmentType

Application AS WPAApplication class

Description

IsValid

Name

Parent AS BaseObject class

Position

VersionID

Methods

RevertToStyle

Events

None

Word Pro: Amikake class members

Properties

AmikakeType

Application AS WPAApplication class

Background AS Background class

Description

IsAmikake

IsValid

Name

Parent AS BaseObject class

VersionID

Methods

RevertToStyle

Events

None

Word Pro: ApplicationWindow class members

Properties

[Accelerators](#) AS [Accelerators class](#)
[Active](#)
[ActiveDocument](#) AS [TextDocument class](#)
[Application](#) AS [WPAApplication class](#)
[Caption](#)
[ContextMenuOptions](#) AS [ContextMenuOptions class](#)
[CurrentRunningScriptName](#)
[CurrentRunningScriptPath](#)
[Description](#)
[DocWindows](#) AS [DocWindowCollection class](#)
[Filter](#) AS [Filter class](#)
[FreeMenus](#) AS [MenuItem class](#)
[Height](#)
[HorzRuler](#) AS [Ruler class](#)
[Hwnd](#)
[IconBarManager](#) AS [IconBarManager class](#)
[IsValid](#)
[Left](#)
[LwpMenuBar](#) AS [MenuItem class](#)
[Macro](#) AS [Macro class](#)
[Name](#)
[Parent](#) AS [BaseObject class](#)
[ReviewVersions](#) AS [ReviewVersions class](#)
[RightMouseMenus](#) AS [MenuItem class](#)
[SectionTabs](#) AS [SectionTabs class](#)
[SetTabsDialog](#) AS [SetTabsDialog class](#)
[StatusBar](#) AS [StatusBar class](#)
[StatusBarVisible](#)
[TitleBarVisible](#)
[Top](#)
[UserInterfacePrefs](#) AS [UserInterfacePrefs class](#)
[VersionID](#)
[VertRuler](#) AS [Ruler class](#)
[Visible](#)
[Width](#)

Methods

[Cascade](#)
[Close](#)
[DarkMode](#)
[InternetExtraFile](#)
[Maximize](#)
[Minimize](#)
[Move](#)
[Open](#)

Play

Resize

Restore

SaveUserDefaults

Tile

Update

UpdateAllMenus

Events

Moved

Word Pro: Application class members

Properties

[ActiveDocument](#) AS [TextDocument](#) class
[ActiveDocWindow](#) AS [DocWindow](#) class
[Application](#) AS [WPAApplication](#) class
[ApplicationWindow](#) AS [ApplicationWindow](#) class
[DefaultFilePath](#)
[Description](#)
[Documents](#) AS [Documents](#) class
[FullName](#)
[Interactive](#)
[IsValid](#)
[Language](#)
[Location](#)
[Name](#)
[Parent](#) AS [BaseObject](#) class
[Path](#)
[VersionID](#)
[Visible](#)

Methods

[GetEnum](#)
[NewDocument](#)
[OpenDocument](#)
[Quit](#)
[UseLSX](#)

Events

[DocumentCreated](#)
[DocumentOpen](#)
[DocumentOpened](#)
[Quit](#)

Word Pro: AppViewPrefs class members

Properties

Application AS WPAApplication class

Description

IsValid

MarginColor AS Color class

Name

Parent AS BaseObject class

SelectionBorderColor1 AS Color class

SelectionBorderColor2 AS Color class

SelectionBorderColor3 AS Color class

SpellColor AS Color class

SpellFocusedColor AS Color class

VersionID

Methods

None

Events

None

Word Pro: Attributes class members

Properties

Application AS WPAApplication class

BaseLineOffset

Description

HiddenMode

HideOutlineLevels

HighLightMode

IsDoubleWordError

IsGrammarError

IsHiddenMark

IsMisspelled

IsValid

Name

NoHyphenate

Parent AS BaseObject class

ProtectedMode

SkipWordMode

VersionID

Methods

Clear

RevertToStyle

Events

None

Word Pro: AutoRunMacro class members

Properties

Application AS WPAApplication class

CloseDocMacroName

Description

IsValid

Name

NewDocMacroName

OpenDocMacroName

Parent AS BaseObject class

RunOnCloseDoc

RunOnNewDoc

RunOnOpenDoc

VersionID

Methods

None

Events

None

Word Pro: Background class members

Properties

Application AS WPAApplication class

BackColor AS Color class

BackColorIndex

Color AS Color class

Description

ForeColorIndex

IsValid

Name

Parent AS BaseObject class

Pattern

Shape

VersionID

Methods

None

Events

None

Word Pro: BagCollection class members

Properties

Application AS WPAApplication class

Count

Description

IsValid

Name

Parent AS BaseObject class

VersionID

Methods

IsEmpty

Item

Events

None

Word Pro: Bag class members

Properties

Application AS WPAApplication class

Description

IsValid

Length

Name

Parent AS BaseObject class

VersionID

Methods

Clear

DeleteBag

Read

Reset

Write

Events

None

Word Pro: BaseCollection class members

Properties

Application AS WPAApplication class

Count

Description

IsValid

Name

Parent AS BaseObject class

VersionID

Methods

IsEmpty

Events

None

Word Pro: BaseContainer class members

Properties

[AbsoluteTextOrientation](#)
[Application](#) AS [WPAApplication class](#)
[ClientHeight](#)
[ClientWidth](#)
[ContentHeight](#)
[ContentName](#)
[ContentWidth](#)
[Description](#)
[DisplayablePageNum](#)
[DivisionInfo](#) AS [DivisionInfo class](#)
[DivisionName](#)
[Height](#)
[IsInOleDivision](#)
[IsValid](#)
[Layout](#) AS [Layout class](#)
[MaxContentHeight](#)
[MaxContentWidth](#)
[Name](#)
[NumContainers](#)
[PageNum](#)
[Parent](#) AS [BaseObject class](#)
[PositionXOnPage](#)
[PositionYOnPage](#)
[Presentation](#) AS [Presentation class](#)
[RelativePageNum](#)
[SectionName](#)
[TextOrientation](#)
[VersionID](#)
[Width](#)

Methods

[Abandon](#)
[AddContainer](#)
[Adopt](#)
[Anchor](#)
[Backward](#)
[ConnectContainer](#)
[DeleteContainer](#)
[Disconnect](#)
[DoesMultiCellPaste](#)
[EditWrap](#)
[Ending](#)
[Forward](#)
[GetObjectList](#)
[GetPasteFormatCategories](#)

GoToContainer

Hide

IsPointWithin

LinkContainers

ShowContainers

Start

UnlinkContainers

Events

None

Word Pro: BaseObject class members

Properties

Application AS WPAApplication class

Description

IsValid

Name

Parent AS BaseObject class

VersionID

Methods

None

Events

None

Word Pro: BaseTable class members

Properties

Application AS WPAApplication class
CanEmbed
CellLayouts AS StringCollection class
ColumnLayouts AS StringCollection class
ContentType
CurrentCell AS CellLayout class
CurrentColumn AS Layout class
CurrentRow AS RowLayout class
DefCellStyleName
DefColWidth
DefRowHeight
Description
EndingColOfSelection
EndingRowOfSelection
IsAutoGrow
IsEmpty
IsParagraphNumberingDown
IsReplaceable
IsResetParagraphNumber
IsSizingViaMouse
IsValid
Layout AS Layout class
MaxBottomBorder
MaxBottomGutter
MaxLeftBorder
MaxLeftGutter
MaxNumColsAllowed
MaxNumRowsAllowed
MaxRightBorder
MaxRightGutter
MaxSplitCols
MaxSplitRows
MaxTopBorder
MaxTopGutter
Name
NumCols
NumRows
Parent AS BaseObject class
RowLayouts AS StringCollection class
SelectionType
SingleCellSelected
StartingColOfSelection
StartingColStringOfSelection
StartingRowOfSelection

TableFill AS TableFill class

TableLine AS TableLine class

VersionID

Methods

CellLayout

Connect

Copy

DeleteTable

DisconnectCells

DoesMarkerNameMatch

FindCellLayout

GetMarkerName

GoToTableCell

InsertRowOrColumn

Mark

SelectTableItem

Split

Events

None

Word Pro: BetweenLines class members

Properties

AllBorders AS Border class

Application AS WPAApplication class

BottomBorder AS Border class

Description

IsValid

LeftBorder AS Border class

LinePlacement

LineValid

Name

Parent AS BaseObject class

RightBorder AS Border class

TopBorder AS Border class

VersionID

Methods

None

Events

None

Word Pro: BookmarkCollection class members

Properties

Application AS WPAApplication class

Count

Description

IsValid

Name

Parent AS BaseObject class

VersionID

Methods

IsEmpty

Item

Events

None

Word Pro: BookmarkManager class members

Properties

Application AS WPAApplication class

Bookmarks AS BookmarkCollection class

BookmarksByMarkerName AS BookmarkCollection class

Description

IsValid

Name

Parent AS BaseObject class

VersionID

Methods

AddBookmark

Find

GetUniqueName

RemoveBookmark

Events

None

Word Pro: Bookmark class members

Properties

Application AS WPAApplication class

Description

IsExportedToNotesFX

IsValid

MarkerName

Name

Parent AS BaseObject class

VersionID

Methods

None

Events

None

Word Pro: BorderLines class members

Properties

[AllBorders](#) AS [Border class](#)

[Application](#) AS [WPAApplication class](#)

[BottomBorder](#) AS [Border class](#)

[Description](#)

[IsValid](#)

[LeftBorder](#) AS [Border class](#)

[LinePlacement](#)

[LineValid](#)

[Name](#)

[Parent](#) AS [BaseObject class](#)

[RightBorder](#) AS [Border class](#)

[TopBorder](#) AS [Border class](#)

[VersionID](#)

Methods

None

Events

None

Word Pro: Border class members

Properties

Application AS WPAApplication class

BackColorIndex

Color AS Color class

Description

ForeColorIndex

IsValid

Name

NameOfExternalBorder

Parent AS BaseObject class

Pattern

Style

VersionID

Width

WidthInTwips

Methods

None

Events

None

Word Pro: Breaks class members

Properties

Application AS WPAApplication class

Description

IsColumnBreakAfter

IsColumnBreakBefore

IsPageBreakAfter

IsPageBreakBefore

IsPageBreakWithin

IsValid

KeepWithNext

KeepWithPrev

Name

NextStyleName

Parent AS BaseObject class

UseNextStyle

VersionID

Methods

RevertToStyle

Events

None

Word Pro: Bullet class members

Properties

Application AS WPAApplication class

Description

Editable

IsValid

Name

Parent AS BaseObject class

RightAlign

SilverBullet AS SilverBullet class

Skipped

Text AS Text class

Valid

VersionID

Methods

RevertToStyle

Events

None

Word Pro: CellCollection class members

Properties

Application AS WPAApplication class

Count

Description

IsValid

Name

Parent AS BaseObject class

VersionID

Methods

IsEmpty

Item

Events

None

Word Pro: CellContainer class members

Properties

[AbsoluteTextOrientation](#)
[Application](#) AS [WPAApplication class](#)
[CellLayout](#) AS [CellLayout class](#)
[ClientHeight](#)
[ClientWidth](#)
[ContentHeight](#)
[ContentName](#)
[ContentWidth](#)
[Description](#)
[DisplayablePageNum](#)
[DivisionInfo](#) AS [DivisionInfo class](#)
[DivisionName](#)
[Height](#)
[IsInOleDivision](#)
[IsValid](#)
[Layout](#) AS [Layout class](#)
[MaxContentHeight](#)
[MaxContentWidth](#)
[Name](#)
[NumContainers](#)
[PageNum](#)
[Parent](#) AS [BaseObject class](#)
[PositionXOnPage](#)
[PositionYOnPage](#)
[Presentation](#) AS [Presentation class](#)
[RelativePageNum](#)
[SectionName](#)
[TextOrientation](#)
[VersionID](#)
[Width](#)

Methods

[Abandon](#)
[AddContainer](#)
[Adopt](#)
[Anchor](#)
[Backward](#)
[ConnectContainer](#)
[DeleteContainer](#)
[Disconnect](#)
[DoesMultiCellPaste](#)
[EditWrap](#)
[Ending](#)
[Forward](#)
[GetObjectList](#)

GetPasteFormatCategories

GoToContainer

Hide

IsPointWithin

linkcontainers

RevertToStyle

SetStyle

ShowContainers

Start

unlinkcontainers

Events

None

Word Pro: CellEngine class members

Properties

Application AS WPAApplication class

Description

IsValid

Name

Parent AS BaseObject class

VersionID

Methods

GetFormula

SetFormula

Events

None

Word Pro: CellGroupLayout class members

Properties

[AbsoluteOn](#)

[AbsoluteXPos](#)

[AbsoluteYPos](#)

[Accelerator](#)

[AllowResizeWhenCrop](#)

[AmtTether](#)

[AmtToRotateContent](#)

[AmtToTetherFrom](#)

[Application](#) AS [WPAApplication class](#)

[Background](#) AS [Background class](#)

[BaseLineOffset](#)

[BinName](#)

[BorderLines](#) AS [BorderLines class](#)

[BorderOffset](#)

[BottomExternalMargin](#)

[Center](#)

[ChildLayouts](#) AS [LayoutCollection class](#)

[ClassName](#)

[ColumnBalance](#)

[ColumnGap](#)

[ConditionType](#)

[Content](#)

[ContentName](#)

[ContentStyleName](#)

[Definition](#)

[Description](#)

[DirectionDown](#)

[DirectionLeft](#)

[DirectionRight](#)

[DirectionUp](#)

[DropCapPosition](#)

[EditorName](#)

[Footer](#) AS [Layout class](#)

[GridDistance](#)

[GridType](#)

[Gutter](#) AS [Gutter class](#)

[Header](#) AS [Layout class](#)

[Height](#)

[IsBreakable](#)

[IsChildSpannable](#)

[IsCollapsed](#)

[IsCollapsible](#)

[IsColumnBreakable](#)

[IsComplex](#)

[IsConnected](#)
[IsErrorChecking](#)
[IsExpandDown](#)
[IsExpandLeft](#)
[IsExpandRight](#)
[IsExpandUp](#)
[IsIndexMarkAble](#)
[IsLocal](#)
[IsLocked](#)
[IsMarginSameAsParent](#)
[IsNotCopyable](#)
[IsNotGroupable](#)
[IsNoUICommAllowed](#)
[IsOverridden](#)
[IsOverride](#)
[IsPageBreak](#)
[IsPartOfGroup](#)
[IsPrintable](#)
[IsProtected](#)
[IsScrollable](#)
[IsSingleClickEntry](#)
[IsSizable](#)
[IsSnapTo](#)
[IsStyle](#)
[IsTableHeading](#)
[IsTOC](#)
[IsTOCMarkAble](#)
[IsValid](#)
[Join AS Join class](#)
[Justifiable](#)
[Layer AS Layout class](#)
[LayerName](#)
[LeaderDotType](#)
[LeftExternalMargin](#)
[LeftPage AS Layout class](#)
[LeftTopCellId](#)
[LineLocation](#)
[LinkFrame](#)
[MaintainAspectRatio](#)
[MarginBottom](#)
[MarginLeft](#)
[MarginRight](#)
[MarginTop](#)
[MasterName](#)
[MinBottomMargin](#)
[MinHeight](#)
[MinLeftMargin](#)

[MinRightMargin](#)
[MinTopMargin](#)
[Name](#)
[NameBasedOnStyle](#)
[NumberOfLines](#)
[NumCols](#)
[NumColsSpannedOneCell](#)
[NumericFormat](#) AS [NumericFormat class](#)
[NumRowsSpannedOneCell](#)
[PageToUseLayoutOn](#)
[PaperName](#)
[Parent](#) AS [BaseObject class](#)
[RelativeType](#)
[RelativeXDistance](#)
[RelativeYDistance](#)
[RevisionType](#)
[RightExternalMargin](#)
[RightPage](#) AS [Layout class](#)
[ScaleHeight](#)
[ScaleMode](#)
[ScalePercentage](#)
[ScaleWidth](#)
[SelectType](#)
[Shadow](#) AS [Shadow class](#)
[Span](#)
[Style](#) AS [Layout class](#)
[StyleExceptions](#)
[TabRack](#) AS [TabRack class](#)
[TextOrient](#)
[Tile](#)
[TopExternalMargin](#)
[TopLeftCellRowId](#)
[UseFooter](#)
[UseHeader](#)
[UsePrinterSettings](#)
[UseWhen](#) AS [UseWhen class](#)
[VersionID](#)
[VertAlign](#)
[WasDeletedInRevMarkMode](#)
[WasInsertedInRevMarkMode](#)
[Where](#)
[Width](#)
[WPDataSets](#) AS [WPDataSetCollection class](#)
[WrapType](#)
[XOffset](#)
[XPosition](#)
[YOffset](#)

YPosition

Methods

AddChildToLayout

Backward

CreateLayer

DeleteContents

DeleteLayout

DoesMarkerNameMatch

FindClass

Forward

GetMarkerName

GetNamedProperty

GoToLayout

HasNamedProperty

ImportWatermarkGraphic

Mark

MirrorPage

MoveToBack

MoveToFront

RegisterWPDataSet

RemoveChildFromLayout

RemoveNamedProperty

RevisionAcceptLayoutChange

RevisionCancelLayoutChange

SetAllMargins

SetMinimumOrigin

SetNamedProperty

SetPaperNameAndUpdateSize

SetupForCropping

UnregisterWPDataSet

Update

Events

EnterLayout

KeyStroke

MouseDown

MouseUp

Word Pro: CellLayoutCollection class members

Properties

Application AS WPAApplication class

Count

Description

IsValid

Name

Parent AS BaseObject class

VersionID

Methods

IsEmpty

Item

Events

None

Word Pro: CellLayout class members

Properties

[AbsoluteOn](#)

[AbsoluteXPos](#)

[AbsoluteYPos](#)

[Accelerator](#)

[AllowResizeWhenCrop](#)

[AmtTether](#)

[AmtToRotateContent](#)

[AmtToTetherFrom](#)

[Application](#) AS [WPAApplication class](#)

[Background](#) AS [Background class](#)

[BaseLineOffset](#)

[BinName](#)

[BorderLines](#) AS [BorderLines class](#)

[BorderOffset](#)

[BottomExternalMargin](#)

[Center](#)

[ChildLayouts](#) AS [LayoutCollection class](#)

[ClassName](#)

[ColumnBalance](#)

[ColumnGap](#)

[ConditionType](#)

[content](#)

[ContentName](#)

[ContentStyleName](#)

[Definition](#)

[Description](#)

[DirectionDown](#)

[DirectionLeft](#)

[DirectionRight](#)

[DirectionUp](#)

[DropCapPosition](#)

[EditorName](#)

[Footer](#) AS [Layout class](#)

[Formula](#)

[GetValue](#)

[GridDistance](#)

[GridType](#)

[Gutter](#) AS [Gutter class](#)

[Header](#) AS [Layout class](#)

[Height](#)

[IsBadReference](#)

[IsBreakable](#)

[IsChildSpannable](#)

[IsCollapsed](#)

[IsCollapsible](#)
[IsColumnBreakable](#)
[IsComplex](#)
[IsConnected](#)
[IsErrorChecking](#)
[IsExpandDown](#)
[IsExpandLeft](#)
[IsExpandRight](#)
[IsExpandUp](#)
[IsIndexMarkAble](#)
[IsLocal](#)
[IsLocked](#)
[IsMarginSameAsParent](#)
[IsNotCopyable](#)
[IsNotGroupable](#)
[IsNoUICommAllowed](#)
[IsOverridden](#)
[IsOverride](#)
[IsPageBreak](#)
[IsPartOfGroup](#)
[IsPrintable](#)
[IsProtected](#)
[IsScrollable](#)
[IsSingleClickEntry](#)
[IsSizable](#)
[IsSnapTo](#)
[IsStyle](#)
[IsTableHeading](#)
[IsTOC](#)
[IsTOCMarkAble](#)
[IsValid](#)
[Join AS Join class](#)
[Justifiable](#)
[Layer AS Layout class](#)
[LayerName](#)
[LeaderDotType](#)
[LeftExternalMargin](#)
[LeftPage AS Layout class](#)
[LeftTopCellId](#)
[LineLocation](#)
[LinkFrame](#)
[MaintainAspectRatio](#)
[MarginBottom](#)
[MarginLeft](#)
[MarginRight](#)
[MarginTop](#)
[MasterName](#)

[MinBottomMargin](#)
[MinHeight](#)
[MinLeftMargin](#)
[MinRightMargin](#)
[MinTopMargin](#)
[Name](#)
[NameBasedOnStyle](#)
[NumberOfLines](#)
[NumCols](#)
[NumColsSpannedOneCell](#)
[NumericFormat](#) AS [NumericFormat class](#)
[NumRowsSpannedOneCell](#)
[PageToUseLayoutOn](#)
[PaperName](#)
[Parent](#) AS [BaseObject class](#)
[RelativeType](#)
[RelativeXDistance](#)
[RelativeYDistance](#)
[RevisionType](#)
[RightExternalMargin](#)
[RightPage](#) AS [Layout class](#)
[ScaleHeight](#)
[ScaleMode](#)
[ScalePercentage](#)
[ScaleWidth](#)
[SelectType](#)
[Shadow](#) AS [Shadow class](#)
[Span](#)
[Style](#) AS [Layout class](#)
[StyleExceptions](#)
[TabRack](#) AS [TabRack class](#)
[TextOrient](#)
[Tile](#)
[TopExternalMargin](#)
[TopLeftCellRowId](#)
[UseFooter](#)
[UseHeader](#)
[UsePrinterSettings](#)
[UseWhen](#) AS [UseWhen class](#)
[VersionID](#)
[VertAlign](#)
[WasDeletedInRevMarkMode](#)
[WasInsertedInRevMarkMode](#)
[Where](#)
[Width](#)
[WPDataSets](#) AS [WPDataSetCollection class](#)
[WrapType](#)

XOffset
XPosition
YOffset
YPosition

Methods

AddChildToLayout
Backward
createlayer
DeleteContents
DeleteLayout
DoesMarkerNameMatch
FindClass
Forward
GetMarkerName
GetNamedProperty
GoToLayout
HasNamedProperty
ImportWatermarkGraphic
Mark
MirrorPage
MoveToBack
MoveToFront
RegisterWPDataset
RemoveChildFromLayout
RemoveNamedProperty
RevisionAcceptLayoutChange
RevisionCancelLayoutChange
setallmargins
SetMinimumOrigin
SetNamedProperty
SetPaperNameAndUpdateSize
SetupForCropping
UnregisterWPDataSet
Update
ValidateValue

Events

EnterLayout
KeyStroke
MouseDown
MouseUp

Word Pro: CharacterBorder class members

Properties

Application AS WPAApplication class

BorderLines AS BorderLines class

Description

IsBorder

IsValid

MarginBottom

MarginLeft

MarginRight

MarginTop

Name

Parent AS BaseObject class

VersionID

WidthAbove

WidthBelow

Methods

RevertToStyle

Events

None

Word Pro: CharacterSet class members

Properties

Application AS WPAApplication class

CharSet

Description

IsValid

LeaderDotDashChar

LeaderDotDotChar

LeaderDotUnderscoreChar

Name

ParagraphSymbolChar

Parent AS BaseObject class

TabSymbolChar

VersionID

Methods

None

Events

None

Word Pro: CharacterStyleCollection class members

Properties

Application AS WPAApplication class

Count

Description

IsValid

Name

Parent AS BaseObject class

VersionID

Methods

IsEmpty

Item

Events

None

Word Pro: CharacterStyle class members

Properties

[Accelerator](#)

[Amikake](#) AS [Amikake class](#)

[Application](#) AS [WPAApplication class](#)

[AttrStyleName](#)

[CharacterBorder](#) AS [CharacterBorder class](#)

[Definition](#)

[Description](#)

[FaceStyleName](#)

[Font](#) AS [Font class](#)

[FontStyleName](#)

[IsLocal](#)

[IsPrivate](#)

[IsTemp](#)

[IsValid](#)

[Language](#) AS [Language class](#)

[Name](#)

[Parent](#) AS [BaseObject class](#)

[SizeStyleName](#)

[StyleName](#)

[TextAttributes](#) AS [Attributes class](#)

[Type](#)

[VersionID](#)

[WPDataSets](#) AS [WPDataSetCollection class](#)

Methods

[GetNamedProperty](#)

[HasNamedProperty](#)

[RegisterWPDataSet](#)

[RemoveNamedProperty](#)

[SetNamedProperty](#)

[UnregisterWPDataSet](#)

[Update](#)

Events

None

Word Pro: ClickHereCollection class members

Properties

Application AS WPAApplication class

Count

Description

IsValid

Name

Parent AS BaseObject class

VersionID

Methods

IsEmpty

Item

Events

None

Word Pro: ClickHere class members

Properties

[Action](#)

[Alignment AS Alignment class](#)

[AllowListEdit](#)

[AllowListMultiValues](#)

[Amikake AS Amikake class](#)

[Application AS WPAApplication class](#)

[AtBeginning](#)

[AtBeginningOfLine](#)

[AtBeginningOfObject](#)

[AtBeginningOfParagraph](#)

[AtBeginningOfWord](#)

[AtEnd](#)

[AtEndOfLine](#)

[AtEndOfObject](#)

[AtEndOfParagraph](#)

[AtEndOfWord](#)

[Attributes AS Attributes class](#)

[Breaks AS Breaks class](#)

[Bullet AS Bullet class](#)

[CharacterBorder AS CharacterBorder class](#)

[CharacterStyle AS CharacterStyle class](#)

[CharacterStyleName](#)

[CodePage](#)

[ColumnNumber](#)

[ColumnWidth](#)

[CurrentLanguage](#)

[Description](#)

[DivisionInfo AS DivisionInfo class](#)

[EffectiveColumnWidth](#)

[FieldType](#)

[First](#)

[Font AS Font class](#)

[HasLocalTabs](#)

[HelpText](#)

[Indent AS Indent class](#)

[IsChanged](#)

[IsEmpty](#)

[IsInBulletEditMode](#)

[IsLocked](#)

[IsMarkerValid](#)

[IsOverridden](#)

[IsParagraphParent](#)

[IsPrompting](#)

[IsRegistered](#)

[IsRevisionMark](#)
[IsValid](#)
[Kinsoku AS Kinsoku class](#)
[Language AS Language class](#)
[Last](#)
[LastEditorName](#)
[Layout AS Layout class](#)
[LayoutName](#)
[MaintainEditor](#)
[MarkerClass](#)
[MirrorName](#)
[MultiCompareParaTag](#)
[MultiCompareParaTagSet](#)
[Name](#)
[NextClickHere](#)
[NextText](#)
[NormalParagraph](#)
[Numbering AS Numbering class](#)
[NumberOfRevisions](#)
[NumCharsInParagraph](#)
[NumCols](#)
[NumRows](#)
[ObjectType](#)
[PageNumber](#)
[PageNumberAsText](#)
[ParagraphBackground AS ParagraphBackground class](#)
[ParagraphBorder AS ParagraphBorder class](#)
[ParagraphHasDropCap](#)
[ParagraphHasText](#)
[ParagraphStyle AS ParagraphStyle class](#)
[ParagraphStyleName](#)
[Parent AS BaseObject class](#)
[Partial](#)
[PositionXInContainer](#)
[PositionXOnPage](#)
[PositionYInContainer](#)
[PositionYOnPage](#)
[PreviousClickHere](#)
[Prompt AS Text class](#)
[PromptHidden](#)
[RelativeIndent AS RelativeIndent class](#)
[RenderedPageNumber](#)
[ReturnExits](#)
[RevisionMark AS Revision class](#)
[SectionName](#)
[SectionUserName](#)
[SelectionHidden](#)

[SelectionType](#)
[ShowMarks](#)
[Spacing AS Spacing class](#)
[StartColumns](#)
[StartRow](#)
[StateID](#)
[StyleExceptions](#)
[TabExits](#)
[TabOrder](#)
[TabRack AS TabRack class](#)
[TextMarkersInSelection AS StringCollection class](#)
[TextViewAttributes](#)
[UsesHelp](#)
[VersionID](#)
[WPDataSets AS WPDataSetCollection class](#)

Methods

[Add](#)
[AdjustShade](#)
[Backspace](#)
[Backward](#)
[CalculateSmartLevels](#)
[CanRepeatClickHere](#)
[Clear](#)
[CloseObject](#)
[ContractOutlineLevel](#)
[DeleteChars](#)
[DeleteContents](#)
[DeleteMarker](#)
[Demote](#)
[Deselect](#)
[DoesMultiCellPaste](#)
[Embed](#)
[ExpandOutline](#)
[Find](#)
[Forward](#)
[GetContents](#)
[GetCopyFormatCategories](#)
[GetCount](#)
[GetCurrentMarkerName](#)
[GetMarkedText](#)
[GetMisspelledWord](#)
[GetNamedProperty](#)
[getparagraphnumber](#)
[GetParagraphNumberString](#)
[GetParaNumber](#)
[GetPasteFormatCategories](#)

GetPosition
GetSpellStatus
GetSpellUserDictStatus
GetText
GetWordMisspelled
GoTo
HasNamedProperty
HideCaretAndSelection
IndexAll
InsertBreak
InsertDocInfo
InsertHardSpace
InsertMarker
InsertNumber
InsertPageNumber
InsertSpecialTab
InsertTab
InsertText
InternalCopy
InternalCut
InternalPaste
IsMarkerEqualToSelection
IsPointWithin
Mark
MorphSelectionToTable
MoveDown
MoveParagraph
MoveToEnd
MoveToStart
MoveUp
Next
NextToObject
OpenObject
Previous
Promote
registerwpdataSet
Remove
RemoveNamedProperty
Replace
ReplaceContents
RevertToStyle
RevisionAccept
RevisionCancel
Select
SetCursorPosition
SetNamedProperty
SetStyle

Shade

ShowCaretAndSelection

ShowCursor

Skip

SortParagraphs

SpellWord

SplitParagraph

SRReplace

UnregisterWPDataSet

Events

EnterClickHere

ExitClickHere

Word Pro: Color class members

Properties

Application AS WPAApplication class

Blue

Description

Green

IsValid

Name

Override

Parent AS BaseObject class

Red

VersionID

Methods

GetRGB

RevertToStyle

SetRGB

Events

None

Word Pro: ColumnGroupLayout class members

Properties

[AbsoluteOn](#)

[AbsoluteXPos](#)

[AbsoluteYPos](#)

[Accelerator](#)

[AllowResizeWhenCrop](#)

[AmtTether](#)

[AmtToRotateContent](#)

[AmtToTetherFrom](#)

[Application](#) AS [WPAApplication class](#)

[Background](#) AS [Background class](#)

[BaseLineOffset](#)

[BinName](#)

[BorderLines](#) AS [BorderLines class](#)

[BorderOffset](#)

[BottomExternalMargin](#)

[Center](#)

[ChildLayouts](#) AS [LayoutCollection class](#)

[ClassName](#)

[ColumnBalance](#)

[ColumnGap](#)

[ConditionType](#)

[Content](#)

[ContentName](#)

[ContentStyleName](#)

[Definition](#)

[Description](#)

[DirectionDown](#)

[DirectionLeft](#)

[DirectionRight](#)

[DirectionUp](#)

[DropCapPosition](#)

[EditorName](#)

[Footer](#) AS [Layout class](#)

[GridDistance](#)

[GridType](#)

[Gutter](#) AS [Gutter class](#)

[Header](#) AS [Layout class](#)

[Height](#)

[IsBreakable](#)

[IsChildSpannable](#)

[IsCollapsed](#)

[IsCollapsible](#)

[IsColumnBreakable](#)

[IsComplex](#)

IsConnected
IsErrorChecking
IsExpandDown
IsExpandLeft
IsExpandRight
IsExpandUp
IsIndexMarkAble
IsLocal
IsLocked
IsMarginSameAsParent
IsNotCopyable
IsNotGroupable
IsNoUICommAllowed
IsOverridden
IsOverride
IsPageBreak
IsPartOfGroup
IsPrintable
IsProtected
IsScrollable
IsSingleClickEntry
IsSizable
IsSnapTo
IsStyle
IsTableHeading
IsTOC
IsTOCMarkAble
IsValid
Join AS Join class
Justifiable
Layer AS Layout class
LayerName
LeaderDotType
LeftExternalMargin
LeftPage AS Layout class
LeftTopCellId
LineLocation
LinkFrame
MaintainAspectRatio
MarginBottom
MarginLeft
MarginRight
MarginTop
MasterName
MinBottomMargin
MinHeight
MinLeftMargin

[MinRightMargin](#)
[MinTopMargin](#)
[Name](#)
[NameBasedOnStyle](#)
[NumberOfLines](#)
[NumCols](#)
[NumColsSpannedOneCell](#)
[NumericFormat](#) AS [NumericFormat class](#)
[NumRowsSpannedOneCell](#)
[PageToUseLayoutOn](#)
[PaperName](#)
[Parent](#) AS [BaseObject class](#)
[RelativeType](#)
[RelativeXDistance](#)
[RelativeYDistance](#)
[RevisionType](#)
[RightExternalMargin](#)
[RightPage](#) AS [Layout class](#)
[ScaleHeight](#)
[ScaleMode](#)
[ScalePercentage](#)
[ScaleWidth](#)
[SelectType](#)
[Shadow](#) AS [Shadow class](#)
[Span](#)
[Style](#) AS [Layout class](#)
[StyleExceptions](#)
[TabRack](#) AS [TabRack class](#)
[TextOrient](#)
[Tile](#)
[TopExternalMargin](#)
[TopLeftCellRowId](#)
[UseFooter](#)
[UseHeader](#)
[UsePrinterSettings](#)
[UseWhen](#) AS [UseWhen class](#)
[VersionID](#)
[VertAlign](#)
[WasDeletedInRevMarkMode](#)
[WasInsertedInRevMarkMode](#)
[Where](#)
[Width](#)
[WPDataSets](#) AS [WPDataSetCollection class](#)
[WrapType](#)
[XOffset](#)
[XPosition](#)
[YOffset](#)

YPosition

Methods

AddChildToLayout

Backward

CreateLayer

DeleteContents

DeleteLayout

DoesMarkerNameMatch

FindClass

Forward

GetMarkerName

GetNamedProperty

GoToLayout

HasNamedProperty

ImportWatermarkGraphic

Mark

MirrorPage

MoveToBack

MoveToFront

RegisterWPDataSet

RemoveChildFromLayout

RemoveNamedProperty

RevisionAcceptLayoutChange

RevisionCancelLayoutChange

SetAllMargins

SetMinimumOrigin

SetNamedProperty

SetPaperNameAndUpdateSize

SetupForCropping

UnregisterWPDataSet

Update

Events

EnterLayout

KeyStroke

MouseDown

MouseUp

Word Pro: ConnectedLayoutCollection class members

Properties

Application AS WPAApplication class

Count

Description

IsValid

Name

Parent AS BaseObject class

VersionID

Methods

IsEmpty

Item

Events

None

Word Pro: ConnectedLayout class members

Properties

[AbsoluteOn](#)

[AbsoluteXPos](#)

[AbsoluteYPos](#)

[Accelerator](#)

[AllowResizeWhenCrop](#)

[AmtTether](#)

[AmtToRotateContent](#)

[AmtToTetherFrom](#)

[Application](#) AS [WPAApplication class](#)

[Background](#) AS [Background class](#)

[BaseLineOffset](#)

[BinName](#)

[BorderLines](#) AS [BorderLines class](#)

[BorderOffset](#)

[BottomExternalMargin](#)

[Center](#)

[ChildLayouts](#) AS [LayoutCollection class](#)

[ClassName](#)

[ColumnBalance](#)

[ColumnGap](#)

[ConditionType](#)

[content](#)

[ContentName](#)

[ContentStyleName](#)

[Definition](#)

[Description](#)

[DirectionDown](#)

[DirectionLeft](#)

[DirectionRight](#)

[DirectionUp](#)

[DropCapPosition](#)

[EditorName](#)

[Footer](#) AS [Layout class](#)

[Formula](#)

[GetValue](#)

[GridDistance](#)

[GridType](#)

[Gutter](#) AS [Gutter class](#)

[Header](#) AS [Layout class](#)

[Height](#)

[IsBadReference](#)

[IsBreakable](#)

[IsChildSpannable](#)

[IsCollapsed](#)

[IsCollapsible](#)
[IsColumnBreakable](#)
[IsComplex](#)
[IsConnected](#)
[IsErrorChecking](#)
[IsExpandDown](#)
[IsExpandLeft](#)
[IsExpandRight](#)
[IsExpandUp](#)
[IsIndexMarkAble](#)
[IsLocal](#)
[IsLocked](#)
[IsMarginSameAsParent](#)
[IsNotCopyable](#)
[IsNotGroupable](#)
[IsNoUICommAllowed](#)
[IsOverridden](#)
[IsOverride](#)
[IsPageBreak](#)
[IsPartOfGroup](#)
[IsPrintable](#)
[IsProtected](#)
[IsScrollable](#)
[IsSingleClickEntry](#)
[IsSizable](#)
[IsSnapTo](#)
[IsStyle](#)
[IsTableHeading](#)
[IsTOC](#)
[IsTOCMarkAble](#)
[IsValid](#)
[Join AS Join class](#)
[Justifiable](#)
[Layer AS Layout class](#)
[LayerName](#)
[LeaderDotType](#)
[LeftExternalMargin](#)
[LeftPage AS Layout class](#)
[LeftTopCellId](#)
[LineLocation](#)
[LinkFrame](#)
[MaintainAspectRatio](#)
[MarginBottom](#)
[MarginLeft](#)
[MarginRight](#)
[MarginTop](#)
[MasterName](#)

[MinBottomMargin](#)
[MinHeight](#)
[MinLeftMargin](#)
[MinRightMargin](#)
[MinTopMargin](#)
[Name](#)
[NameBasedOnStyle](#)
[NumberOfLines](#)
[NumCols](#)
[NumColsSpannedOneCell](#)
[NumericFormat](#) AS [NumericFormat class](#)
[NumRowsSpannedOneCell](#)
[PageToUseLayoutOn](#)
[PaperName](#)
[Parent](#) AS [BaseObject class](#)
[RelativeType](#)
[RelativeXDistance](#)
[RelativeYDistance](#)
[RevisionType](#)
[RightExternalMargin](#)
[RightPage](#) AS [Layout class](#)
[ScaleHeight](#)
[ScaleMode](#)
[ScalePercentage](#)
[ScaleWidth](#)
[SelectType](#)
[Shadow](#) AS [Shadow class](#)
[Span](#)
[Style](#) AS [Layout class](#)
[StyleExceptions](#)
[TabRack](#) AS [TabRack class](#)
[TextOrient](#)
[Tile](#)
[TopExternalMargin](#)
[TopLeftCellRowId](#)
[UseFooter](#)
[UseHeader](#)
[UsePrinterSettings](#)
[UseWhen](#) AS [UseWhen class](#)
[VersionID](#)
[VertAlign](#)
[WasDeletedInRevMarkMode](#)
[WasInsertedInRevMarkMode](#)
[Where](#)
[Width](#)
[WPDataSets](#) AS [WPDataSetCollection class](#)
[WrapType](#)

XOffset
XPosition
YOffset
YPosition

Methods

AddChildToLayout
Backward
createlayer
DeleteContents
DeleteLayout
DoesMarkerNameMatch
FindClass
Forward
GetMarkerName
GetNamedProperty
GoToLayout
HasNamedProperty
ImportWatermarkGraphic
Mark
MirrorPage
MoveToBack
MoveToFront
registerwpdataset
RemoveChildFromLayout
RemoveNamedProperty
RevisionAcceptLayoutChange
RevisionCancelLayoutChange
setallmargins
SetMinimumOrigin
SetNamedProperty
SetPaperNameAndUpdateSize
SetupForCropping
UnregisterWPDataSet
Update
ValidateValue

Events

EnterLayout
KeyStroke
MouseDown
MouseUp

Word Pro: ContentCollection class members

Properties

Application AS WPAApplication class

Count

Description

IsValid

Name

Parent AS BaseObject class

VersionID

Methods

IsEmpty

Item

Events

None

Word Pro: Content class members

Properties

Application AS WPAApplication class

CanEmbed

ContentType

Description

IsEmpty

IsReplaceable

IsValid

Name

Parent AS BaseObject class

VersionID

Methods

None

Events

None

Word Pro: ContextMenuOptions class members

Properties

Application AS WPAApplication class

Description

IsCellMenuEnabled

IsFrameMenuEnabled

IsGraphicMenuEnabled

IsParallelColumnsMenuEnabled

IsTextMenuEnabled

IsValid

Name

Parent AS BaseObject class

VersionID

Methods

None

Events

None

Word Pro: DdeLinkCollection class members

Properties

Application AS WPAApplication class

Count

Description

IsValid

Name

Parent AS BaseObject class

VersionID

Methods

IsEmpty

Item

Events

None

Word Pro: DdeLinkManager class members

Properties

Application AS WPAApplication class

DdeLinks AS DdeLinkCollection class

Description

IsValid

Name

Parent AS BaseObject class

VersionID

Methods

AddDdeLink

GetMarkerName

InsertLink

MakeUniqueLinkName

RemoveDdeLink

RequestAndProcessData

Events

None

Word Pro: DdeLink class members

Properties

Application AS WPAApplication class

Description

GetFormatName

GetItemName

GetServerName

GetStatus

GetTopicName

IsUpdateAutomatic

IsValid

Name

Parent AS BaseObject class

UserLinkName

VersionID

Methods

EditLinkInfo

Events

None

Word Pro: DivisionCollection class members

Properties

Application AS WPAApplication class

Count

Description

IsValid

Name

Parent AS BaseObject class

VersionID

Methods

IsEmpty

Item

Events

None

Word Pro: DivisionInfo class members

Properties

AnyOleDdeLinks

Application AS WPAApplication class

Changed

ClassName

Color AS Color class

ContentName

Description

ExternalName

ExternalType

FillerPageText AS Text class

HasContents

IgnoreTab

IsDocObject

IsExpandRight

IsExternalFile

IsGotoable

IsScrollable

IsValid

LayoutName

LayoutOverride AS LayoutOverride class

Name

Parent AS BaseObject class

ShowTabs

SuppressHeaders

VersionID

Methods

None

Events

None

Word Pro: DivisionOptions class members

Properties

Application AS WPAApplication class

Description

HyphenationOptions AS HyphenationOptions class

IsTextLocked

IsValid

Language AS Language class

Name

Parent AS BaseObject class

ShowHiddenText

VersionID

Methods

None

Events

None

Word Pro: Division class members

Properties

[Application](#) AS [WPAApplication class](#)
[BookmarkManager](#) AS [BookmarkManager class](#)
[CanCreatePreviewBitmap](#)
[ContextLocked](#)
[CreatePreviewBitmap](#)
[DdeLinkManager](#) AS [DdeLinkManager class](#)
[DdeOutboundInfo](#)
[DemandLoad](#)
[Description](#)
[DivisionInfo](#) AS [DivisionInfo class](#)
[DivisionNames](#) AS [StringCollection class](#)
[DivisionOptions](#) AS [DivisionOptions class](#)
[Divisions](#) AS [DivisionCollection class](#)
[DocControl](#) AS [DocControl class](#)
[DocInfo](#) AS [DocInfo class](#)
[DocOptions](#) AS [Options class](#)
[EditorManager](#) AS [EditorManager class](#)
[EditorName](#)
[Epoch](#)
[FinishedSpellChecking](#)
[FirstChild](#)
[FirstCursorableDivision](#)
[FirstDivision](#)
[FirstName](#)
[FirstPage](#)
[FootnoteOptions](#) AS [FootnoteOptions class](#)
[Foundry](#) AS [Foundry class](#)
[HasIndex](#)
[HasToc](#)
[IsAnnotateOnly](#)
[IsChanged](#)
[IsChangedOtherThanLinkTo](#)
[IsChangedSinceTimeSave](#)
[IsChangedToLinkTo](#)
[IsDivisionExternal](#)
[IsDocLoading](#)
[IsEmptyDoc](#)
[IsEndnoteDivision](#)
[IsLockedForRevisions](#)
[IsProtected](#)
[IsSpellBarUp](#)
[IsStyleSheet](#)
[IsUndoOn](#)
[IsValid](#)

[LastChild](#)
[LastCursorableDivision](#)
[LastDivision](#)
[LastName](#)
[LastPage](#)
[LineNumberOptions](#) AS [LineNumberOptions class](#)
[Locked](#)
[LockLevel](#)
[Master](#) AS [TextDocument class](#)
[Name](#)
[NextCursorableDivision](#)
[NextName](#)
[NextNeighbor](#)
[NonUserDocument](#)
[NumDisplayablePagesInDoc](#)
[NumPagesInDoc](#)
[NumWindowsViewingDoc](#)
[Parent](#) AS [BaseObject class](#)
[ParentName](#)
[PathName](#)
[PreviousCursorableDivision](#)
[PreviousName](#)
[PreviousNeighbor](#)
[ReadCompressed](#)
[RevisionMarkMode](#)
[SelectionType](#)
[SortOptions](#) AS [SortOptions class](#)
[StateID](#)
[TabSpacing](#)
[UseContents](#)
[UsedFirstPageHeight](#)
[VersionID](#)
[VersionManager](#) AS [VersionManager class](#)
[WPDataSets](#) AS [WPDataSetCollection class](#)

Methods

[AddDivision](#)
[AnyEdits](#)
[BeginChange](#)
[Clear](#)
[ConsistencyCheck](#)
[Copy](#)
[EndChange](#)
[ForceDocToLoad](#)
[FXGetNotesString](#)
[FXGetNotesWriteHandle](#)
[FXSetNotesString](#)

[GetActiveList](#)
[GetNamedProperty](#)
[GetNameFromPage](#)
[HasNamedProperty](#)
[Hit](#)
[Link](#)
[Localize](#)
[Move](#)
[Purge](#)
[RegisterWPDataSet](#)
[RemoveDivision](#)
[RemoveNamedProperty](#)
[SaveDivision](#)
[SavePreviewFile](#)
[SetDocumentEpoch](#)
[SetNamedProperty](#)
[Unlink](#)
[UnregisterWPDataSet](#)
[UpdatePowerFields](#)
[UpdatePowerFieldsOnNew](#)

Events

[EnterClickHere](#)
[EnterLayout](#)
[ExitClickHere](#)
[KeyStroke](#)
[MouseDown](#)
[MouseUp](#)

Word Pro: DocControl class members

Properties

[AllowAlternateVerification](#)
[Application](#) AS [WPAApplication class](#)
[AutoVersion](#)
[DelayedGreeting](#)
[Description](#)
[DisableClickHereAction](#)
[DisableClickHeres](#)
[DisableExportToNotes](#)
[DisableVersionReview](#)
[DocControlPassword](#)
[DocControlProtection](#)
[DocControlRestrictedToEditor](#)
[EditorVerificationType](#)
[FilePassword](#)
[FileProtectionType](#)
[Greeting](#)
[IsValid](#)
[Name](#)
[NotesFlow](#)
[OcxDesignMode](#)
[Parent](#) AS [BaseObject class](#)
[RequestRemarkOnClose](#)
[RequireStartupScripts](#)
[ShowDivisionTabs](#)
[UseGreeting](#)
[VersionID](#)

Methods

None

Events

None

Word Pro: DocInfoFieldCollection class members

Properties

Application AS WPAApplication class

Count

Description

IsValid

Name

Parent AS BaseObject class

VersionID

Methods

IsEmpty

Item

Events

None

Word Pro: DocInfoFieldManager class members

Properties

Application AS WPAApplication class

Description

Fields AS DocInfoFieldCollection class

IsValid

Name

NumFields

Parent AS BaseObject class

VersionID

Methods

AddField

DeleteField

Events

None

Word Pro: DocInfoField class members

Properties

Application AS WPAApplication class

Contents

Description

ExportToNotesFX

IsValid

Name

Parent AS BaseObject class

VersionID

Methods

None

Events

None

Word Pro: DocInfo class members

Properties

Application AS WPAApplication class
AuthorName
CreationDateString
CreationTimeString
CreationTimeValue
Description
DocSize
FieldManager AS DocInfoFieldManager class
IsValid
Keywords
ModifiedDateString
ModifiedTimeString
ModifiedTimeValue
Name
NumCharsInDoc
NumPagesInDoc
NumWordsInDoc
Parent AS BaseObject class
TotalEditingTime
VersionID

Methods

ExportAllAsNotesFX
ExportAsNotesFX
IsExportedAsNotesFX
UpdateSelectedFields

Events

None

Word Pro: Documents class members

Properties

Application AS WPAApplication class

Count

Description

IsValid

Name

Parent AS BaseObject class

VersionID

Methods

IsEmpty

Item

Events

None

Word Pro: Document class members

Properties

Application AS WPAApplication class

Changed

Description

Embedded

FullName

IsOpen

IsValid

Location

Name

Parent AS BaseObject class

Path

PrintSettings AS PrintSettings class

ReadOnly

Saved

VersionID

Methods

Activate

Close

CopySelection

CutSelection

Paste

Print

PrintOut

Save

SaveAs

Events

PreClose

Save

SaveAs

Saved

SavedAs

Word Pro: DocWindowCollection class members

Properties

Application AS WPAApplication class

Count

Description

IsValid

Name

Parent AS BaseObject class

VersionID

Methods

IsEmpty

Item

Events

None

Word Pro: DocWindow class members

Properties

Active

Application AS WPAApplication class

BlockPaint

Caption

ClientWndHeight

ClientWndWidth

Description

Document AS TextDocument class

FitType

GapBetweenPanels

HasFocus

Height

HorizScrollBarVisible

Hwnd

IsValid

Left

MaxHorzPaneDistance

MaxVertPaneDistance

Name

NumCols

NumRowsThatFit

PageNumFirstPageShowing

PageNumLastPageShowing

Parent AS BaseObject class

TileWindow

Top

UsesPalette

VersionID

VertScrollBarVisible

ViewLevel

ViewType

Visible

Width

WindowId

WinViewPrefs AS WinViewPrefs class

XOffset

YOffset

Methods

Close

DarkMode

DestroyDocWindow

EndChange

GetPageNumFromDisplayablePageNum

Hide

Invalidate
Maximize
Minimize
Move
Open
RenderClipBitmap
RenderClipDIB
RenderClipMetafile
RenderClipPalette
Repaint
Resize
Restore
SetFocus
Show
ShowScrollBar
Update

Events

Moved
Moved

Word Pro: DropCapContainer class members

Properties

[AbsoluteTextOrientation](#)
[Application](#) AS [WPAApplication class](#)
[ClientHeight](#)
[ClientWidth](#)
[ContentHeight](#)
[ContentName](#)
[ContentWidth](#)
[Description](#)
[DisplayablePageNum](#)
[DivisionInfo](#) AS [DivisionInfo class](#)
[DivisionName](#)
[Height](#)
[IsInOleDivision](#)
[IsValid](#)
[Layout](#) AS [Layout class](#)
[MaxContentHeight](#)
[MaxContentWidth](#)
[Name](#)
[NumContainers](#)
[PageNum](#)
[Parent](#) AS [BaseObject class](#)
[PositionXOnPage](#)
[PositionYOnPage](#)
[Presentation](#) AS [Presentation class](#)
[RelativePageNum](#)
[SectionName](#)
[TextOrientation](#)
[VersionID](#)
[Width](#)

Methods

[Abandon](#)
[AddContainer](#)
[Adopt](#)
[Anchor](#)
[Backward](#)
[ConnectContainer](#)
[DeleteContainer](#)
[Disconnect](#)
[DoesMultiCellPaste](#)
[EditWrap](#)
[Ending](#)
[Forward](#)
[GetObjectList](#)
[GetPasteFormatCategories](#)

GoToContainer

Hide

IsPointWithin

linkcontainers

RevertToStyle

SetStyle

ShowContainers

Start

unlinkcontainers

Events

None

Example: AcquireTWAINImage method

'This example launches your scanner software and attempts to scan
'an image. If you do not have a scanner installed, an error message displays.

.AcquireTWAINImage

```
'Example: AddNewSectionTabs method
'This example displays divider tabs in the current document.
'The new AddNewSectionTabs method is called to create a new division.
'Finally, the new division is named 'my new division'.

.ActiveDocWindow.WinViewPrefs.IsViewSectionTabs = True
NewDiv = .ApplicationWindow.SectionTabs.AddNewSectionTabs()
.ActiveDocument.Divisions(NewDiv).DivisionInfo.Name = "My New Division"
```

Example: CreateDropCap method

'This script types some text into a new paragraph, then creates a Drop Cap four lines high beside the newly created paragraph.

For I = 1 to 10

 .Type "Now is the time for all good men to come to the aid of their country. "

Next

.CreateDropCap 4, 3

.Page.Layout.GoToLayout ' return the insertion point to the page

'Example: CreateEmptyList method

'This example creates a SmartFill list named "Day Parts", and adds

'four items to the list.

'Note that 1033 is the numeric value for the American language code.

```
.Application.SmartFill(1033).CreateEmptyList "Day Parts", True
```

```
.Application.SmartFill(1033).AddStringToList "Day Parts", "Morning"
```

```
.Application.SmartFill(1033).AddStringToList "Day Parts", "Afternoon"
```

```
.Application.SmartFill(1033).AddStringToList "Day Parts", "Evening"
```

```
.Application.SmartFill(1033).AddStringToList "Day Parts", "Night"
```

Example: CreateLayer method

'This example creates a new layer if one does not already exist, and then
'imports a watermark into the layer.

```
If .Layout.LayerName = "" Then ' if no name, we need to create the layer
```

```
    .Layout.CreateLayer
```

```
Else ' we have one already; get rid of current contents
```

```
    .Layout.Layer.DeleteContents
```

```
End If
```

```
wmpath = .ApplicationWindow.UserInterfacePrefs.WatermarksPath
```

```
' place the watermark graphic
```

```
.Layout.Layer.ImportWatermarkGraphic wmpath & "\_draft.wmf", "", False, False
```

Example: DeleteBag method

```
' This example creates a bag in the active division and then writes some data  
' to the bag. The data from the created bag is read and printed to the LotusScript  
' Output panel. Next, data from all bags in the Bag Collection is  
' printed and then each bag is deleted.  
' RUNTIME DEPENDENCIES: You must have a document open for this script to work.
```

```
Dim BagName As String  
Dim MyBag As Bag  
Dim BagData As String
```

```
BagData = "This is data for the bag."  
LenBagData = Len(BagData)
```

```
' create the bag, and set up a variable representing it.  
BagName = .Division.Foundry.Create($LwpFoundryCreateTypeBag)  
Set MyBag = .Division.Foundry.Bags.Item(BagName)
```

```
' Write the data to the bag
```

```
Stat = MyBag.Write(BagData, LenBagData)  
If Stat = True Then  
    Print "BagData= " & MyBag.Read(LenBagData)  
End If
```

```
' list the contents of the bags.
```

```
Forall ThisBag In .Division.Foundry.Bags  
    ThisBag.Reset  
    Print "Name = " ThisBag.Name  
    Print "Length = " ThisBag.Length  
    Print "Contents = " & ThisBag.Read(ThisBag.Length)  
    ThisBag.DeleteBag  
End Forall
```

Example: EnvelopeBarCode method

'This example creates an envelope in a new document. The send to address
'is filled in, and a bar code is inserted.

```
.NewDocument  
.StartEnvelopeDiv  
.Division.Foundry.Layouts("Envelope Send").GoToLayout  
.Text.MoveToStart $LwpLocationTypeStream  
.Text.InsertText "Mr. John Doe"  
.Text.SplitParagraph  
.Text.InsertText "123 Main Street"  
.Text.SplitParagraph  
.Text.InsertText "Anywhere, GA 30244"  
.EnvelopeBarCode
```

'Example: GetParagraphNumberString method
'This example inserts some numbered headings using the default outline
'style sequence. The GetParagraphNumberString method is used to get the
'numbering of the second heading paragraph, this is then inserted into the document.

```
.Text.Bullet.Name = "Default Outline"  
.Text.Numbering.Position = 1  
.Text.Indent.First = 360  
.Text.Indent.Rest = 360  
.Text.InsertText "This is heading paragraph 1.", True  
.Text.Bullet.Name = "Default Outline"  
.Text.Numbering.Position = 1  
.Text.Indent.First = 360  
.Text.Indent.Rest = 360  
.Text.InsertText "This is heading paragraph 2."  
MyMark = .Text.Mark($LwpMarkerTypeDefault)  
.Text.InsertText "", True  
.Text.InsertText "This some body text.", True  
ParaRef = .Foundry.Markers(MyMark).GetParagraphNumberString  
.Text.InsertText "This is a reference to the second heading paragraph, numbered " &  
ParaRef & "."
```

```
'Example: GetParagraphNumber method
'This example inserts some numbered text into the current document.
'The second paragraph is marked with a default marker. Finally, the
'GetParagraphNumber method is used to reference the paragraph number.

.Text.Bullet.Name = "Default Outline"
.Text.Numbering.Position = 1
.Text.Indent.First = 360
.Text.Indent.Rest = 360
.Text.InsertText "This is heading paragraph 1.", True
.Text.Bullet.Name = "Default Outline"
.Text.Numbering.Position = 1
.Text.Indent.First = 360
.Text.Indent.Rest = 360
.Text.InsertText "This is heading paragraph 2."
MyMark = .Text.Mark($LwpMarkerTypeDefault)
.Text.InsertText "", True
.Text.InsertText "This some body text.", True
ParaRef = .Foundry.Markers(MyMark).GetParagraphNumber(1)
.Text.InsertText "This is a reference to the second heading paragraph, numbered " &
ParaRef & "."
```

'Example: GoToClickHere method

'This example inserts a click here block in the document, then moves the

'insertion point to the block.

```
.Text.InsertText "This is some text which precedes the Click Here Block. ", False
MyClickHere = .InsertClickHere()
.Division.Foundry.ClickHeres(MyClickHere).Prompt.Clear
.Division.Foundry.ClickHeres(MyClickHere).Prompt.InsertText "Click here to type Text",
False, $LwpTextTypeNative
.Division.Foundry.ClickHeres(MyClickHere).Action = 1
.Division.Foundry.ClickHeres(MyClickHere).HelpText = "Click here to type Text"
.Division.Foundry.ClickHeres(MyClickHere).UsesHelp = False
.Division.Foundry.ClickHeres(MyClickHere).TabOrder = 1
.Division.Foundry.ClickHeres(MyClickHere).TabExits = True
.Division.Foundry.ClickHeres(MyClickHere).Name = MyClickHere
.Text.InsertText "This is some text which follows the Click Here Block. ", True
.Text.MoveToStart $LwpLocationTypeDocument
MessageBox "Click OK to go to the Click Here Block", MB_OK, "Example Script"
.GoToClickHere (MyClickHere)
```

Example: ImportWatermarkGraphic method

'This example creates a new layer if one does not already exist, and then
'imports a watermark into the layer.

```
If .Layout.LayerName = "" Then ' if no name, we need to create the layer
```

```
    .Layout.CreateLayer
```

```
Else ' we have one already; get rid of current contents
```

```
    .Layout.Layer.DeleteContents
```

```
End If
```

```
wmpath = .ApplicationWindow.UserInterfacePrefs.WatermarksPath
```

```
' place the watermark graphic
```

```
.Layout.Layer.ImportWatermarkGraphic wmpath & "\_draft.wmf", "", False, False
```


Example: NextCycleStyle method

'This example types some text, and then uses the NextCycleStyle method to
'change the paragraph style used.

```
.Text.InsertText "Now is the time for all good men to come to the aid of their party."  
", False
```

```
.NextCycleStyle
```

Example: RestorePreviousView method

'This example displays the current document in custom view, then
'restores the previous view.

```
.ActiveDocWindow.WinViewPrefs.ViewType = &H8 ' custom view level  
MessageBox "Click OK to restore the previous view.", MB_OK, "Example Script"  
.ActiveDocWindow.WinViewPrefs.RestorePreviousView
```

Example: SelectTWAINSource method

'This example launches your scanner software to allow you to select
'an image source. If you do not have scanner software installed,
'an error message displays.

.SelectTWAINSource

Example: SetAllMargins method

'This example creates a frame in the current document, then sets the
'left and right margins to 1/8 inch, and the top and bottom margins to 1/4 inch.

```
.NewFrame 3927, 2807, 4951, 1654, "Default Frame"  
.Frame.Layout.SetAllMargins &HF, 180, 180, 360, 360
```

Example: SetCustomNumber method

'This example inserts two paragraphs of custom numbered text into the
'current document.

```
.SetCustomNumber 0, "", $LwpNumberIncludenone, "Item ", $LwpNumberingStyleBasic, 0, 0,  
""
```

```
.Text.Indent.First = 720
```

```
.Text.Indent.Rest = 720
```

```
.Text.InsertText "This is a sample paragraph. ", True
```

```
.SetCustomNumber $LwpNumberResetonspecificstyle, "Default Text",  
$LwpNumberIncludedivision, "", $LwpNumberingStyleUppercaseroman, 0, 0, "."
```

```
.Text.InsertText "This is a sample paragraph. ", True
```

Example: SetNumberingLevelInfo method

'This example sets sets the outline numbering style for the first three levels
'of the Default Outline numbering sequence.

```
.SetNumberingLevelInfo "Default Outline", 1, $LwpNumberResetonlevel, "", -3,  
$LwpNumberIncludenone, "", -3, $LwpNumberingStyleUppercaseroman, 0, 0, "."  
.SetNumberingLevelInfo "Default Outline", 2, $LwpNumberResetonlevel, "", -5,  
$LwpNumberIncludenone, "", -5, $LwpNumberingStyleUppercaseletters, 0, 0, "."  
.SetNumberingLevelInfo "Default Outline", 3, $LwpNumberResetonlevel, "", -9,  
$LwpNumberIncludenone, "", -9, $LwpNumberingStyleBasic, 0, 0, "."
```

'Example: SpecialView method

'This example displays the current document in Panorama view.

.SpecialView 1

Example: SplitDivision method

'This example displays divider tabs, then creates splits the current
'division at the insertion point.

```
.ActiveDocWindow.WinViewPrefs.IsViewSectionTabs = True
```

```
.SplitDivision
```


'Example: SplitWindow method

'This example splits the current document window horizontally, then

'clears the split, and splits the window vertically.

```
MessageBox "Click OK to split the window horizontally", MB_OK, "Example Script"
```

```
.SplitWindow True
```

```
MessageBox "Click OK to split the window vertically", MB_OK, "Example Script"
```

```
.ActiveDocWindow.WinViewPrefs.ClearSplits ' clear current split
```

```
.SplitWindow False
```

'Example: WordCount method

'This example inserts some text into the current document, then

'runs a word count.

.Text.InsertText "This is some sample text which will be used to test the word count method."

.WordCount

Word Pro: Action property

This property is defined in the following classes:

[ClickHere](#)

[MenuItem](#)

Word Pro: Contents property

This property is defined in the following classes:

DocInfoField

Foundry

Word Pro: Foundry property

{button ,AL(^H_DIVISION_CLASS;H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

(Read-only) The Foundry object which you use for accessing objects in a Division. In Word Pro, there is always one Foundry object for each Division object. The Foundry property on the WPApplication object (always named WordPro) always contains the Foundry object for the currently active division.

Data Type

[Foundry](#)

Syntax

foundryvalue = [objectreference].Foundry

Legal values

Usage

Division.Foundry

The Division Foundry provides access to all the objects in that division, including Layouts, Text, Graphics, Markers, Tables, Footnotes, and so on. You can access all the objects in a division through the appropriate collection in the Division Foundry.

WordPro.Foundry

The Foundry property on WordPro provides a shortcut to the currently active division's Foundry object. The Foundry object in WordPro.Foundry changes as the focus changes from one Division object to another. For example, if you had a document with one division named Overview and another division named Summary, the contents of the Foundry property on WordPro change as you move the focus from Overview to Summary. When the focus is on the Overview division, this property contains the Foundry object for the Division object named Overview. When the focus changes to the Summary division, the contents of this property change to the Foundry object for the Division object named Summary.

Word Pro: Shadow property

This property is defined in the following classes:

FontMetrics

Layout

ParagraphBorder

Word Pro: URLDownloadComplete property

{button ,AL(^H_OLEOBJECT_CLASS',0)} [See list of classes](#)

(Read-only)

Indicates whether or not the download of an OLE object is complete. This only applies to OLE objects which are part of a document opened from the internet.

Data Type

Integer

Syntax

urldownloadcompletevalue = [objectreference].URLDownloadComplete

Legal values

This property is read-only. The value of this property cannot be set by a script.

Usage

You can set up a Forall loop which continually iterates over all the OLE objects in a division until all of them return True. This would allow you to continue your script only after the entire document has been downloaded from the internet.

Word Pro: UseConsistentSpaceBetweenSentences property

{button ,AL('H_FORMATCHECKPREF_CLASS',0)} [See list of classes](#)

{button ,AL('H_USECONSISTENTSPACEBETWEENSENTENCES_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Determines whether Word Pro will check spacing between sentences when performing a format check.

Data Type

[Integer](#)

Syntax

useconsistentspacebetweensentencesvalue = [objectreference].UseConsistentSpaceBetweenSentences

[objectreference].UseConsistentSpaceBetweenSentences = useconsistentspacebetweensentencesvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

This property is equivalent to the "Check spacing between sentences" option in the Format Check Options dialog box.

Word Pro: UseContents property

{button ,AL('H_DIVISION_CLASS;H_TEXTDOCUMENT_CLASS',0)} [See list of classes](#)

(Read-write)

Data Type

Integer

Syntax

usecontentsvalue = [objectreference].UseContents

[objectreference].UseContents = usecontentsvalue

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Word Pro: UseCycleKeys property

{button ,AL('H_USERINTERFACEPREFS_CLASS','0')} [See list of classes](#)

{button ,AL('H_USECYCLEKEYS_PROPERTY_EXSCRIPT',1')} [See example](#)

(Read-write)

This language element has not yet been defined. You can check for a newer Help file on the Lotus web site.

Data Type

Integer

Syntax

usecyclekeysvalue = [objectreference].UseCycleKeys

[objectreference].UseCycleKeys = usecyclekeysvalue

Legal values

The legal values for this property are -1 and 0 but you may use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Word Pro: UseDefaultPrinter property

{button ,AL('H_PRINTMANAGER_CLASS',0)} [See list of classes](#)

(Read-write)

Data Type

Integer

Syntax

usedefaultprintvalue = [objectreference].UseDefaultPrinter

[objectreference].UseDefaultPrinter = usedefaultprintvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Word Pro: UsedFirstPageHeight property

{button ,AL('H_DIVISION_CLASS',0)} [See list of classes](#)

(Read-only)

The amount of space, expressed in twips, used by the first page of a division.

Note Not valid on TextDocument objects or divisions with more than one page.

Data Type

Long

Syntax

usedfirstpageheightvalue = [objectreference].UsedFirstPageHeight

Legal values

Data type is Long but the unit of measurement used for this property is Twips. There are 1440 Twips per inch.

Usage

Use this property when you need to know how much space a 1-page division uses.

Assuming a division has only one page, this property records how much of that page is actually used (i.e. how far down the page the content extends). For example, you can use this property to determine the height of an OLE embedded document, assuming that the document is not more than one page.

Word Pro: UseFindStyle property

{button ,AL('H_FINDANDREPLACE_CLASS',0)} [See list of classes](#)

(Read-write) Enables the user to find a specific paragraph style when using Find & Replace.

Data Type

Integer

Syntax

usefindstylevalue = [objectreference].UseFindStyle

[objectreference].UseFindStyle = usefindstylevalue

Legal values

The legal values for this property are 0 and 1.

Usage

Use this property in conjunction with the FindStyleName property to find a specific paragraph style.

Setting this property to 1 is equivalent to choosing Edit - Find & Replace Text, clicking Options, clicking the Font button in the "Find options" section, and selecting the "Style" box.

Setting this property to 0 is equivalent to deselecting the "Style" box.

Word Pro: UseFooter property

{button ,AL(^H_LAYOUT_CLASS;H_CELLGROUPLAYOUT_CLASS;H_CELLLAYOUT_CLASS;H_CONNECTEDLAYOUT_CLASS;H_COLUMNGROUPLAYOUT_CLASS;H_FOOTERLAYOUT_CLASS;H_FRAMELAYOUT_CLASS;H_FRAMEGROUPLAYOUT_CLASS;H_DROP_CAPLAYOUT_CLASS;H_GROUPLAYOUT_CLASS;H_HEADERLAYOUT_CLASS;H_NOTELAYOUT_CLASS;H_PAGELAYOUT_CLASS;H_ROWGROUPLAYOUT_CLASS;H_ROWLAYOOUT_CLASS;H_RUBYLAYOUT_CLASS;H_SUPERTABLEGROUPLAYOUT_CLASS;H_SUPERTABLELAYOUT_CLASS;H_TABLELAYOUT_CLASS;H_ENDNOTELAYOUT_CLASS;H_FOOTNOTELAYOUT_CLASS;H_TABLEHEADINGLAYOUT_CLASS;H_TOCSUPERTABLELAYOUT_CLASS',0)} [See list of classes](#)

(Read-write) Allows you to add or remove a footer in a layout object.

Data Type

Integer

Syntax

usefootervalue = [objectreference].UseFooter

[objectreference].UseFooter = usefootervalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

When you assign a value of True to this property, the footer object inserted in the layout object has a height equal to the bottom margin of the layout object, and a width equal to the total width of the layout object.

Word Pro: UseGreeting property

{button ,AL('H_DOCCONTROL_CLASS',0)} [See list of classes](#)

{button ,AL('H_USEGREETING_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Indicates whether or not a greeting displays when a file is opened.

Data Type

[Integer](#)

Syntax

usegreetingvalue = [objectreference].UseGreeting

[objectreference].UseGreeting = usegreetingvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Equivalent to choosing File - TeamSecurity and selecting "Display Greeting with this text" box on the Editing Rights panel.

Word Pro: UseHeader property

```
{button ,AL(^H_LAYOUT_CLASS;H_CELLGROUPLAYOUT_CLASS;H_CELLLAYOUT_CLASS;H_CONNECTEDLAYOUT_CLASS;H_COLUMNGROUPLAYOUT_CLASS;H_FOOTERLAYOUT_CLASS;H_FRAMELAYOUT_CLASS;H_FRAMEGROUPLAYOUT_CLASS;H_DROP_CAPLAYOUT_CLASS;H_GROUPLAYOUT_CLASS;H_HEADERLAYOUT_CLASS;H_NOTELAYOUT_CLASS;H_PAGELAYOUT_CLASS;H_ROWGROUPLAYOUT_CLASS;H_ROWLAYOUT_CLASS;H_RUBYLAYOUT_CLASS;H_SUPERTABLEGROUPLAYOUT_CLASS;H_SUPERTABLELAYOUT_CLASS;H_TABLELAYOUT_CLASS;H_ENDNOTELAYOUT_CLASS;H_FOOTNOTELAYOUT_CLASS;H_TABLEHEADINGLAYOUT_CLASS;H_TOCSUPERTABLELAYOUT_CLASS',0)} See list of classes
```

(Read-write) Allows you to add or remove a header in a layout object.

Data Type

Integer

Syntax

useheadervalue = [objectreference].UseHeader

[objectreference].UseHeader = useheadervalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

When you assign a value of True to this property, the header object inserted in the layout object has a height equal to the top margin of the layout object, and a width equal to the total width of the layout object.

Word Pro: UseNextStyle property

{button ,AL('H_BREAKS_CLASS',0)} [See list of classes](#)

{button ,AL('H_USENEXTSTYLE_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Enables or disables the option to use the specified style in the next paragraph.

Data Type

[Integer](#)

Syntax

usenextstylevalue = [objectreference].UseNextStyle

[objectreference].UseNextStyle = usenextstylevalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

This property is equivalent to the "Style to use for next paragraph" box in the Text InfoBox.

Use the NextStyleName property to assign the paragraph style to be used in the next paragraph.

{button ,AL('H_NEXTSTYLENAME_PROPERTY_MEMDEF',0)} [See related topics](#)

Word Pro: UsePrinterSettings property

```
{button ,AL(^H_LAYOUT_CLASS;H_CELLGROUPLAYOUT_CLASS;H_CELLLAYOUT_CLASS;H_CONNECTEDLAY  
OUT_CLASS;H_COLUMNGROUPLAYOUT_CLASS;H_FOOTERLAYOUT_CLASS;H_FRAMELAYOUT_CLASS;H  
_FRAMEGROUPLAYOUT_CLASS;H_DROP_CAPLAYOUT_CLASS;H_GROUPLAYOUT_CLASS;H_HEADERLAY  
OUT_CLASS;H_NOTELAYOUT_CLASS;H_PAGELAYOUT_CLASS;H_ROWGROUPLAYOUT_CLASS;H_ROWLA  
YOUT_CLASS;H_RUBYLAYOUT_CLASS;H_SUPERTABLEGROUPLAYOUT_CLASS;H_SUPERTABLELAYOUT  
_CLASS;H_TABLELAYOUT_CLASS;H_ENDNOTELAYOUT_CLASS;H_FOOTNOTELAYOUT_CLASS;H_TABLE  
HEADINGLAYOUT_CLASS;H_TOCSUPERTABLELAYOUT_CLASS',0)} See list of classes
```

```
{button ,AL(^H_USEPRINTERSETTINGS_PROPERTY_EXSCRIPT',1)} See example
```

(Read-write) Instructs the layout object to retrieve printer setting information, such as orientation, printer bin, scale, and so on.

Data Type

Integer

Syntax

useprintersettingsvalue = [objectreference].UsePrinterSettings

[objectreference].UsePrinterSettings = useprintersettingsvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Word Pro: UserClassNameApp property

{button ,AL(^H_OLEOBJECT_CLASS',0)} [See list of classes](#)

(Read-only)

Note This property is not supported for the class OLEObject within OS/2.

Data Type

String

Syntax

userclassnameappvalue = [objectreference].UserClassNameApp

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Word Pro: UserClassNameFull property

{button ,AL(^H_OLEOBJECT_CLASS',0)} [See list of classes](#)

(Read-only)

Note This property is not supported for the class OLEObject within OS/2.

Data Type

String

Syntax

userclassnamefullvalue = [objectreference].UserClassNameFull

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Word Pro: UserClassNameShort property

{button ,AL(^H_OLEOBJECT_CLASS',0)} [See list of classes](#)

(Read-only)

Note This property is not supported for the class OLEObject within OS/2.

Data Type

String

Syntax

userclassnameshortvalue = [objectreference].UserClassNameShort

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Word Pro: UserDefinedFilter property

{button ,AL(`H_USERINTERFACEPREFS_CLASS',0)} [See list of classes](#)

{button ,AL(`H_USERDEFINEDFILTER_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Sets or returns the list of file extensions that are displayed when opening files using the Custom filter.

Data Type

String

Syntax

userdefinedfiltervalue = [objectreference].UserDefinedFilter

[objectreference].UserDefinedFilter = userdefinedfiltervalue

Legal values

Must contain a string that follows valid file naming conventions.

Usage

Equivalent to the "Default File Open Types (*.lwp;*.sam;...)" field on the Default Files panel of the Word Pro Preferences dialog box.

The value assigned to this property must adhere to valid file naming conventions. For example, to display documents with the .lwp or .txt extension, you can use the following statement:

```
.ApplicationWindow.UserInterfacePrefs.UserDefinedFilter = "*.lwp;*.txt"
```

{button ,AL(`H_WORKINGTYPE_PROPERTY_MEMDEF',0)} [See related topics](#)

Word Pro: UserDictFiles property

{button ,AL('H_USERINTERFACEPREFS_CLASS',0)} [See list of classes](#)

(Read-only) The default user dictionary file.

Data Type

String

Syntax

userdictfilesvalue = [objectreference].UserDictFiles

Legal values

A valid user dictionary file with the file extension .UDC.

Usage

Equivalent to the "Default user dictionary" field on the Default files panel of the Word Pro Preferences dialog box. In Word Pro, the "Default user dictionary" field can contain multiple paths. This property contains the first path listed in the "Default user dictionary" field.

Word Pro: UserDictionaryFiles property

{button ,AL(^H_USERINTERFACEPREFS_CLASS',0)} [See list of classes](#)

(Read-only) Stores multiple user dictionary names.

Data Type

[StringCollection](#)

Syntax

userdictionaryfilesvalue = [objectreference].UserDictionaryFiles

Legal values

Always contains an instance of the StringCollection class.

Usage

Equivalent to the "Default user dictionary" field on the Default files panel of the Word Pro Preferences dialog box. In Word Pro, the "Default user dictionary" field can contain multiple file names. You can use this property to read these multiple file names, including the default file which is stored in the UserDictFiles property.

Word Pro: UserDictionaryPaths property

{button ,AL('H_USERINTERFACEPREFS_CLASS',0)} [See list of classes](#)

(Read-only) Contains multiple paths (drive and directory) for the Word Pro user dictionary.

Data Type

[StringCollection](#)

Syntax

userdictionarypathsvalue = [objectreference].UserDictionaryPaths

Legal values

Always contains an instance of the StringCollection class.

Usage

Equivalent to the "User dictionaries" field on the Locations panel of the Word Pro Preferences dialog box. This field can contain multiple paths. This property returns a collection of String objects which contain the names of all default Word Pro user dictionary paths.

Word Pro: UserDictionaryPath property

{button ,AL(^H_USERINTERFACEPREFS_CLASS,0)} [See list of classes](#)

(Read-write) The default path (drive and directory) for the Word Pro user dictionary.

Data Type

String

Syntax

userdictionarypathvalue = [objectreference].UserDictionaryPath

[objectreference].UserDictionaryPath = userdictionarypathvalue

Legal values

A valid path including drive, directory, and file name.

Usage

Equivalent to the "User dictionaries" option on the Locations panel of the Word Pro Preferences dialog box. In Word Pro, this field can contain multiple paths. This property contains the first path listed in the "User dictionaries" field.

Word Pro: UserDictStates property

{button ,AL('H_USERINTERFACEPREFS_CLASS',0)} [See list of classes](#)

(Read-only) Indicates the state (on or off) of each user dictionary.

Data Type

String

Syntax

userdictstatesvalue = [objectreference].UserDictStates

Legal values

Although the legal value for this property is a String, the string contains a series of Integers (0s and 1s) that indicate if a dictionary is turned on or off. A value of 0 means the dictionary is not being used. A value of 1 means the dictionary is being used.

The series of integers is separated by commas and corresponds to the string of user dictionary files found in the UserDictFiles property.

For example, UserDictFiles may be set to "d:\wordpro\userdic.udc; c:\lotus\components\wordpro\userdic.udc;s:\lotus\userdic.udc." Then the UserDictStates property might contain the string "0,1,0," indicating that the only user dictionary enabled is the second one, or c:\lotus\components\wordpro\userdic.udc.

Usage

Equivalent to the "User Dictionary(s) to use" option in the Spell Check Options dialog box. The user can select multiple user dictionaries in this box.

Word Pro: UseRelative property

{button ,AL('H_INDENT_CLASS;H_RELATIVEINIDENT_CLASS',0)} [See list of classes](#)

{button ,AL('H_USERELATIVE_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write)

Data Type

Integer

Syntax

userrelativevalue = [objectreference].UseRelative

[objectreference].UseRelative = userrelativevalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Word Pro: UseReplaceStyle property

{button ,AL('H_FINDANDREPLACE_CLASS',0)} [See list of classes](#)

(Read-write) Enables the user to replace a paragraph style in Find & Replace.

Data Type

Integer

Syntax

usereplacestylevalue = [objectreference].UseReplaceStyle

[objectreference].UseReplaceStyle = usereplacestylevalue

Legal values

0 = Don't replace

1 = Replace

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

The UI equivalent is the "Style" check box in the replace properties panel, not the style name itself.

Use this property to replace a paragraph style in Find & Replace. Equivalent to choosing Edit - Find & Replace Text, clicking Options, clicking the Font button in the "Replace options" section, and selecting the Style checkbox.

Word Pro: UserInitials property

{button ,AL('H_PREFERENCES_CLASS',0)} [See list of classes](#)

{button ,AL('H_USERINITIALS_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write)

Data Type

[String](#)

Syntax

userinitialsvalue = [objectreference].UserInitials

[objectreference].UserInitials = userinitialsvalue

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Word Pro: UserInterfacePrefs property

{button ,AL('H_APPLICATIONWINDOW_CLASS',0)} [See list of classes](#)

(Read-only) An instance of the UserInterfacePrefs class.

Data Type

[UserInterfacePrefs](#)

Syntax

userinterfaceprefsvalue = [objectreference].UserInterfacePrefs

Legal values

Always contains an instance of the UserInterfacePrefs class.

Usage

This property provides access to the UserInterfacePrefs class. With this class you can set User Preference settings, such as units of measurement, welcome screen options, document paths, SmartMaster paths, and so on.

Word Pro: UserLinkName property

{button ,AL(^H_DDELINK_CLASS',0)} [See list of classes](#)

(Read-only) The user name for a DDE Link.

Data Type

String

Syntax

userlinknamevalue = [objectreference].UserLinkName

Legal values**Usage**

This property gives you the user name for the DdeLink (composed of the server name, the topic name, and the item name).

Word Pro: UserName property

{button ,AL('H_PREFERENCES_CLASS;H_SECTION_CLASS;H_INDEXSECTION_CLASS',0)} [See list of classes](#)

{button ,AL('H_USERNAME_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write)

Data Type

[String](#)

Syntax

usernamevalue = [objectreference].UserName

[objectreference].UserName = usernamevalue

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

{button ,AL('H_SECTIONUSERNAME_PROPERTY_MEMDEF;H_SECTIONNAME_PROPERTY_MEMDEF',0)} [See related topics](#)

Word Pro: UseSeparatorLine property

{button ,AL(^H_FOOTNOTESEPOPT_CLASS;H_FOOTNOTECONTSEP_CLASS;H_FOOTNOTESEPARATOR_CLASSES',0)} [See list of classes](#)

{button ,AL(^H_USESEPARATORLINE_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Determines whether or not to use a separator line for a footnote.

Data Type

Integer

Syntax

useseparatorlinevalue = [objectreference].UseSeparatorLine

[objectreference].UseSeparatorLine = useseparatorlinevalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Equivalent to the "Separator line for" box on the Separators panel of the Footnote and Endnote Options dialog box.

Word Pro: UsesHelp property

{button ,AL('H_CLICKHERE_CLASS',0)} [See list of classes](#)

{button ,AL('H_USESHELP_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Indicates whether a ClickHere object has bubble help.

Data Type

[Integer](#)

Syntax

useshelpvalue = [objectreference].UsesHelp

[objectreference].UsesHelp = useshelpvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

{button ,AL('H_HELPTEXT_PROPERTY_MEMDEF',0)} [See related topics](#)

Word Pro: UsesPalette property

{button ,AL(^H_DOCWINDOW_CLASS',0)} [See list of classes](#)

(Read-only)

Data Type

[Integer](#)

Syntax

usespalettevalue = [objectreference].UsesPalette

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Word Pro: UseSuperscriptReferenceNum property

{button ,AL('H_FOOTNOTENUMOPT_CLASS;H_ENDNOTEDIVISIONGROUPNUM_CLASS;H_ENDNOTEDIVISIONNUM_CLASS;H_ENDNOTEDOCNUM_CLASS;H_FOOTNOTENUMBERING_CLASS',0)} [See list of classes](#)

mayday_button(See Example, H_USESUPERSCRIPTREFERENCENUM_PROPERTY_EXSCRIPT)

(Read-write) Changes footnote or endnote references in the text to superscript.

Data Type

Integer

Syntax

usesuperscriptreferencenumvalue = [objectreference].UseSuperscriptReferenceNum

[objectreference].UseSuperscriptReferenceNum = usesuperscriptreferencenumvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Equivalent to choosing Create - Footnote/Endnote, clicking Options, and selecting "Superscript reference number" on the Numbering panel.

Word Pro: UseTwoSpacesBetweenSentences property

{button ,AL('H_FORMATCHECKPREF_CLASS',0)} [See list of classes](#)

{button ,AL('H_USETWOSPACESBETWEENSENTENCES_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Determines whether Format Check will check for one or two spaces between sentences.

Data Type

[Integer](#)

Syntax

usetwospacesbetweensentencesvalue = [objectreference].UseTwoSpacesBetweenSentences

[objectreference].UseTwoSpacesBetweenSentences = usetwospacesbetweensentencesvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

This property is equivalent to selecting "1 space" or "2 spaces" in the Format Check Options dialog box.

Use the UseConsistentSpaceBetweenSentences property to determine whether Format Check checks the spacing between sentences.

Word Pro: UseWhen property

```
{button ,AL('H_LAYOUT_CLASS;H_CELLGROUPLAYOUT_CLASS;H_CELLLAYOUT_CLASS;H_CONNECTEDLAY  
OUT_CLASS;H_COLUMNGROUPLAYOUT_CLASS;H_FOOTERLAYOUT_CLASS;H_FRAMELAYOUT_CLASS;H  
_FRAMEGROUPLAYOUT_CLASS;H_DROP_CAPLAYOUT_CLASS;H_GROUPLAYOUT_CLASS;H_HEADERLAY  
OUT_CLASS;H_NOTELAYOUT_CLASS;H_PAGELAYOUT_CLASS;H_ROWGROUPLAYOUT_CLASS;H_ROWLA  
YOUT_CLASS;H_RUBYLAYOUT_CLASS;H_SUPERTABLEGROUPLAYOUT_CLASS;H_SUPERTABLELAYOUT  
_CLASS;H_TABLELAYOUT_CLASS;H_ENDNOTELAYOUT_CLASS;H_FOOTNOTELAYOUT_CLASS;H_TABLE  
HEADINGLAYOUT_CLASS;H_TOCSUPERTABLELAYOUT_CLASS',0)} See list of classes
```

(Read-only) Allows you to determine when a specific layout object should be used.

Data Type

[UseWhen](#)

Syntax

usewhenevervalue = [objectreference].UseWhen

Legal values

Always contains an instance of the UseWhen class.

Usage

Word Pro: Valid property

{button ,AL(^H_BULLET_CLASS',0)} [See list of classes](#)

(Read-only)

Data Type

Variant (Enumerated)

CommandState

Syntax

validvalue = [objectreference].Valid

Legal values

\$LwpCommandStateOff (151)

\$LwpCommandStateOn (152)

\$LwpCommandStateStyle (153)

Usage

Word Pro: VersionID property

{button ,AL(^H_BASEOBJECT_CLASS',0)} [See list of classes](#)

(Read-only) A numeric expression that represents the version number of the LotusScript engine in which an object was generated. This is essential when comparing the relative functionality of certain LotusScript language elements and Word Pro classes.

Data Type

Long

Syntax

versionidvalue = [objectreference].VersionID

Legal values

You cannot set the value of this property.

Usage

Use this property to determine in which version of the Word Pro object model an object was generated. By comparing the VersionID properties of two language elements, you can determine if the two language elements are likely to be compatible. This is especially useful when debugging or analyzing inherited or legacy scripts.

Word Pro: VersionManager property

{button ,AL(^H_DIVISION_CLASS;H_TEXTDOCUMENT_CLASS',0)} [See list of classes](#)

{button ,AL(^H_VERSIONMANAGER_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-only)

Data Type

[VersionManager](#)

Syntax

versionmanagervalue = [objectreference].VersionManager

Legal values

Always contains an instance of the VersionManager class.

Usage

Word Pro: VersionName property

{button ,AL('H_VERSION_CLASS',0)} [See list of classes](#)

{button ,AL('H_VERSIONNAME_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write)

Data Type

[String](#)

Syntax

versionnamevalue = [objectreference].VersionName

[objectreference].VersionName = versionnamevalue

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros, and then choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Word Pro: Versions property

{button ,AL('H_VERSIONMANAGER_CLASS',0)} [See list of classes](#)

{button ,AL('H_VERSIONS_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-only)

Data Type

[VersionCollection](#)

Syntax

versionsvalue = [objectreference].Versions

Legal values

Always contains an instance of the VersionCollection class.

Usage

Word Pro: VertAlign property

```
{button ,AL(^H_LAYOUT_CLASS;H_CELLGROUPLAYOUT_CLASS;H_CELLLAYOUT_CLASS;H_CONNECTEDLAY  
OUT_CLASS;H_COLUMNGROUPLAYOUT_CLASS;H_FOOTERLAYOUT_CLASS;H_FRAMELAYOUT_CLASS;H  
_FRAMEGROUPLAYOUT_CLASS;H_DROP_CAPLAYOUT_CLASS;H_GROUPLAYOUT_CLASS;H_HEADERLAY  
OUT_CLASS;H_NOTELAYOUT_CLASS;H_PAGELAYOUT_CLASS;H_ROWGROUPLAYOUT_CLASS;H_ROWLA  
YOUT_CLASS;H_RUBYLAYOUT_CLASS;H_SUPERTABLEGROUPLAYOUT_CLASS;H_SUPERTABLELAYOUT  
_CLASS;H_TABLELAYOUT_CLASS;H_ENDNOTELAYOUT_CLASS;H_FOOTNOTELAYOUT_CLASS;H_TABLE  
HEADINGLAYOUT_CLASS;H_TOCSUPERTABLELAYOUT_CLASS',0)} See list of classes
```

(Read-write) Sets the vertical alignment of a layout object.

Data Type

Data type is [Variant](#), which allows the value of this property to be one of the constants listed below or its numeric equivalent (in parentheses).

Syntax

vertalignvalue = [objectreference].VertAlign

[objectreference].VertAlign = vertalignvalue

Legal values

<u>Value</u>	<u>Effect</u>
\$LtsAlignmentTop (1056964614)	Sets the vertical alignment of a layout object to the top.
\$LtsAlignmentVertCenter (1056964615)	Sets the vertical alignment of a layout object to the center.
\$LtsAlignmentBottom (1056964616)	Sets the vertical alignment of a layout object to the bottom.

Usage

Equivalent to the "Vertical alignment" option on the Misc panel of the InfoBox for certain layout objects.

Word Pro: VerticalSplitWindow property

{button ,AL('H_USERINTERFACEPREFS_CLASS',0)} [See list of classes](#)

{button ,AL('H_VERTICALSPLITWINDOW_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Sets or returns whether Word Pro will split the document area vertically when the next document is opened or created.

Data Type

[Integer](#)

Syntax

verticalsplitwindowvalue = [objectreference].VerticalSplitWindow

[objectreference].VerticalSplitWindow = verticalsplitwindowvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values. Default is False (0).

Usage

Word Pro automatically sets this property to False after a document is opened or created in a split window.

Word Pro: VertRuler property

{button ,AL('H_APPLICATIONWINDOW_CLASS',0)} [See list of classes](#)

{button ,AL('H_VERTRULER_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-only) Contains a ruler object that indicates tab settings, indents, margins, and columns.

Data Type

[Ruler](#)

Syntax

verrulervalue = [objectreference].VertRuler

Legal values

Always contains an instance of the Ruler class.

Usage

Use this property to access the vertical ruler for the document.

Word Pro: VertScrollBarVisible property

{button ,AL(^H_DOCWINDOW_CLASS',0)} [See list of classes](#)

(Read-only)

Data Type

[Integer](#)

Syntax

vertscrollbarvisiblevalue = [objectreference].VertScrollBarVisible

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Word Pro: ViewLevel property

{button ,AL('H_PRINTSETTINGS_CLASS;H_DOCWINDOW_CLASS',0)} [See list of classes](#)

{button ,AL('H_VIEWLEVEL_PROPERTY_EXSCRIPT',1)} [See example](#)

[PrintSettings]

(Read-write) Allows you to print a document at different zoom levels.

[DocWindow]

(Read-write) Allows you to display a document at different zoom levels.

Data Type

Integer

Syntax

viewlevelvalue = [objectreference].ViewLevel

[objectreference].ViewLevel = viewlevelvalue

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

The values in this property represent percentages. For example, a value of 100 displays or prints the actual size of the document. A value of 200 represents 200%, and will print or display at twice the normal size.

Word Pro: ViewType property

{button ,AL('H_PRINTSETTINGS_CLASS;H_DOCWINDOW_CLASS;H_WINVIEWPREFS_CLASS',0)} [See list of classes](#)

{button ,AL('H_VIEWTYPE_PROPERTY_EXSCRIPT',1)} [See example](#)

[PrintSettings]

(Read-write) Allows you to print a specific view of a document.

[DocWindow]

(Read-write) Allows you to display the document in draft, layout, or outline view.

Data Type

Data type is Variant which allows the value of this property to be one of the constants listed below or its numeric equivalent (in parentheses).

Syntax

viewtypevalue = [objectreference].ViewType

[objectreference].ViewType = viewtypevalue

Legal values

[PrintSettings]

The value of this property must be one of the following strings or its numeric equivalent (in parentheses) when this property is found on a PrintSettings object.

<u>Value</u>	<u>Effect</u>
\$LwpPresentationTypeLayout (1646)	The document is printed in Layout view.
\$LwpPresentationTypeOutline (1649)	The document is printed in Outline view.

[DocWindow]

<u>Value</u>	<u>Effect</u>
\$LwpPresentationTypeDraft (1647)	The document is displayed in Draft view.
\$LwpPresentationTypeLayout (1646)	The document is displayed in Layout view.
\$LwpPresentationTypeOutline (1649)	The document is displayed in Outline view.

[WinViewPrefs]

The value of this property must be one of the following strings or its hexadecimal equivalent (in parentheses) when this property is found on a WinViewPrefs object.

<u>Value</u>	<u>Effect</u>
LwpViewsStandard (&H1)	The standard view (100%) is displayed.
LwpViewsEnlarged (&H2)	The enlarged view (150%) is displayed.
LwpViewsFullpage (&H4)	The full page view is displayed.
LwpViewsCustom (&H8)	The custom zoom level view is displayed.
LwpViewsFacingpages (&H10)	The facing pages view is displayed.
LwpViewsPagesort (&H20)	The page sorter view is displayed.
LwpViewsMultiplepages (&H40)	The multiple page view is displayed.
LwpViewsPagemargin (&H80)	The zoom to page margin view is displayed.
LwpViewsPagewidth (&H100)	The zoom to page width view is displayed.
LwpViewsReduced (&H200)	The reduced zoom view (75%) is displayed.
LwpViewsMoreenlarged (&H400)	The enlarged view (200%) is displayed.

Usage

[WinViewPrefs]

The legal values for this property on a WinViewPrefs object are derived from a bitmask. This bitmask allows Word Pro

to use one or more of the view types listed above.

{button ,AL('H_CUSTOMVIEWLEVEL_PROPERTY_MEMDEF',0)} [See related topics](#)

Word Pro: Visible property

{button ,AL(^H_APPLICATION_CLASS;H_WPAPPLICATION_CLASS;H_WINDOW_CLASS;H_APPLICATIONWINDOW_CLASS;H_STATUSBAR_CLASS;H_DOCWINDOW_CLASS',0)} [See list of classes](#)

{button ,AL(^H_VISIBLE_PROPERTY_EXSCRIPT',1)} [See example](#)

[Application]

(Read-Write) Determines whether or not the Word Pro application window is visible to the user.

[Window]

(Read-only) Returns a boolean value indicating whether or not the object from which you called this property is visible to the user.

Data Type

[Integer](#)

Syntax

visiblevalue = [objectreference].Visible

Legal values

The value of this property cannot be set by a script. The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

You can call this property from WPAplication, ApplicationWindow, DocWindow, or StatusBar.

Use this property to determine whether Word Pro, the application workspace, a document window, or the status bar is currently visible to the user.

Word Pro: WasDeletedInRevMarkMode property

```
{button ,AL('H_LAYOUT_CLASS;H_CELLGROUPLAYOUT_CLASS;H_CELLLAYOUT_CLASS;H_CONNECTEDLAY  
OUT_CLASS;H_COLUMNGROUPLAYOUT_CLASS;H_FOOTERLAYOUT_CLASS;H_FRAMELAYOUT_CLASS;H  
_FRAMEGROUPLAYOUT_CLASS;H_DROP_CAPLAYOUT_CLASS;H_GROUPLAYOUT_CLASS;H_HEADERLAY  
OUT_CLASS;H_NOTELAYOUT_CLASS;H_PAGELAYOUT_CLASS;H_ROWGROUPLAYOUT_CLASS;H_ROWLA  
YOUT_CLASS;H_RUBYLAYOUT_CLASS;H_SUPERTABLEGROUPLAYOUT_CLASS;H_SUPERTABLELAYOUT  
_CLASS;H_TABLELAYOUT_CLASS;H_ENDNOTELAYOUT_CLASS;H_FOOTNOTELAYOUT_CLASS;H_TABLE  
HEADINGLAYOUT_CLASS;H_TOCSUPERTABLELAYOUT_CLASS',0)} See list of classes
```

```
{button ,AL('H_WASDELETEDINREVMARKMODE_PROPERTY_EXSCRIPT',1)} See example
```

(Read-write) Indicates whether or not a layout object was deleted while in revision marking mode.

Data Type

[Integer](#)

Syntax

wasdeletedinrevmarkmodevalue = [objectreference].WasDeletedInRevMarkMode

[objectreference].WasDeletedInRevMarkMode = wasdeletedinrevmarkmodevalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Word Pro: WasInsertedInRevMarkMode property

```
{button ,AL('H_LAYOUT_CLASS;H_CELLGROUPLAYOUT_CLASS;H_CELLLAYOUT_CLASS;H_CONNECTEDLAY  
OUT_CLASS;H_COLUMNGROUPLAYOUT_CLASS;H_FOOTERLAYOUT_CLASS;H_FRAMELAYOUT_CLASS;H  
_FRAMEGROUPLAYOUT_CLASS;H_DROP_CAPLAYOUT_CLASS;H_GROUPLAYOUT_CLASS;H_HEADERLAY  
OUT_CLASS;H_NOTELAYOUT_CLASS;H_PAGELAYOUT_CLASS;H_ROWGROUPLAYOUT_CLASS;H_ROWLA  
YOUT_CLASS;H_RUBYLAYOUT_CLASS;H_SUPERTABLEGROUPLAYOUT_CLASS;H_SUPERTABLELAYOUT  
_CLASS;H_TABLELAYOUT_CLASS;H_ENDNOTELAYOUT_CLASS;H_FOOTNOTELAYOUT_CLASS;H_TABLE  
HEADINGLAYOUT_CLASS;H_TOCSUPERTABLELAYOUT_CLASS',0)} See list of classes
```

```
{button ,AL('H_WASINSERTEDINREVMARKMODE_PROPERTY_EXSCRIPT',1)} See example
```

(Read-write) Indicates whether or not a layout object was inserted while in revision marking mode.

Data Type

[Integer](#)

Syntax

wasinsertedinrevmarkmodevalue = [objectreference].WasInsertedInRevMarkMode

[objectreference].WasInsertedInRevMarkMode = wasinsertedinrevmarkmodevalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Word Pro: WaterMarkName property

{button ,AL('H_GRAPHICOLEBJECT_CLASS;H_GRAPHIC_CLASS;H_OLEOBJECT_CLASS',0)} [See list of classes](#)

(Read-only)

Data Type

String

Syntax

watermarknamevalue = [objectreference].WaterMarkName

Legal values

Unknown

Usage

Word Pro: WaterMarksPath property

{button ,AL(^H_USERINTERFACEPREFS_CLASS',0)} [See list of classes](#)

{button ,AL(^H_WATERMARKSPATH_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-only) Indicates the default path (drive and directory) for watermark graphics.

Data Type

[String](#)

Syntax

watermarkspathpathvalue = [objectreference].WaterMarksPath

Legal values

A valid path, including drive and directory.

Usage

There is no equivalent to this property in the Word Pro user interface. The WaterMarksPath property defaults to a value that is calculated as follows:

```
Word Pro executable path + "\graphicspath" + "\backgrnd"
```

For example, if Word Pro is installed in a directory called "C:\Lotus\WordPro," the default WaterMarksPath property value is "C:\Lotus\WordPro\graphicspath\backgrnd."

Word Pro: Weight property

{button ,AL(^H_FONTMETRICS_CLASS',0)} [See list of classes](#)

(Read-only)

Data Type

Integer

Syntax

weightvalue = [objectreference].Weight

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Word Pro: Where property

{button ,AL(^H_LAYOUT_CLASS;H_CELLGROUPLAYOUT_CLASS;H_CELLLAYOUT_CLASS;H_CONNECTEDLAYOUT_CLASS;H_COLUMNGROUPLAYOUT_CLASS;H_FOOTERLAYOUT_CLASS;H_FRAMELAYOUT_CLASS;H_FRAMEGROUPLAYOUT_CLASS;H_DROPDOWNLAYOUT_CLASS;H_GROUPLAYOUT_CLASS;H_HEADERLAYOUT_CLASS;H_NOTELAYOUT_CLASS;H_PAGELAYOUT_CLASS;H_ROWGROUPLAYOUT_CLASS;H_ROWLAYO
UT_CLASS;H_RUBYLAYOUT_CLASS;H_SUPERTABLEGROUPLAYOUT_CLASS;H_SUPERTABLELAYOUT
_CLASS;H_TABLELAYOUT_CLASS;H_ENDNOTELAYOUT_CLASS;H_FOOTNOTELAYOUT_CLASS;H_TABLE
HEADINGLAYOUT_CLASS;H_TOCSUPERTABLELAYOUT_CLASS;H_FINDANDREPLACE_CLASS',0)} [See list
of classes](#)

{button ,AL(^H_WHERE_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write)

[Layout]

Indicates where an anchor is connected to a parent layout object.

[FindandReplace]

Indicates where a Find & Replace text or character string is located in the document.

Data Type

Data type is [Variant](#), which allows the value of this property to be one of the constants listed below or its numeric equivalent (in parentheses).

Syntax

wherevalue = [objectreference].Where

[objectreference].Where = wherevalue

Legal values

[Layout]

<u>Value</u>	<u>Effect</u>
\$LwpWhereTypeLayoutUpperRight (1978)	Connects the anchor to the upper right corner of the parent layout object.
\$LwpWhereTypeLowerLeft (1981)	Connects the anchor to the lower left corner of the parent layout object.
\$LwpWhereTypeLowerRight (1983)	Connects the anchor to the lower right corner of the parent layout object.
\$LwpWhereTypeMiddle (1984)	Connects the anchor to the middle of the parent layout object.
\$LwpWhereTypeMiddleBottom (1982)	Connects the anchor to the middle of the bottom corner of the parent layout object.
\$LwpWhereTypeMiddleLeft (1979)	Connects the anchor to the middle of the left corner of the parent layout object.
\$LwpWhereTypeMiddleRight (1980)	Connects the anchor to the middle of the right corner of the parent layout object.
\$LwpWhereTypeMiddleTop (1977)	Connects the anchor to the middle of the top corner of the parent layout object.
\$LwpWhereTypeUpperLeft (1976)	Connects the anchor to the upper left corner of the parent layout object.

[FindandReplace]

<u>Value</u>	<u>Effect</u>
\$LwpLookWhereCurrentDivision (576)	Search the current division.
\$LwpLookWhereCurrentSection (577)	Search the current section.
\$LwpLookWhereCurrentSelection (578)	Search highlighted text.
\$LwpLookWhereEntireDocument (575)	Search entire document.

Usage

[FindandReplace]

Use this property to search a specific area in Find & Replace. Equivalent to choosing Edit - Find & Replace Text, clicking Options, and choosing from the "Look in" box in the "Find & replace scope" section.

Word Pro: WidowOrphan property

{button ,AL('H_OPTIONS_CLASS',0)} [See list of classes](#)

{button ,AL('H_WIDOWORPHAN_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Determines whether or not Word Pro allows widowed or orphaned text.

Data Type

[Integer](#)

Syntax

widoworphanvalue = [objectreference].WidowOrphan

[objectreference].WidowOrphan = widoworphanvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

This property is equivalent to the "Widow/Orphan control" option in the Document Properties dialog box.

A widow occurs when the first line of a paragraph displays as the last line of a page or column. An orphan occurs when the last line of a paragraph displays as the first line of a page or column.

Word Pro: WidthAbove property

{button ,AL('H_CHARACTERBORDER_CLASS;H_PARAGRAPHBORDER_CLASS',0)} [See list of classes](#)

{button ,AL('H_WIDTHABOVE_PROPERTY_EXSCRIPT',1)} [See example](#)

[ParagraphBorder]

(Read-write) Determines the line length above a paragraph.

Data Type

Long

Syntax

widthabovevalue = [objectreference].WidthAbove

[objectreference].WidthAbove = widthabovevalue

Legal values

Data type is Long but the unit of measurement used for this property is Twips. There are 1440 Twips per inch.

Usage

[ParagraphBorder]

If the TypeAbove property is set to \$LwpParaBorderWidthOther, this property controls the length of the line above the paragraph.

Word Pro: WidthBelow property

{button ,AL('H_CHARACTERBORDER_CLASS;H_PARAGRAPHBORDER_CLASS',0)} [See list of classes](#)

{button ,AL('H_WIDTHBELOW_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) The width/length of lines below a paragraph, if the value of the TypeBelow property is set to "Other."

Data Type

Long

Syntax

widthbelowvalue = [objectreference].WidthBelow

[objectreference].WidthBelow = widthbelowvalue

Legal values

Data type is Long, but the unit of measurement used for this property is Twips. There are 1440 Twips per inch.

Usage

Equivalent to choosing "Custom" as the "Line length" on the InfoBox, and then specifying the line length/position in the Line Length dialog box.

Word Pro: WidthBetween property

{button ,AL('H_PARAGRAPHBORDER_CLASS','0')} [See list of classes](#)

{button ,AL('H_WIDTHBETWEEN_PROPERTY_EXSCRIPT',1')} [See example](#)

(Read-write)

This language element has not yet been defined. You can check for a newer Help file on the Lotus web site.

Data Type

Long

Syntax

widthbetweenvalue = [objectreference].WidthBetween

[objectreference].WidthBetween = widthbetweenvalue

Legal values

Any numeric value of data type Long. The value represents the number of twips. There are 1440 Twips per inch.

Usage

Word Pro: WidthInTwips property

{button ,AL('H_BORDER_CLASS',0)} [See list of classes](#)

{button ,AL('H_WIDTHINTWIPS_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write)

Data Type

Long

Syntax

widthintwipsvalue = [objectreference].WidthInTwips

[objectreference].WidthInTwips = widthintwipsvalue

Legal values

Data type is Long but the unit of measurement used for this property is Twips. There are 1440 Twips per inch.

Usage

Word Pro: Width property

{button ,AL('H_WINDOW_CLASS;H_APPLICATIONWINDOW_CLASS;H_STATUSBAR_CLASS;H_DOCWINDOW_CLASS;H_BASECONTAINER_CLASS;H_CELLCONTAINER_CLASS;H_FRAMECONTAINER_CLASS;H_NOTECONTAINER_CLASS;H_RUBYCONTAINER_CLASS;H_DROPDOWNCONTAINER_CLASS;H_PAGECONTAINER_CLASS;H_SUBPAGECONTAINER_CLASS;H_SUPERPAGECONTAINER_CLASS;H_ROWCONTAINER_CLASS;H_SUPERTABLECONTAINER_CLASS;H_TABLECONTAINER_CLASS;H_PARALLELCOLSCONTAINER_CLASS;H_TABLEONLYCONT_CLASS;H_FONT_CLASS;H_GRAPHICOLEBJECT_CLASS;H_GRAPHIC_CLASS;H_OLEOBJECT_CLASS;H_ICONBAR_CLASS;H_LAYOUT_CLASS;H_CELLGROUPLAYOUT_CLASS;H_CELLLAYOUT_CLASS;H_CONNECTEDLAYOUT_CLASS;H_COLUMNGROUPLAYOUT_CLASS;H_FOOTERLAYOUT_CLASS;H_FRAMELAYOUT_CLASS;H_FRAMEGROUPLAYOUT_CLASS;H_DROPDOWNLAYOUT_CLASS;H_GROUPLAYOUT_CLASS;H_HEADERLAYOUT_CLASS;H_NOTELAYOUT_CLASS;H_PAGELAYOUT_CLASS;H_ROWGROUPLAYOUT_CLASS;H_ROWLAYOUT_CLASS;H_RUBYLAYOUT_CLASS;H_SUPERTABLEGROUPLAYOUT_CLASS;H_SUPERTABLELAYOUT_CLASS;H_TABLELAYOUT_CLASS;H_ENDNOTELAYOUT_CLASS;H_FOOTNOTELAYOUT_CLASS;H_TABLEHEADINGLAYOUT_CLASS;H_TOCSUPERTABLELAYOUT_CLASS;H_BORDER_CLASS',0)} [See list of classes](#)

{button ,AL('H_WIDTH_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write)

[IconBar]

Used to represent the actual width of an icon bar object; the actual width of the entire object.

[Layout]

Allows you to set or return the actual width of a layout object.

[StatusBar]

Used in the StatusBar class to return the actual width of the status bar object in Twips.

[ApplicationWindow]

The actual width of the application window.

[BaseContainer]

Allows you to set or return the total width of a container from the outside border edge to the outside border edge.

Data Type

[Font]

[Single](#)

[All others]

[Long](#)

Syntax

widthvalue = [objectreference].Width

[objectreference].Width = widthvalue

Legal values

[Font]

Represents the actual width in points; a single precision floating point value.

[All others]

Data type is Long but the unit of measurement used for this property is [Twips](#). There are 1440 Twips per inch.

Usage

[ApplicationWindow]

Used to manipulate the width of the application window.

[IconBar]

Used to represent the actual width of an icon bar object. Returns the distance in Twips from the left border of an icon bar object to its right border; the actual width of the entire object.

[StatusBar]

This property appears with data type Long in the Window class.

[Layout]

The width of a layout object is the distance between the left border and right border of the layout object.

Word Pro: WindowId property

{button ,AL(^H_DOCWINDOW_CLASS',0)} [See list of classes](#)

(Read-only)

Data Type

Long

Syntax

windowidvalue = [objectreference].WindowId

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Word Pro: WindowsName property

{button ,AL('H_FONT_CLASS',0)} [See list of classes](#)

{button ,AL('H_WINDOWSNAME_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) The name of the font that is assigned to the specified font object.

Data Type

[String](#)

Syntax

windowsnamevalue = [objectreference].WindowsName

[objectreference].WindowsName = windowsnamevalue

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

If the specified font is not available for Word Pro to use, you can check the ActualName property for the name of the font that is being substituted.

Word Pro: WinViewPrefs property

{button ,AL('H_USERINTERFACEPREFS_CLASS;H_DOCWINDOW_CLASS',0)} [See list of classes](#)

{button ,AL('H_WINVIEWPREFS_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-only) Returns the view preferences object of the specified DocWindow or UserInterfacePrefs object. See the Usage section for more information.

Data Type

[WinViewPrefs](#)

Syntax

winviewprefsvalue = [objectreference].WinViewPrefs

Legal values

Always contains an instance of the WinViewPrefs class.

Usage

The WinViewPrefs property of the DocWindow class

The WinViewPrefs object that is stored within the DocWindow class represents the current view preference settings within a specific document. You can use this object to determine a document's current custom view level, or whether tab marks are being displayed within the document.

The WinViewPrefs property of the UserInterfacePrefs class

The WinViewPrefs object that is stored within the UserInterfacePrefs class represents the current default view preference settings within Word Pro. You can use this object to determine the default custom view level or whether a document's margins will display in color by default.

Changes that are made to the winviewprefs object of the UserInterfacePrefs class affect all future documents that are opened or created.

Word Pro: WithComments property

{button ,AL('H_PRINTSETTINGS_CLASS',0)} [See list of classes](#)

{button ,AL('H_WITHCOMMENTS_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Allows you to print each comment note at its location in the document.

Data Type

[Integer](#)

Syntax

withcommentsvalue = [objectreference].WithComments

[objectreference].WithComments = withcommentsvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

A comment note must be open in a document before you can print it. Equivalent to choosing File - Print, clicking Options, and selecting "With comments" in the "Print options" section of the Print Options dialog box.

Word Pro: WordDoubleUnderline property

{button ,AL(^H_FONT_CLASS',0)} [See list of classes](#)

{button ,AL(^H_WORDDOUBLEUNDERLINE_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write)

Data Type

[Integer](#)

Syntax

worddoubleunderlinevalue = [objectreference].WordDoubleUnderline

[objectreference].WordDoubleUnderline = worddoubleunderlinevalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Word Pro: WordProObjectName property

{button ,AL('H_OLECONTROL_CLASS',0)} [See list of classes](#)

(Read-only)

The complete hierarchical name of an embedded OLE control.

Data Type

String

Syntax

From within an OLE control's event script:

```
wordproobjectnamevalue = Source.WordProObjectName
```

Legal values

This property is read-only. The value of this property cannot be set by a script.

Usage

The only time this property is available on an OLE control is when the control is embedded in a Word Pro document and you are calling the control through the Source argument on the control's event script. For example:

For example, the event script below is for the Click event on a LotusCommandButton control.

```
Sub Click(Source as Lotuscommandbutton)
    MsgBox Source.WordProObjectName
End Sub
```

When you click the LotusCommandButton which contains this event script, Word Pro passes the OLE control object to this script through the Source argument. In addition to the usual properties and methods provided by the developer of the OLE control, Word Pro provides the WordProObjectName property.

Note You do not need this property to access an OLE control.

Word Pro: WordUnderline property

{button ,AL('H_FONT_CLASS',0)} [See list of classes](#)

{button ,AL('H_WORDUNDERLINE_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write)

Data Type

[Integer](#)

Syntax

wordunderlinevalue = [objectreference].WordUnderline

[objectreference].WordUnderline = wordunderlinevalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Word Pro: WorkingType property

{button ,AL(`H_USERINTERFACEPREFS_CLASS',0)} [See list of classes](#)

{button ,AL(`H_WORKINGTYPE_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Sets or returns the value that determines whether Word Pro defaults to the type of document last opened.

Data Type

[Integer](#)

Syntax

workingtypevalue = [objectreference].WorkingType

[objectreference].WorkingType = workingtypevalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values. Default is False (0).

Usage

Equivalent to the "Use working type" option on the Locations panel of the Word Pro Preferences dialog box. If this property is set to True (-1), Word Pro continues to display files of the type you most recently chose in the Open or Save dialog box. If set to False (0), Word Pro uses ".LWP" as the default file type.

{button ,AL(`H_USERDEFINEDFILTER_PROPERTY_MEMDEF',0)} [See related topics](#)

Word Pro: WPDataSets property

```
{button ,AL('H_WPAPPLICATION_CLASS;H_CHARACTERSTYLE_CLASS;H_DIVISION_CLASS;H_LAYOUT_CLASSES;H_CELLGROUPLAYOUT_CLASS;H_CELLLAYOUT_CLASS;H_CONNECTEDLAYOUT_CLASS;H_COLUMNNGROUPLAYOUT_CLASS;H_FOOTERLAYOUT_CLASS;H_FRAMELAYOUT_CLASS;H_FRAMEGROUPLAYOUT_CLASS;H_DROPCAPLAYOUT_CLASS;H_GROUPLAYOUT_CLASS;H_HEADERLAYOUT_CLASS;H_NOTELAYOUT_CLASS;H_PAGELAYOUT_CLASS;H_ROWGROUPLAYOUT_CLASS;H_ROWLAYOUT_CLASS;H_RUBYLAYOUT_CLASS;H_SUPERTABLEGROUPLAYOUT_CLASS;H_SUPERTABLELAYOUT_CLASS;H_TABLELAYOUT_CLASS;H_ENDNOTELAYOUT_CLASS;H_FOOTNOTELAYOUT_CLASS;H_TABLEHEADINGLAYOUT_CLASS;H_TOCSUPERTABLELAYOUT_CLASS;H_MARKER_CLASS;H_POWERFIELD_CLASS;H_CLICKHERE_CLASS;H_TEXTMARKER_CLASS;H_RUBYMARKER_CLASS;H_TABLEMARKER_CLASS;H_PARAGRAPHSTYLE_CLASS;H_TEXT_CLASS;H_TEXTDOCUMENT_CLASS',0)} See list of classes
```

```
{button ,AL('H_WPDATASETS_PROPERTY_EXSCRIPT',1)} See example
```

(Read-only) This property contains an instance of the WPDataSetCollection class.

Data Type

[WPDataSetCollection](#)

Syntax

wpdatasetsvalue = [objectreference].WPDataSets

Legal values

Always contains an instance of the WPDataSetCollection class.

Usage

This WPDataSetCollection object gives you access to WPDataSet objects. For more information, see the WPDataSetCollection and WPDataSet classes.

```
{button ,AL('H_WPDATASET_CLASS;H_WPDATASETCOLLECTION_CLASS',0)} See related topics
```

Word Pro: WrapType property

```
{button ,AL('H_LAYOUT_CLASS;H_CELLGROUPLAYOUT_CLASS;H_CELLAYOUT_CLASS;H_CONNECTEDLAYOUT_CLASS;H_COLUMNGROUPLAYOUT_CLASS;H_FOOTERLAYOUT_CLASS;H_FRAMELAYOUT_CLASS;H_FRAMEGROUPLAYOUT_CLASS;H_DROPDOWNLAYOUT_CLASS;H_GROUPLAYOUT_CLASS;H_HEADERLAYOUT_CLASS;H_NOTELAYOUT_CLASS;H_PAGELAYOUT_CLASS;H_ROWGROUPLAYOUT_CLASS;H_ROWLAYO  
UT_CLASS;H_RUBYLAYOUT_CLASS;H_SUPERTABLEGROUPLAYOUT_CLASS;H_SUPERTABLELAYOUT  
_CLASS;H_TABLELAYOUT_CLASS;H_ENDNOTELAYOUT_CLASS;H_FOOTNOTELAYOUT_CLASS;H_TABLE  
HEADINGLAYOUT_CLASS;H_TOCSUPERTABLELAYOUT_CLASS',0)} See list of classes
```

```
{button ,AL('H_WRAPTYPE_PROPERTY_EXSCRIPT',1)} See example
```

(Read-write) Specifies how text should wrap around a layout object.

Data Type

Data type is [Variant](#), which allows the value of this property to be one of the constants listed below or its numeric equivalent (in parentheses).

Syntax

wrapypevalue = [objectreference].WrapType

[objectreference].WrapType = wrapypevalue

Legal values

<u>Value</u>	<u>Effect</u>
\$LwpWrapLayoutWrapRight (1999)	Wraps text to the right of a layout object.
\$LwpWrapTypeLayoutNoWrapAround (1997)	Prevents text from wrapping around a layout object; allows text to flow behind a layout object.
\$LwpWrapTypeLayoutNoWrapBeside (1996)	Wraps text above and below a layout object.
\$LwpWrapTypeLayoutWrapAround (1995)	Wraps text around a layout object.
\$LwpWrapTypeLayoutWrapLeft (1998)	Wraps text to the left of a layout object.
\$LwpWrapTypeLayoutWrapRight (2070)	Wraps text to the right of a layout object.

Usage

Equivalent to the "Wrap options" setting on the Placement panel of the InfoBox for frame or table layout objects.

Word Pro: XOffset property

{button ,AL('H_LAYOUT_CLASS;H_CELLGROUPLAYOUT_CLASS;H_CELLLAYOUT_CLASS;H_CONNECTEDLAYOUT_CLASS;H_COLUMNGROUPLAYOUT_CLASS;H_FOOTERLAYOUT_CLASS;H_FRAMELAYOUT_CLASS;H_FRAMEGROUPLAYOUT_CLASS;H_DROP_CAPLAYOUT_CLASS;H_GROUPLAYOUT_CLASS;H_HEADERLAYOUT_CLASS;H_NOTELAYOUT_CLASS;H_PAGELAYOUT_CLASS;H_ROWGROUPLAYOUT_CLASS;H_ROWLAYOUT_CLASS;H_RUBYLAYOUT_CLASS;H_SUPERTABLEGROUPLAYOUT_CLASS;H_SUPERTABLELAYOUT_CLASS;H_TABLELAYOUT_CLASS;H_ENDNOTELAYOUT_CLASS;H_FOOTNOTELAYOUT_CLASS;H_TABLEHEADINGLAYOUT_CLASS;H_TOCSUPERTABLELAYOUT_CLASS;H_DOCWINDOW_CLASS',0)} [See list of classes](#)

{button ,AL('H_XOFFSET_PROPERTY_EXSCRIPT',1)} [See example](#)

[DocWindow]

(Read-only) Indicates the distance from the origin of the DocWindow object to the left edge of the page within the DocWindow.

[Layout]

(Read-write) Horizontally offsets the content of a layout without requiring a change in margins.

Data Type

Long

Syntax

xoffsetvalue = [objectreference].XOffset

[objectreference].XOffset = xoffsetvalue

Legal values

Data type is Long, but the unit of measurement used for this property is Twips. There are 1440 Twips per inch.

Usage

[DocWindow]

Indicates the amount of space, in Twips, between the left edge of the page and the origin of the DocWindow. The DocWindow is the window in which a document is displayed in Word Pro. Its origin is the top left corner of the window.

[Layout]

Used for cropping graphics in a layout object.

{button ,AL('H_ISPOINTWITHIN_METHOD_MEMDEF;H_LEFT_PROPERTY_MEMDEF;H_TOP_PROPERTY_MEMDEF',0)} [See related topics](#)

Word Pro: XPosition property

{button ,AL('H_LAYOUT_CLASS;H_CELLGROUPLAYOUT_CLASS;H_CELLLAYOUT_CLASS;H_CONNECTEDLAYOUT_CLASS;H_COLUMNGROUPLAYOUT_CLASS;H_FOOTERLAYOUT_CLASS;H_FRAMELAYOUT_CLASS;H_FRAMEGROUPLAYOUT_CLASS;H_DROP_CAPLAYOUT_CLASS;H_GROUPLAYOUT_CLASS;H_HEADERLAYOUT_CLASS;H_NOTELAYOUT_CLASS;H_PAGELAYOUT_CLASS;H_ROWGROUPLAYOUT_CLASS;H_ROWLAYO
UT_CLASS;H_RUBYLAYOUT_CLASS;H_SUPERTABLEGROUPLAYOUT_CLASS;H_SUPERTABLELAYOUT
_CLASS;H_TABLELAYOUT_CLASS;H_ENDNOTELAYOUT_CLASS;H_FOOTNOTELAYOUT_CLASS;H_TABLE
HEADINGLAYOUT_CLASS;H_TOCSUPERTABLELAYOUT_CLASS;H_SHADOW_CLASS',0)} [See list of classes](#)

{button ,AL('H_XPOSITION_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write)

[Layout]

The X coordinate for the origin of this layout.

[Shadow]

Determines the X axis depth of a shadow object for text, frames, tables, headers, footers, and pages in a document.

Data Type

Long

Syntax

xpositionvalue = [objectreference].XPosition

[objectreference].XPosition = xpositionvalue

Legal values

Data type is Long, but the unit of measurement used for this property is Twips. There are 1440 Twips per inch.

Usage

[Shadow]

Used to set the depth of a shadow object from left to right for text, frames, tables, headers, footers, and pages in a document.

Word Pro: YOffset property

{button ,AL(`H_LAYOUT_CLASS;H_CELLGROUPLAYOUT_CLASS;H_CELLLAYOUT_CLASS;H_CONNECTEDLAY
OUT_CLASS;H_COLUMNGROUPLAYOUT_CLASS;H_FOOTERLAYOUT_CLASS;H_FRAMELAYOUT_CLASS;H
_FRAMEGROUPLAYOUT_CLASS;H_DROP_CAPLAYOUT_CLASS;H_GROUPLAYOUT_CLASS;H_HEADERLAY
OUT_CLASS;H_NOTELAYOUT_CLASS;H_PAGELAYOUT_CLASS;H_ROWGROUPLAYOUT_CLASS;H_ROWLA
YOUT_CLASS;H_RUBYLAYOUT_CLASS;H_SUPERTABLEGROUPLAYOUT_CLASS;H_SUPERTABLELAYOUT
_CLASS;H_TABLELAYOUT_CLASS;H_ENDNOTELAYOUT_CLASS;H_FOOTNOTELAYOUT_CLASS;H_TABLE
HEADINGLAYOUT_CLASS;H_TOCSUPERTABLELAYOUT_CLASS;H_DOCWINDOW_CLASS',0)} [See list of
classes](#)

{button ,AL(`H_YOFFSET_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Vertically offsets the content of a layout without requiring a change in margins.

Data Type

[Long](#)

Syntax

yoffsetvalue = [objectreference].YOffset

[objectreference].YOffset = yoffsetvalue

Legal values

Data type is Long, but the unit of measurement used for this property is [Twips](#). There are 1440 Twips per inch.

Usage

Used for cropping graphics in a layout object.

{button ,AL(`H_ISPOINTWITHIN_METHOD_MEMDEF;H_LEFT_PROPERTY_MEMDEF;H_TOP_PROPERTY_MEM
DEF',0)} [See related topics](#)

Word Pro: YPosition property

```
{button ,AL('H_LAYOUT_CLASS;H_CELLGROUPLAYOUT_CLASS;H_CELLLAYOUT_CLASS;H_CONNECTEDLAYOUT_CLASS;H_COLUMNGROUPLAYOUT_CLASS;H_FOOTERLAYOUT_CLASS;H_FRAMELAYOUT_CLASS;H_FRAMEGROUPLAYOUT_CLASS;H_DROP_CAP_LAYOUT_CLASS;H_GROUPLAYOUT_CLASS;H_HEADERLAYOUT_CLASS;H_NOTELAYOUT_CLASS;H_PAGELAYOUT_CLASS;H_ROWGROUPLAYOUT_CLASS;H_ROWLAYOUT_CLASS;H_RUBY_LAYOUT_CLASS;H_SUPERTABLEGROUPLAYOUT_CLASS;H_SUPERTABLELAYOUT_CLASS;H_TABLELAYOUT_CLASS;H_ENDNOTELAYOUT_CLASS;H_FOOTNOTELAYOUT_CLASS;H_TABLEHEADINGLAYOUT_CLASS;H_TOCSUPERTABLELAYOUT_CLASS;H_SHADOW_CLASS',0)} See list of classes
```

```
{button ,AL('H_YPOSITION_PROPERTY_EXSCRIPT',1)} See example
```

(Read-write)

[Layout]

The Y coordinate for the origin of this layout.

[Shadow]

Determines the Y axis depth of a shadow object for text, frames, tables, headers, footers, and pages in a document.

Data Type

Long

Syntax

ypositionvalue = [objectreference].YPosition

[objectreference].YPosition = ypositionvalue

Legal values

Data type is Long, but the unit of measurement used for this property is Twips. There are 1440 Twips per inch.

Usage

[Shadow]

Used to set the depth of a shadow object up or down for text, frames, tables, headers, footers, and pages in a document.

Word Pro: Zero property

{button ,AL('H_NUMERICFORMAT_CLASS',0)} [See list of classes](#)

{button ,AL('H_ZERO_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-only) Allows you to modify the "Zero" condition of the current number format.

Data Type

[NumericFormatSubset](#)

Syntax

zerovalue = [objectreference].Zero

Legal values

Always contains an instance of the NumericFormatSubset class.

Usage

Equivalent to the "Zero" condition for edit in the Edit Format dialog box. You can access this dialog box by clicking the Format Options button on the Number Format panel of the InfoBox for cell layout objects.

By accessing the "Zero" condition of a number format, you can modify how zero values appear within table cells, and choose prefix or suffix text.

Word Pro: ZipCode property

{button ,AL('H_PREFERENCES_CLASS',0)} [See list of classes](#)

{button ,AL('H_ZIPCODE_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write)

Data Type

String

Syntax

zipcodevalue = [objectreference].ZipCode

[objectreference].ZipCode = zipcodevalue

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Word Pro: ActivateAs method

{button ,AL(^H_OLEOBJECT_CLASS',0)} [See list of classes](#)

Converts a linked object into a readable format without changing the source object's type.

Note This method is not implemented for the OLEObject class within OS/2.

Syntax

[objectreference].ActivateAs(OldClassID, NewClassID, NewDisplayAsIcon, IconChanged, NewMetaPict)

Parameters

OldClassID

Data type is String.

NewClassID

Data type is String.

NewDisplayAsIcon

Data type is Integer.

IconChanged

Data type is Integer.

NewMetaPict

Data type is Long.

Return value

Integer.

Usage

Equivalent to selecting an OLE object, choosing its menu, choosing Convert, choosing Activate As, and then choosing a different application.

Word Pro: AddACondition method

{button ,AL(^H_MERGEOPTIONS_CLASS',0)} [See list of classes](#)

Adds a merge condition.

Syntax

[objectreference].AddACondition(EntryName, CondOp, CondString, IsExactCase, BoolOp)

Parameters

EntryName

A String parameter indicating which field will be used in the merge condition.

CondOp

An Integer parameter indicating the type of operator. Legal values for this parameter are shown below.

<u>Value</u>	<u>Operator</u>
1	=
2	!=
3	>=
4	>

CondString

A String parameter indicating which value to use in the merge condition.

IsExactCase

An Integer parameter indicating whether the merge condition is case sensitive.

BoolOp

An Integer parameter indicating whether the condition should be combined with other conditions with a boolean AND or a boolean OR.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Equivalent to adding a merge condition in the Merge Records dialog box.

Word Pro: AddNewSectionTabs method

{button ,AL('H_SECTIONTABS_CLASS',0)} [See list of classes](#)

{button ,AL('H_ADDNEWSECTIONTABS_METHOD_EXSCRIPT',1)} [See example](#)

Adds a new quick division to a document.

Syntax

[objectreference].AddNewSectionTabs()

Parameters

None.

Return value

A String value representing the name of the new division.

Usage

Equivalent to right-clicking a Word Pro section tab and choosing "Quick Division".

Word Pro: AddStringToList method

{button ,AL('H_SMARTFILL_CLASS',0)} [See list of classes](#)

{button ,AL('H_ADDSTRINGTOLIST_METHOD_EXSCRIPT',1)} [See example](#)

Adds an item to the specified SmartFill list.

Syntax

[objectreference].AddStringToList (ListName, String)

Parameters

ListName

The name of the SmartFill list to which you want to add an entry.

String

The entry that you want to add to the SmartFill list specified by the ListName parameter.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Word Pro: BinNameFromNumber method

{button ,AL(`H_PRINTMANAGER_CLASS',0)} [See list of classes](#)

{button ,AL(`H_BINNAMEFROMNUMEBR_METHOD_EXSCRIPT',1)} [See example](#)

Syntax

[objectreference].BinNameFromNumber(Number)

Parameters

Number

Data type is Integer.

Return value

String.

Usage

Word Pro: CheckFieldEntries method

{button ,AL(^H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

Syntax

[objectreference].checkfieldentries(bOpenDataFile)

Parameters

bOpenDataFile

Data type is Integer.

Return value

Integer.

Usage

Word Pro: ConvertTo method

{button ,AL('H_OLEOBJECT_CLASS',0)} [See list of classes](#)

This language element is not yet defined.

Note This method is not implemented for the OLEObject class within OS/2.

Syntax

[objectreference].ConvertTo(OldClassID, NewClassID, NewDisplayAsIcon, IconChanged, NewMetaPict)

Parameters

OldClassID

Data type is String.

NewClassID

Data type is String.

NewDisplayAsIcon

Data type is Integer.

IconChanged

Data type is Integer.

NewMetaPict

Data type is Long.

Return value

Integer.

Usage

Word Pro: CreateEmptyList method

{button ,AL('H_SMARTFILL_CLASS',0)} [See list of classes](#)

{button ,AL('H_CREATEEMPTYLIST_METHOD_EXSCRIPT',1)} [See example](#)

This method creates a new SmartFill list with the specified name.

Syntax

[objectreference].CreateEmptyList(ListName, IsCaseExact)

Parameters

ListName

A String parameter representing the name of the new SmartFill list.

IsCaseExact

An Integer parameter specifying whether the list is filled using exact case.

The legal values for this parameter are -1 and 0 but you may use the LotusScript constants True (-1) and False (0) instead of the integer values.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Word Pro: CreateLayer method

```
{button ,AL('H_LAYOUT_CLASS;H_CELLGROUPLAYOUT_CLASS;H_CELLLAYOUT_CLASS;H_CONNECTEDLAY  
OUT_CLASS;H_COLUMNGROUPLAYOUT_CLASS;H_FOOTERLAYOUT_CLASS;H_FRAMELAYOUT_CLASS;H  
_FRAMEGROUPLAYOUT_CLASS;H_DROP_CAPLAYOUT_CLASS;H_GROUPLAYOUT_CLASS;H_HEADERLAY  
OUT_CLASS;H_NOTELAYOUT_CLASS;H_PAGELAYOUT_CLASS;H_ROWGROUPLAYOUT_CLASS;H_ROWLA  
YOUT_CLASS;H_RUBYLAYOUT_CLASS;H_SUPERTABLEGROUPLAYOUT_CLASS;H_SUPERTABLELAYOUT  
_CLASS;H_TABLELAYOUT_CLASS;H_ENDNOTELAYOUT_CLASS;H_FOOTNOTELAYOUT_CLASS;H_TABLE  
HEADINGLAYOUT_CLASS;H_TOCSUPERTABLELAYOUT_CLASS',0)} See list of classes
```

```
{button ,AL('H_CREATELAYER_METHOD_EXSCRIPT',1)} See example
```

Creates a watermark layer within the layout object.

Syntax

[objectreference].createlayer

Parameters

Return value

The return value for this method will always be -1.

Usage

When you call this method, a watermark layer is created for the specified layout object.

Because the return value of CreateLayer is always -1, you cannot tell if a watermark layer is created successfully when you call the method. Use the LayerName property of a layout object to determine if its layer property actually contains a watermark layer object.

To specify a watermark graphic for a layer, use the ImportWatermarkGraphic method.

```
{button ,AL('H_IMPORTWATERMARKGRAPHIC_METHOD_MEMDEF',0)} See related topics
```

Word Pro: CreateOleEmbeddedFile method

{button ,AL('H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL('H_CREATEOLEEMBEDDEDFILE_METHOD_EXSCRIPT',1)} [See example](#)

Creates an embedded OLE object from an existing file. Equivalent to choosing Create - Object, selecting "Create an object from a file," and specifying a file to embed.

Note OLE is not supported in OS/2.

Syntax

[objectreference].CreateOleEmbeddedFile(sClassID,FilePath,IconMetaFilePictHandle)

Parameters

sClassID

A String expression which specifies, in the form of a ClassID or a ProgID, the server application for the file you are embedding. For the purposes of this method, you can let Word Pro determine the proper file type by using the following value:

```
"{00000000-0000-0000-0000-000000000000}"
```

FilePath

A String expression indicating the path and name of the file you are embedding.

IconMetaFilePictHandle

An optional Numeric expression that allows you to specify which icon to use in representing the embedded object in the Word Pro file. Using any value other than 0 automatically tells Word Pro to display the embedded object as an icon. This value serves as a numeric handle (known as the HGLOBAL) to the metafile pict for an icon. You can get the HGLOBAL for a specific metafile pict by using the appropriate Windows API calls. This value must be of type Long. Default is 0, which indicates that you want the contents of the embedded OLE object to be displayed in the Word Pro document.

Caution If you record the process of choosing Create - Object and embedding a file to be displayed as an icon, Word Pro records a value for IconMetaFilePict which is valid only during the recording. When you play back the recorded script, the IconMetaFilePict value is invalid and Word Pro treats the value as if you passed 0.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Word Pro: CreateOleLinkedFile method

{button ,AL('H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL('H_CREATEOLELINKEDFILE_METHOD_EXSCRIPT',1)} [See example](#)

Creates a linked OLE object from an external file. Equivalent to choosing Create - Object, selecting "Create an object from a file," selecting "Link to file," and specifying a file to link.

Note OLE is not supported under OS/2.

Syntax

```
[objectreference].CreateOleLinkedFile(FilePath,[IconMetaFilePictHandle])
```

Parameters

FilePath

A String expression which specifies the file path and name of the file you are linking.

IconMetaFilePictHandle

An optional Numeric expression which allows you to specify which icon to use in representing the linked object in the Word Pro file. Using any value other than 0 automatically tells Word Pro to display the linked object as an icon. This value serves as a numeric handle (known as the HGLOBAL) to the metafile pict for that icon. You can get the HGLOBAL for a specific metafile pict by using the appropriate Windows API calls. This value must be of type Long. Default is 0 which indicates that you want the contents of the linked OLE object to be displayed in the Word Pro document.

Caution If you record the process of choosing Create - Object and linking a file to be displayed as icon, Word Pro records a value for IconMetaFilePict which is valid only during the recording. When you play back the recorded script, the IconMetaFilePict value is invalid and Word Pro treats the value as if you passed 0.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Word Pro: CreateOleNew method

{button ,AL('H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL('H_CREATEOLENEW_METHOD_EXSCRIPT',1)} [See example](#)

Creates a new OLE object and launches the server application. Equivalent to choosing Create - Object, selecting "Create a new object," and specifying type of object to create.

Note OLE is not supported in OS/2.

Syntax

[objectreference].CreateOleNew(sClassID, IconMetaFilePictHandle)

Parameters

sClassID

A String expression which specifies the type of object you are creating. The type of object is expressed as the ClassID or ProgID for the application which creates that type of object. (The application used to create an OLE object is often referred to as the server application.) For example, a Lotus Freelance Presentation has a ClassID of "{CF746000-94FB-101B-8C12-02608C454BFF}" and a ProgID of "FLW3Presentation."

Here are the server application IDs for some other SmartSuite application objects:

1-2-3 Worksheet

ClassID = {00045295-0000-0000-C000-000000000046}

ProgID = 123Worksheet

Launches 1-2-3 and opens an untitled worksheet.

Lotus Approach Report

ClassID = {00028703-0000-0000-C000-000000000046}

ProgID = ApproachReport

Launches Approach and prompts you to select an existing database from which to create the report. Once the database is open, the Report Assistant opens and waits for you to create the report.

Lotus Approach Application

ClassID = {00028701-0000-0000-C000-000000000046}

ProgID = ApproachApplication

Launches Approach and prompts you to select an existing database.

Lotus Freelance Presentation

ClassID = {CF746000-94FB-101B-8C12-02608C454BFF}

ProgID = FLW3Presentation

Launches Freelance and prompts you with the New Presentation dialog box.

Lotus Freelance Drawing

ClassID = {CF746001-94FB-101B-8C12-02608C454BFF}

ProgID = FLW3Drawing

Launches Freelance and opens a new presentation with one blank page.

Lotus ScreenCam Movie 2.1

ClassID = {00041920-0000-0000-C000-000000000046}

ProgID = ScreenCamMovie2

Launches ScreenCam 2.1 and displays the ScreenCam control panel for you to start a recording.

You can find the ClassIDs and ProgIDs for other server applications in the Windows Registry for Windows 3.1 and Windows 95.

IconMetaFilePictHandle

An optional Numeric expression which allows you to specify which icon to use in representing the new OLE object. Using any value other than 0 automatically tells Word Pro to display the new object as an icon. This value serves as a numeric handle (known as the HGLOBAL) to the metafile pict for that icon. You can get the HGLOBAL for a specific metafile pict by using the appropriate Windows API calls. This value must be of type Long. Default is 0, which indicates that you want Word Pro to display the contents of the new OLE object.

Caution If you record the process of creating a new OLE object to be displayed as an icon, Word Pro records a value for IconMetaFilePict which is valid only during the recording. When you play back the recorded script, the IconMetaFilePict value is invalid and Word Pro treats the value as if you passed 0.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Word Pro: CreateParallelColumns method

{button ,AL('H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL('H_CREATEPARALLELOLUMNS_METHOD_EXSCRIPT',1)} [See example](#)

Creates parallel columns at the insertion point in the currently active document.

Syntax

[objectreference].CreateParallelColumns(NumCols, [AlignmentType])

Parameters

NumCols

An Integer value, which specifies the number of parallel columns you want to create.

AlignmentType

Allows you to specify what type of alignment you want to use for the data in your parallel columns. Data type is Variant, which allows the value of this parameter to be one of the nine string values listed below or their numeric equivalents (in parentheses). Default is `$LtsAlignmentHorizCenter`.

<u>Value</u>	<u>Effect</u>
<code>\$LtsAlignmentHorizCenter</code> (1056964611)	Centers all data horizontally.
<code>\$LtsAlignmentJustify</code> (1056964613)	Justifies the alignment of all data with the left and right boundaries of the columns.
<code>\$LtsAlignmentLeft</code> (1056964609)	Aligns all data with the left side of the column.
<code>\$LtsAlignmentRight</code> (1056964610)	Aligns all data with the right side of the column.
<code>\$LtsAlignmentSmart</code> (1056964612)	Aligns data according to the type of data. Numbers are right-aligned, text is left-aligned, and so on.
<code>\$LwpAlignmentTypeAlignRevert</code> (8)	Reverts the alignment of all data to the alignment of provided by the paragraph style.
<code>\$LwpAlignmentTypeJustifyall</code> (5)	Justifies the alignment of all data with the left and right boundaries of the columns.
<code>\$LwpAlignmentTypeNumericleft</code> (6)	Aligns all data to the left-most decimal (or period if text is present).
<code>\$LwpAlignmentTypeNumericright</code> (7)	Aligns all data to the right-most decimal (or period if text is present).

Return value

None.

Usage

Equivalent to choosing Create - Parallel Columns.

Data in parallel columns flows from the top to the bottom of the left column and then into the next column to the right. Once the data reaches the bottom of the right parallel column on a page, it then flows to the top of the left parallel column on the next page.

When created, you can access and manipulate the currently active ParallelColumns object through the ParallelColumns property on WPApplication.

Word Pro: CreateRemark method

{button ,AL('H_VERSION_CLASS',0)} [See list of classes](#)

{button ,AL('H_CREATEREMARK_METHOD_EXSCRIPT',1)} [See example](#)

Allows you to insert an editorial remark in a version of a document.

Syntax

[objectreference].CreateRemark[(Remarks, Time, EditorInitials)]

Parameters

Remarks

Data type is String.

Time

Data type is Long. This value represents the number of seconds that have elapsed since midnight on January 1, 1970.

EditorInitials

Data type is String.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Equivalent to choosing File - Versions, selecting the desired version, and clicking New Remark.

Word Pro: CreateTable method

{button ,AL(`H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL(`H_CREATETABLE_METHOD_EXSCRIPT',1)} [See example](#)

Creates a table at the insertion point in the currently active document. Equivalent to choosing Create - Table.

Syntax

[objectreference].CreateTable([UseDefault,][FrameStyle,][Columns,][Rows]

Parameters

UseDefault

Uses or bypasses the default table style. Data type is Integer. The legal values for this parameter are -1 and 0 but you may use the LotusScript constants True (-1) and False (0). Optional parameter. Default is True, which tells Word Pro to use the default table style.

FrameStyle

A String expression which specifies the name of the table style you want Word Pro to use in creating the new table. Use this parameter when you set the UseDefault parameter to False.

Columns

An Integer value indicating the number of columns you want to have in the new table. Data type is Long. You must provide a value for this parameter if you set UseDefault to False.

Rows

An Integer value indicating the number of rows you want to have in the new table. Data type is Long. You must provide a value for this parameter if you set UseDefault to False.

Return value

None.

Usage

Issued without parameters, this method creates a table based on the default table style.

Word Pro: CreateVersion method

{button ,AL('H_VERSIONMANAGER_CLASS',0)} [See list of classes](#)

{button ,AL('H_CREATEVERSION_METHOD_EXSCRIPT',1)} [See example](#)

Allows you to create a version of a document for editing. Equivalent to File - Versions.

Syntax

[objectreference].CreateVersion(Name)

Parameters

Name

Data type is String.

Return value

Usage

Word Pro: Create method

{button ,AL('H_FOUNDRY_CLASS',0)} [See list of classes](#)

{button ,AL('H_CREATE_METHOD_EXSCRIPT',1)} [See example](#)

Creates the specified type of object in the Foundry object from which you call this method.

Syntax

[objectreference].Create(CreateType [,P2] [,P3])

Parameters

CreateType

Indicates which type of object you want to create. The need for parameters P2 and P3 is determined by the value of this parameter. Most of the values listed below do not require you to include the P2 or P3 parameters.

Choose one of the string values below or its numeric equivalent (indicated in parentheses):

<u>Value</u>	<u>Effect</u>
\$LwpFoundryCreateTypeStyle (2028)	Creates an object from the one of the following classes: CellLayout, CharacterStyle, FrameLayout, PageLayout, ParagraphStyle, TableLayout. When you use this value for CreateType, you must also include values for <u>P2</u> and <u>P3</u> .
\$LwpFoundryCreateTypeLayout (2029)	Creates a Layout object from one of the following classes: FrameLayout, NoteLayout, PageLayout, RubyLayout. When you use this value for CreateType, you must also include a value for <u>P2</u> but no value is needed for P3.
\$LwpFoundryCreateTypeText (2030)	Creates a Text object from the Text class. When you use this value for CreateType, you do not need values for P2 and P3.
\$LwpFoundryCreateTypeSupertable (2031)	Creates a SuperTable object from the SuperTable class. When you use this value for CreateType, you do not need values for P2 and P3.
\$LwpFoundryCreateTypeToc (2032)	Creates a TOCSuperTable object from the TOCSuperTable class. When you use this value for CreateType, you do not need values for P2 and P3.
\$LwpFoundryCreateTypeParallelcolumns (2033)	Creates a ParallelColumns object from the ParallelColumns class. When you use this value for CreateType, you do not need values for P2 and P3.
\$LwpFoundryCreateTypeGraphic (2034)	Creates a Graphic object from the Graphic class. When you use this value for CreateType, you do not need values for P2 and P3.
\$LwpFoundryCreateTypeOle (2035)	Creates an OLEObject object from the OLEObject class. When you use this value for CreateType, you do not need values for P2 and P3.
\$LwpFoundryCreateTypeFootnote (2036)	Creates a Footnote object from the Footnote class. When you use this value for CreateType, you need a value for <u>P2</u> but you do not need a value for P3.
\$LwpFoundryCreateTypeField (2037)	Creates a DocInfoField object from the DocInfoField class. When you use this value for CreateType, you do not need values for P2 and P3.
\$LwpFoundryCreateTypeSilverbullet (2038)	Creates a SilverBullet object from the SilverBullet class. When you use this value for CreateType, you do not need values for P2 and P3.
\$LwpFoundryCreateTypeSection (2039)	Creates a Section object from the Section class. When you use this value for CreateType, you do not need values for P2 and P3.
\$LwpFoundryCreateTypeIndexsecti	Creates an IndexSection object from the IndexSection

on (2040)	class. When you use this value for CreateType, you do not need values for P2 and P3.
\$LwpFoundryCreateTypeBag (2041)	Creates a Bag object from the Bag class. When you use this value for CreateType, you do not need values for P2 and P3.
\$LwpFoundryCreateTypeOutlineseq (2042)	Creates an OutSeqItem object from the OutSeqItem class. When you use this value for CreateType, you do not need values for P2 and P3.
\$LwpFoundryCreateTypeClickhere (2043)	Creates a ClickHere object from the ClickHere class. When you use this value for CreateType, you do not need values for P2 and P3.

P2

Data type for P2 is Variant, which allows P2 to accept values of any data type. There is no default value for P2. Use P2 only when the CreateType parameter has one of the following three values:

\$LwpFoundryCreateTypeStyle (2028)

If CreateType has a value of \$LwpFoundryCreateTypeStyle (or 2028), then the P2 parameter value must be a String which specifies the name of the new style object you are creating. You must also include a value for P3.

\$LwpFoundryCreateTypeLayout (2029)

If CreateType has a value of \$LwpFoundryCreateTypeLayout (or 2029), you must use one of the following values for P2:

<u>Value</u>	<u>Layout Object Created</u>
7	PageLayout
17	NoteLayout
662	FrameLayout
628	RubyLayout

\$LwpFoundryCreateTypeFootnote (2036)

If CreateType has a value of \$LwpFoundryCreateTypeFootnote (or 2036), the value of the P2 parameter must be one of the following strings or its numeric equivalent (indicated in parentheses):

- \$LwpFnTypeAnyposition (289)
- \$LwpFnTypeAtBottomOfPage (290)
- \$LwpFnTypeAtEndOfDoc (291)
- \$LwpFnTypeAtEndOfDocSepDiv (292)
- \$LwpFnTypeAtEndOfDiv (293)
- \$LwpFnTypeAtEndOfDivisionSepDiv (294)
- \$LwpFnTypeAtEndOfDivisionGroup (295)
- \$LwpFnTypeAtEndOfDivGroupSepDiv (296)

P3

An Integer expression which specifies which type of style object you want to create. Use this parameter only when you set the value of the CreateType parameter to \$LwpFoundryCreateTypeStyle (or 2028). You must use one of the integers below:

<u>Value</u>	<u>Style Object Created</u>
7	Page
8	Frame
23	Table
25	Cell
35	Paragraph

39 Character
107 Header
108 Footer
676 DropCap

Return value

A String representing the name of the object created.

Usage

In order to create an object which you can see in your document, you must call this method from the Foundry object, found in the Foundry property on WPAApplication or a Division. DO NOT use the Foundry object stored in WPAApplication.AppFoundry or TextDocument.Foundry.

Word Pro: CutSelection method

{button ,AL('H_WPAPPLICATION_CLASS;H_DOCUMENT_CLASS;H_TEXTDOCUMENT_CLASS',0)} [See list of classes](#)

{button ,AL('H_CUTSELECTION_METHOD_EXSCRIPT',1)} [See example](#)

Deletes the current selection and places a copy in the Clipboard and the Foundry object located in the AppFoundry property on WPAApplication. Equivalent to choosing Edit - Cut.

Syntax

[objectreference].CutSelection()

Parameters

None.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Word Pro: DarkMode method

{button ,AL('H_APPLICATIONWINDOW_CLASS;H_DOCWINDOW_CLASS',0)} [See list of classes](#)

{button ,AL('H_DARKMODE_METHOD_EXSCRIPT',1)} [See example](#)

Used to turn off updating of the Word Pro screen during execution of a script.

Syntax

[Objectreference].ApplicationWindow.DarkMode (Integer) Integer

Parameters

Integer

Data type is Integer. The legal values for this parameter are -1 or 0 but you may use the LotusScript constants of True (-1) and False (0).

Return value

The return values for this method will always be -1 or 0. When testing the return value, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

When this property is turned on, the screen display in the current MDI window is frozen. The screen is not updated until DarkMode is turned off. Make sure that you turn DarkMode off before the script finishes, or you will never see the screen repaint.

Word Pro: DataObjectGetDataHere method

{button ,AL(^H_GRAPHIC_CLASS',0)} [See list of classes](#)

Syntax

[objectreference].DataObjectGetDataHere(pFormatEtc,pStgMedium)

Parameters

pFormatEtc

Data type is Long.

pStgMedium

Data type is Long.

Return value

Integer.

Usage

Word Pro: DataObjectGetData method

{button ,AL(^H_GRAPHIC_CLASS',0)} [See list of classes](#)

Syntax

[objectreference].DataObjectGetData(pFormatEtc,pStgMedium)

Parameters

pFormatEtc

Data type is Long.

pStgMedium

Data type is Long.

Return value

Integer.

Usage

Word Pro: DbUnderline method

{button ,AL(`H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL(`H_DBLUNDERLINE_METHOD_EXSCRIPT',1)} [See example](#)

Sets the double underline attribute for selected text, or all following text if no text is selected. Acts as a toggle, turning the attribute off if it is on and on if it is off. Equivalent to choosing Text - Attributes - Other and then choosing "Db Underline" from the Attributes box.

Syntax

[objectreference].DbUnderline()

Parameters

None.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Word Pro: DeleteBag method

{button ,AL('H_BAG_CLASS',0)} [See list of classes](#)

{button ,AL('H_DELETEBAG_METHOD_EXSCRIPT',1)} [See example](#)

Syntax

[objectreference].DeleteBag()

Parameters

None.

Return value

Integer.

Usage

Word Pro: DeleteButton method

{button ,AL('H_STATUSBARBUTTON_CLASS',0)} [See list of classes](#)

{button ,AL('H_DELETEBUTTON_METHOD_EXSCRIPT',1)} [See example](#)

Deletes a button from the status bar.

Syntax

[objectreference].DeleteButton()

Parameters

None.

Return value

Integer. Always returns True.

Usage

Use when you want to delete a selected button from the status bar. Once you delete a button, you can no longer access it.

Word Pro: DeleteChars method

{button ,AL(^H_CLICKHERE_CLASS;H_TEXTMARKER_CLASS;H_TEXT_CLASS',0)} [See list of classes](#)

{button ,AL(^H_DELETECHARS_METHOD_EXSCRIPT',1)} [See example](#)

Deletes the specified number of characters following the insertion point.

Syntax

[objectreference].DeleteChars(Count)

Parameters*Count*

An Integer expression which specifies the number of characters you want to delete.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Equivalent to the forward delete (DEL or Delete) key on most keyboards.

Word Pro: DeleteClickHere method

{button ,AL(`H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL(`H_DELETECLICKHERE_METHOD_EXSCRIPT',1)} [See example](#)

Deletes the Click Here Block at the insertion point. Any contents of the Click Here Block are also deleted.

Syntax

[objectreference].DeleteClickHere()

Parameters

None

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Equivalent to right-clicking a ClickHere Block and choosing Delete Click Here Block.

Word Pro: DeleteContainer method

```
{button ,AL('H_BASECONTAINER_CLASS;H_CELLCONTAINER_CLASS;H_FRAMECONTAINER_CLASS;H_NOTE  
CONTAINER_CLASS;H_RUBYCONTAINER_CLASS;H_DROPCAPCONTAINER_CLASS;H_PAGECONTAINER_  
CLASS;H_SUBPAGECONTAINER_CLASS;H_SUPERPAGECONTAINER_CLASS;H_ROWCONTAINER_CLASS;  
H_SUPERTABLECONTAINER_CLASS;H_TABLECONTAINER_CLASS;H_PARALLELCOLSCONTAINER_CLASS  
;H_TABLEONLYCONT_CLASS',0)} See list of classes
```

```
{button ,AL('H_DELETECONTAINER_METHOD_EXSCRIPT',1)} See example
```

Deletes a specific container object.

Syntax

```
[objectreference].DeleteContainer()
```

Parameters

Return value

The return value for this method will always be -1 or 0. When testing the return value, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Word Pro: DeleteContents method

```
{button ,AL('H_LAYOUT_CLASS;H_CELLGROUPLAYOUT_CLASS;H_CELLLAYOUT_CLASS;H_CONNECTEDLAY  
OUT_CLASS;H_COLUMNGROUPLAYOUT_CLASS;H_FOOTERLAYOUT_CLASS;H_FRAMELAYOUT_CLASS;H  
_FRAMEGROUPLAYOUT_CLASS;H_DROP_CAPLAYOUT_CLASS;H_GROUPLAYOUT_CLASS;H_HEADERLAY  
OUT_CLASS;H_NOTELAYOUT_CLASS;H_PAGELAYOUT_CLASS;H_ROWGROUPLAYOUT_CLASS;H_ROWLA  
YOUT_CLASS;H_RUBYLAYOUT_CLASS;H_SUPERTABLEGROUPLAYOUT_CLASS;H_SUPERTABLELAYOUT  
_CLASS;H_TABLELAYOUT_CLASS;H_ENDNOTELAYOUT_CLASS;H_FOOTNOTELAYOUT_CLASS;H_TABLE  
HEADINGLAYOUT_CLASS;H_TOCSUPERTABLELAYOUT_CLASS;H_MARKER_CLASS;H_POWERFIELD_CLA  
SS;H_CLICKHERE_CLASS;H_TEXTMARKER_CLASS;H_RUBYMARKER_CLASS;H_TABLEMARKER_CLASS',  
0)} See list of classes
```

```
{button ,AL('H_DELETECONTENTS_METHOD_EXSCRIPT',1)} See example
```

Deletes the contents from an object.

Syntax

```
[objectreference].DeleteContents()
```

Parameters

Return value

The return value for this method will always be -1 or 0. When testing the return value, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Word Pro: DeleteContent method

{button ,AL('H_GRAPHICOLEBJECT_CLASS;H_GRAPHIC_CLASS;H_OLEOBJECT_CLASS',0)} [See list of classes](#)

{button ,AL('H_DELETECONTENT_METHOD_EXSCRIPT',1)} [See example](#)

Deletes the content in a GraphicOleObject.

Syntax

[objectreference].DeleteContent()

Parameters

None.

Return value

Integer.

Usage

Word Pro: DeleteDivision method

{button ,AL('H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL('H_DELETEDIVISION_METHOD_EXSCRIPT',1)} [See example](#)

Deletes the currently active division object, its contents, and any child divisions. Equivalent to clicking the right mouse button on the division divider tab and choosing Delete Division.

Syntax

[objectreference].DeleteDivision([DivisionName])

Parameters

DivisionName

An optional String parameter representing the internal name of the division as it appears in the Name property of the division object.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Every document must have at least one division, so you cannot delete the last remaining division in a document.

Word Pro: DeleteField method

{button ,AL('H_DOCINFOFIELDMANAGER_CLASS',0)} [See list of classes](#)

{button ,AL('H_DELETEFIELD_METHOD_EXSCRIPT',1)} [See example](#)

Removes a document field from a document.

Syntax

[objectreference].DeleteField(FieldName)

Parameters

FieldName

The name of the field you want to delete. Data type is String.

Return value

Returns True if the specified document field is deleted. Returns False if the specified document field is not deleted or if the document field does not exist.

Usage

Equivalent to choosing File - Document properties, choosing Document, clicking the Fields tab, and clicking Delete.

Word Pro: DeleteItemByPosition method

{button ,AL('H_MENUITEM_CLASS',0)} [See list of classes](#)

Allows you to delete a MenuItem object by specifying its position within the parent menu. The parent menu is the menu object from which you call this method.

Syntax

[objectreference].DeleteItemByPosition(Position)

Parameters

Position

An Integer value which specifies the position of the menu item to be deleted. The topmost menu item corresponds to a position value of 0.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

When this method is called from the MenuItem object stored in the LwpMenuBar property, the Position parameter refers to the position of the menu on the Word Pro menu bar. Therefore, if you call this method and specify a value of 1 in the Position parameter, the Edit menu and all child menus are deleted.

Word Pro: Deleteltem method

{button ,AL('H_MENUITEM_CLASS',0)} [See list of classes](#)

{button ,AL('H_DELETEITEM_METHOD_EXSCRIPT',1)} [See example](#)

Deletes a menu item from a MenuItem object.

Syntax

[objectreference].Deleteltem(MenuText)

Parameters

MenuText

The String name of the menu item you want to delete from a menu.

Return value

String

Usage

In order to delete a child menu item, call this method from its parent and specify the caption of the menu item that is to be deleted.

If you delete an existing Word Pro menu item, Word Pro regenerates the menu item when you terminate the current Word Pro session and launch Word Pro again. To permanently remove a Word Pro menu item, you must create a startup script to remove the item each time you start Word Pro.

Word Pro: DeleteKey method

{button ,AL('H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL('H_DELETEKEY_METHOD_EXSCRIPT',1)} [See example](#)

Deletes the current selection. If nothing is selected, this method deletes the object in front of the insertion point. Equivalent to pressing the Delete key one time.

Syntax

[objectreference].DeleteKey()

Parameters

None.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Word Pro: DeleteLayout method

```
{button ,AL(`H_LAYOUT_CLASS;H_CELLGROUPLAYOUT_CLASS;H_CELLLAYOUT_CLASS;H_CONNECTEDLAY  
OUT_CLASS;H_COLUMNGROUPLAYOUT_CLASS;H_FOOTERLAYOUT_CLASS;H_FRAMELAYOUT_CLASS;H  
_FRAMEGROUPLAYOUT_CLASS;H_DROP_CAPLAYOUT_CLASS;H_GROUPLAYOUT_CLASS;H_HEADERLAY  
OUT_CLASS;H_NOTELAYOUT_CLASS;H_PAGELAYOUT_CLASS;H_ROWGROUPLAYOUT_CLASS;H_ROWLA  
YOUT_CLASS;H_RUBYLAYOUT_CLASS;H_SUPERTABLEGROUPLAYOUT_CLASS;H_SUPERTABLELAYOUT  
_CLASS;H_TABLELAYOUT_CLASS;H_ENDNOTELAYOUT_CLASS;H_FOOTNOTELAYOUT_CLASS;H_TABLE  
HEADINGLAYOUT_CLASS;H_TOCSUPERTABLELAYOUT_CLASS',0)} See list of classes
```

```
{button ,AL(`H_DELETELAYOUT_METHOD_EXSCRIPT',1)} See example
```

Deletes the layout object from which this method is called. This method also deletes the contents of the layout object.

Syntax

[objectreference].DeleteLayout()

Parameters

Return value

The return value for this method will always be -1 or 0. When testing the return value, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Word Pro does not delete the contents of a layout object if another object is using those contents. For example, if cell A1 is referenced in a formula, you can use the DeleteLayout method to delete the celllayout object of cell A1. However, the content of that celllayout object is still included in the calculation of the formula. The result of the formula does not change until the content of cell A1 is also deleted.

Word Pro: DeleteMacroAccelerator method

{button ,AL('H_ACCELERATORS_CLASS',0)} [See list of classes](#)

Deletes the macro accelerator key.

Syntax

[objectreference].DeleteMacroAccelerator(MacroName)

Parameters

MacroName

Data type is String.

Return value**Usage**

Word Pro: DeleteMarker method

{button ,AL('H_MARKER_CLASS;H_POWERFIELD_CLASS;H_CLICKHERE_CLASS;H_TEXTMARKER_CLASS;H_RUBYSMARKER_CLASS;H_TABLEMARKER_CLASS',0)} [See list of classes](#)

{button ,AL('H_DELETEMARKER_METHOD_EXSCRIPT',1)} [See example](#)

Deletes a marker from a document.

Syntax

[objectreference].DeleteMarker()

Parameters**Return value**

The return value for this method will always be -1 or 0. When testing the return value, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Word Pro: DeleteParallelColumns method

{button ,AL(^H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL(^H_DELETEPARALLELCOLUMNS_METHOD_EXSCRIPT',1)} [See example](#)

Deletes a parallel column from a document. Equivalent to choosing Columns - Delete - All Columns.

Syntax

[objectreference].DeleteParallelColumns()

Parameters

None.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Word Pro: DeleteSection method

{button ,AL(^H_WPAPPLICATION_CLASS;H_SECTION_CLASS;H_INDEXSECTION_CLASS',0)} [See list of classes](#)

{button ,AL(^H_DELETESECTION_METHOD_EXSCRIPT',1)} [See example](#)

Deletes a section object from a division. Equivalent to clicking the right mouse button on the section divider tab and choosing Delete Section.

Syntax

[objectreference].DeleteSection()

Parameters

Return value

Usage

Word Pro: DeleteSmartCorrect method

{button ,AL('H_SMARTCORRECT_CLASS',0)} [See list of classes](#)

{button ,AL('H_DELETESMARTCORRECT_METHOD_EXSCRIPT',1)} [See example](#)

Removes a SmartCorrect entry from the specified SmartCorrect object.

Syntax

[objectreference].DeleteSmartCorrect(Entry)

Parameters

Entry

This String parameter represents the name of the entry to be deleted.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Word Pro: DeleteTable method

{button ,AL(^H_WPAPPLICATION_CLASS;H_BASETABLE_CLASS;H_FOOTNOTETABLE_CLASS;H_PARALLELCOLUMNS_CLASS;H_GLOSSARY_CLASS;H_TABLEHEADING_CLASS;H_TABLE_CLASS',0)} [See list of classes](#)

{button ,AL(^H_DELETETABLE_METHOD_EXSCRIPT',1)} [See example](#)

[WPAApplication]

Deletes the table object which has the focus. If no table object is in the focus, nothing is deleted.

[BaseTable]

Deletes an entire table or specified components of a table.

Syntax

[objectreference].DeleteTable()

[objectreference].DeleteTable(TableDelType, Start, Count)

Parameters

[WPAApplication]

None

[BaseTable]

TableDelType

Indicates which type of item is to be deleted. The value of this Variant parameter must be one of the string constants below or its numeric equivalent.

Value	Effect
\$LwpTableDelTypeRow (1867)	Deletes any rows that are included in the current selection, unless you specify values for the Start and Count parameters.
\$LwpTableDelTypeColumn (1868)	Deletes any columns that are included in the current selection, unless you specify values for the Start and Count parameters.
\$LwpTableDelTypeWholetable (1869)	Deletes the entire table.

Start

An Integer parameter which indicates the beginning column or row ID to be deleted. The ID is zero based, which means that the first row and column within a table have an ID of 0. The Start parameter must be used in combination with the Count parameter. This parameter has no effect when the TableDelType parameter value is \$LwpTableDelTypeWholetable.

Count

An Integer parameter which indicates the number of rows or columns to be deleted. The Count parameter must be used in conjunction with the Start parameter. This parameter has no effect when the TableDelType parameter value is \$LwpTableDelTypeWholetable.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

[WPAApplication]

Equivalent to choosing Table - Delete - Entire Table.

[BaseTable]

Equivalent to choosing Table - Delete and choosing the object to be deleted.

Word Pro: DeleteTab method

{button ,AL('H_TABRACK_CLASS',0)} [See list of classes](#)

{button ,AL('H_DELETETAB_METHOD_EXSCRIPT',1)} [See example](#)

Deletes a tab setting from the horizontal ruler. Equivalent to choosing View - Show/Hide - Ruler to display the horizontal ruler, then dragging a tab off the ruler and releasing the mouse button.

Syntax

[objectreference].DeleteTab(Index)

Parameters

Index

This Integer parameter represents the index number of the tab to delete. The tab index values are zero-based, which means the first tab on the ruler has an index value of 0.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Word Pro: DeleteVersion method

{button ,AL('H_VERSIONMANAGER_CLASS',0)} [See list of classes](#)

{button ,AL('H_DELETEVERSION_METHOD_EXSCRIPT',1)} [See example](#)

Deletes a version of a document.

Syntax

[objectreference].DeleteVersion(VersionID)

Parameters

VersionID

Data type is Long. Use the DocVersionID property for the version you want to delete as this parameter.

Return value**Usage**

Equivalent to choosing File - Versions and clicking Delete Version on the version you want to delete.

Word Pro: DemoteOutlineLevel method

{button ,AL(^H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL(^H_DEMOTEOUTLINELEVEL_METHOD_EXSCRIPT',1)} [See example](#)

Demotes the current paragraph to the next lower outline level. Equivalent to choosing Text - Outline - Demote.

Syntax

[objectreference].DemoteOutlineLevel()

Parameters

None

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Affects the paragraph which has the focus or the paragraphs which are selected when you call the method from WPAplication. If you want to demote a paragraph that does not have the focus, use the Demote method.

Word Pro: Demote method

{button ,AL(^H_CLICKHERE_CLASS;H_TEXTMARKER_CLASS;H_TEXT_CLASS',0)} [See list of classes](#)

Demotes a paragraph to the next lower outline level. Equivalent to choosing Text - Outline - Demote.

Syntax

[objectreference].Demote()

Parameters

None.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Assigns the next lower paragraph style in the outline style sequence. It affects the paragraph or paragraphs in the object from which you call the method. You can call this method from a Text, TextMarker or ClickHere object. If you would like to demote the paragraph that has the focus, use the DemoteOutlineLevel method.

Word Pro: Deselect method

```
{button ,AL('H_WPAPPLICATION_CLASS;H_CLICKHERE_CLASS;H_TEXTMARKER_CLASS;H_TEXT_CLASS',0)}
```

[See list of classes](#)

```
{button ,AL('H_DESELECT_METHOD_EXSCRIPT',1)} See example
```

Deselects the current selection in a document. Equivalent to deselecting by clicking outside the current selection or by pressing Esc.

Syntax

```
[objectreference].Deselect()
```

Parameters

None.

Return value

The return value for this method will always be -1 or 0. When testing the return value, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Word Pro: DestroyDocWindow method

{button ,AL('H_DOCWINDOW_CLASS',0)} [See list of classes](#)

{button ,AL('H_DESTROYDOCWINDOW_METHOD_EXSCRIPT',1)} [See example](#)

Closes the specified DocWindow object without prompting the user to save the document.

Syntax

[objectreference].DestroyDocWindow()

Parameters

None.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Word Pro: Destroy method

{button ,AL('H_TEXTDOCUMENT_CLASS',0)} [See list of classes](#)

Syntax

[objectreference].Destroy()

Parameters

Return value

Usage

Word Pro: DisconnectCells method

{button ,AL(^H_WPAPPLICATION_CLASS;H_BASSETABLE_CLASS;H_FOOTNOTETABLE_CLASS;H_PARALLELCOLUMNS_CLASS;H_GLOSSARY_CLASS;H_TABLEHEADING_CLASS;H_TABLE_CLASS';0)} [See list of classes](#)

{button ,AL(^H_DISCONNECTCELLS_METHOD_EXSCRIPT',1)} [See example](#)

Disconnects table cells which have been connected using the Table menu commands, the ConnectCells method, or the ConnectRows method.

Syntax

[objectreference].DisconnectCells()

Parameters

None.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Equivalent to choosing Table - Disconnect Cell.

Word Pro: Disconnect method

```
{button ,AL(^H_WPAPPLICATION_CLASS;H_BASECONTAINER_CLASS;H_CELLCONTAINER_CLASS;H_FRAME  
CONTAINER_CLASS;H_NOTECONTAINER_CLASS;H_RUBYCONTAINER_CLASS;H_DROPDOWNCONTAINER_  
CLASS;H_PAGECONTAINER_CLASS;H_SUBPAGECONTAINER_CLASS;H_SUPERPAGECONTAINER_CLASS  
;H_ROWCONTAINER_CLASS;H_SUPERTABLECONTAINER_CLASS;H_TABLECONTAINER_CLASS;H_PARAL  
LELCOLSCONTAINER_CLASS;H_TABLEONLYCONT_CLASS',0)} See list of classes
```

```
{button ,AL(^H_DISCONNECT_METHOD_EXSCRIPT',1)} See example
```

Disconnects grouped container objects.

Syntax

```
[objectreference].Disconnect()
```

Parameters**Return value**

The return value for this method will always be -1 or 0. When testing the return value, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Word Pro: DoesMarkerNameMatch method

```
{button ,AL('H_LAYOUT_CLASS;H_CELLGROUPLAYOUT_CLASS;H_CELLLAYOUT_CLASS;H_CONNECTEDLAY  
OUT_CLASS;H_COLUMNGROUPLAYOUT_CLASS;H_FOOTERLAYOUT_CLASS;H_FRAMELAYOUT_CLASS;H  
_FRAMEGROUPLAYOUT_CLASS;H_DROPACPLAYOUT_CLASS;H_GROUPLAYOUT_CLASS;H_HEADERLAY  
OUT_CLASS;H_NOTELAYOUT_CLASS;H_PAGELAYOUT_CLASS;H_ROWGROUPLAYOUT_CLASS;H_ROWLA  
YOUT_CLASS;H_RUBYLAYOUT_CLASS;H_SUPERTABLEGROUPLAYOUT_CLASS;H_SUPERTABLELAYOUT  
_CLASS;H_TABLELAYOUT_CLASS;H_ENDNOTELAYOUT_CLASS;H_FOOTNOTELAYOUT_CLASS;H_TABLE  
HEADINGLAYOUT_CLASS;H_TOCSUPERTABLELAYOUT_CLASS;H_BASETABLE_CLASS;H_FOOTNOTETAB  
LE_CLASS;H_PARALLELCOLUMNS_CLASS;H_GLOSSARY_CLASS;H_TABLEHEADING_CLASS;H_TABLE_C  
LASS',0)} See list of classes
```

```
{button ,AL('H_DOESMARKERNAMEMATCH_METHOD_EXSCRIPT',1)} See example
```

Determines whether an object is marked by a specific marker.

Syntax

```
[objectreference].DoesMarkerNameMatch(MarkerName)
```

Parameters

MarkerName

The DoesMarkerNameMatch method checks the specified object for a marker whose name matches the MarkerName parameter value. This parameter is a String expression.

Return value

The return value for this method will always be -1 or 0. When testing the return value, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

This method operates only on default markers (\$LwpMarkerTypeDefault).

Use this method to check for markers on layout objects, such as frames. If the insertion point is within a stream of text or if text is selected, use the IsMarkerEqualToSelection method instead of DoesMarkerNameMatch.

Word Pro: DoneWithRightContextMenu method

{button ,AL(^H_GRAPHICOLEBJECT_CLASS;H_GRAPHIC_CLASS;H_OLEOBJECT_CLASS',0)} [See list of classes](#)

Indicates that the right mouse menu was used to create a graphic.

Syntax

[objectreference].DoneWithRightContextMenu(MenuHandle)

Parameters

MenuHandle

Data type is Long.

Return value

Integer.

Usage

Word Pro: EditHyperlink method

{button ,AL(^H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

Changes the settings of the Hyperlink at the insertion point. Specify the new settings using the parameters described below.

Syntax

[objectreference].EditHyperlink(URL, Anchor, AltTag, MarkerName[, LinkedText])

Parameters

URL

A string value which specifies the destination of the Hyperlink. This string is usually a URL, but could also be any other valid HTML link code such as and IP address, a "mailto:" command, or an FTP address.

Anchor

ALTTag

MarkerName

LinkedText

A string value which specifies the text you want to appear in the document. This is the "clickable" text seen in a browser. This parameter is optional. If you do not provide a value for this parameter, Word Pro uses the existing linked text.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

For this method to work, the insertion point must be in a Hyperlink, or some Hyperlink text must be selected.

Word Pro: EditLinkInfo method

{button ,AL('H_DDELINK_CLASS',0)} [See list of classes](#)

Allows you to update whatever you are linked to. Equivalent to choosing Edit - Manage Links.

Syntax

[objectreference].EditLinkInfo(LinkInfo)

Parameters

LinkInfo

Data type is String.

Return value

The return value for this method will always be -1 or 0. When testing the return value, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

When you choose Edit - Manage Links, you can change DdeLink information by editing it, changing the name of the link, opening the link source, or breaking the link.

Word Pro: Embed method

```
{button ,AL('H_WPAPPLICATION_CLASS;H_CLICKHERE_CLASS;H_TEXTMARKER_CLASS;H_TEXT_CLASS',0)}
```

[See list of classes](#)

Inserts the named object into the layout of the object from which you call this method: PageLayout, SuperTableLayout, FrameLayout.

Syntax

When called from the WPAApplication object:

```
[objectreference].Embed(ContentName, [Advance])
```

When called from a Text, TextMarker, or ClickHere object:

```
[objectreference].Embed(Objectname, [Advance])
```

Parameters

ContentName

Data type String.

Objectname

Data type is String.

Advance

Indicates whether or not the insertion point is left at the beginning of the embedded object or is advanced to the end of the object. Data type is Boolean. Optional parameter. Default is True.

Return value

The return value for this method will always be -1 or 0. When testing the return value, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Word Pro: EndChange method

{button ,AL(^H_WPAPPLICATION_CLASS;H_DIVISION_CLASS;H_TEXTDOCUMENT_CLASS;H_DOCWINDOW_CLASSES',0)} [See list of classes](#)

{button ,AL(^H_ENDCHANGE_METHOD_EXSCRIPT',1)} [See example](#)

Ends any changes made to the Word Pro application, a document window, a division, or a text document. This method is defined in the following classes: WPAApplication, Division, TextDocument, DocWindow.

Syntax

[objectreference].WPAApplication.EndChange([MarkChanges])

[objectreference].Division.EndChange([MarkChanges])

[objectreference].Text.DocumentEndChange([MarkChanges])

[objectreference].DocWindow.EndChange()

Parameters

MarkChanges

Data type is Boolean. Optional parameter. Default is False.

Return value

Boolean.

Usage

Word Pro: EndCustomLines method

{button ,AL(^H_TABLELINE_CLASS',0)} [See list of classes](#)

This method is used by Word Pro when recording a script, in order to reflect the end of a customized table line style selection.

Syntax

[objectreference].EndCustomLines()

Parameters

Return value

Always returns 0.

Usage

Word Pro: Ending method

```
{button ,AL('H_BASECONTAINER_CLASS;H_CELLCONTAINER_CLASS;H_FRAMECONTAINER_CLASS;H_NOTE  
CONTAINER_CLASS;H_RUBYCONTAINER_CLASS;H_DROPDOWNCONTAINER_CLASS;H_PAGECONTAINER_  
CLASS;H_SUBPAGECONTAINER_CLASS;H_SUPERPAGECONTAINER_CLASS;H_ROWCONTAINER_CLASS;  
H_SUPERTABLECONTAINER_CLASS;H_TABLECONTAINER_CLASS;H_PARALLELCOLSCONTAINER_CLASS  
;H_TABLEONLYCONT_CLASS',0)} See list of classes
```

```
{button ,AL('H_ENDING_METHOD_EXSCRIPT',1)} See example
```

This method moves the insertion point from its current position to the end of the document.

Syntax

[objectreference].Ending (ContainerEndEndOfDocument)

Parameters

ContainerEnd

The value of this Boolean parameter must be the String constant, \$LwpContainerEndEndOfDocument, or its numeric equivalent (160).

Return value

The return value for this method will always be -1 or 0. When testing the return value, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Word Pro: EndPrinting method

{button ,AL('H_PRINTMANAGER_CLASS',0)} [See list of classes](#)

Syntax

[objectreference].EndPrinting()

Parameters

None.

Return value

Integer.

Usage

Word Pro: EnumerateChartLinks method

{button ,AL(^H_GRAPHIC_CLASS',0)} [See list of classes](#)

Syntax

[objectreference].EnumerateChartLinks(LinkCookie)

Parameters

LinkCookie

Data type is Long.

Return value

Long

Usage

Word Pro: EnumerateTerm method

{button ,AL('H_GLOSSARY_CLASS',0)} [See list of classes](#)

{button ,AL('H_ENUMERATETERM_METHOD_EXSCRIPT',1)} [See example](#)

Allows you to retrieve a term by providing its index number.

Syntax

[objectreference].EnumerateTerm(KeyNumber)

Parameters

KeyNumber

The number associated with each entry in a glossary. Word Pro uses this number to enumerate the entries in the glossary.

Return value

A String value that represents the term which corresponds to the number provided in the KeyNumber parameter.

Usage

Use this method to retrieve a term by specifying its number within the glossary. The first term in the glossary has a KeyNumber value of 1.

Word Pro: EnvelopeBarCode method

{button ,AL(^H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL(^H_ENVELOPEBARCODE_METHOD_EXSCRIPT',1)} [See example](#)

Inserts a bar code into an envelope.

Syntax

[objectreference].EnvelopeBarCode()

Parameters

None.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Insert the name and address into the Envelope Send frame before calling this method.

Word Pro: EnvelopePrint method

{button ,AL(^H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

Syntax

[objectreference].EnvelopePrint()

Parameters

Return value

Usage

Word Pro: ExchangeItem method

{button ,AL('H_MENUITEM_CLASS',0)} [See list of classes](#)

{button ,AL('H_EXCHANGEITEM_METHOD_EXSCRIPT',1)} [See example](#)

Exchanges the position of two menu items.

Syntax

[objectreference].ExchangeItem(WithItem)

Parameters

WithItem

Specifies the menu from which you are moving the item.

Return value

Usage

Allows you to switch the position of any two menu items. For example, you can exchange the positions of the Open and Close menu items in the File menu.

Call this method from the menu item to which you are switching. In the above example, you would call the ExchangeItem method from the Open menu item, using the Close menu item as the WithItem parameter value.

Word Pro: ExpandOutline method

{button ,AL('H_CLICKHERE_CLASS;H_TEXTMARKER_CLASS;H_TEXT_CLASS',0)} [See list of classes](#)

Expands the highest level contracted heading(s) which are subordinate to the paragraph for which you are calling the method. For example, when you call this method for a Level 1 heading, it expands the highest level contracted heading(s) which are subordinate to that Level 1 heading.

Syntax

[objectreference].ExpandOutline([All])

Parameters

All

Allows you to expand all the subordinate headings under the heading from which you call this method. Data type is Integer but the legal values for this parameter are -1 and 0. You may use the LotusScript constants True (-1) and False (0). A value of True causes all subordinate headings to be expanded, regardless of their level. Optional parameter. Default is True.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Word Pro: Expand method

{button ,AL('H_SECTIONTABS_CLASS',0)} [See list of classes](#)

{button ,AL('H_EXPAND_METHOD_EXSCRIPT',1)} [See example](#)

Displays divider tabs that are the children of a parent division tab in a document.

Syntax

[objectreference].Expand()

Parameters

None.

Return value

The return value for this method will always be -1 or 0. When testing the return value, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Equivalent to clicking the plus sign on the parent division tab to show all the children divider tabs.

Word Pro: GetAspectMetafilePict method

{button ,AL(^H_OLEOBJECT_CLASS',0)} [See list of classes](#)

Note This method is not supported for the class OLEObject within OS/2.

Syntax

[objectreference].GetAspectMetafilePict(DisplayAsIcon)

Parameters

DisplayAsIcon

Data type is Integer.

Return value

Long.

Usage

Word Pro: GetListName method

{button ,AL('H_SMARTFILL_CLASS',0)} [See list of classes](#)

Syntax

[objectreference].GetListName(ListNum)

Parameters

ListNum

Integer.

Return value

String.

Usage

Word Pro: GetParagraphNumberString method

{button ,AL('H_MARKER_CLASS;H_POWERFIELD_CLASS;H_CLICKHERE_CLASS;H_TEXTMARKER_CLASS;H_RUBYMARKER_CLASS;H_TABLEMARKER_CLASS',0)} [See list of classes](#)

{button ,AL('H_GETPARAGRAPHNUMBERSTRING_METHOD_EXSCRIPT',1)} [See example](#)

Syntax

[objectreference].GetParagraphNumberString()

Parameters

None.

Return value

String.

Usage

Word Pro: GetParagraphNumber method

{button ,AL(^H_MARKER_CLASS;H_POWERFIELD_CLASS;H_CLICKHERE_CLASS;H_TEXTMARKER_CLASS;H_RUBYMARKER_CLASS;H_TABLEMARKER_CLASS',0)} [See list of classes](#)

{button ,AL(^H_GETPARAGRAPHNUMBER_METHOD_EXSCRIPT',1)} [See example](#)

Syntax

[objectreference].GetParagraphNumber(Position)

Parameters

Position

Data type is Integer.

Return value

Integer.

Usage

Word Pro: GetString method

{button ,AL('H_SMARTFILL_CLASS',0)} [See list of classes](#)

Syntax

[objectreference].GetString(ListNum, StringNum)

Parameters

ListNum

Data type is Integer.

StringNum

Data type is Integer.

Return value

String.

Usage

Word Pro: GoToClickHere method

{button ,AL(`H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL(`H_GOTOCCLICKHERE_METHOD_EXSCRIPT',1)} [See example](#)

Syntax

[objectreference].GoToClickHere(Name)

Parameters

Name

Data type is String.

Return value

Integer.

Usage

Word Pro: ImportWatermarkGraphic method

```
{button ,AL('H_LAYOUT_CLASS;H_CELLGROUPLAYOUT_CLASS;H_CELLLAYOUT_CLASS;H_CONNECTEDLAY  
OUT_CLASS;H_COLUMNGROUPLAYOUT_CLASS;H_FOOTERLAYOUT_CLASS;H_FRAMELAYOUT_CLASS;H  
_FRAMEGROUPLAYOUT_CLASS;H_DROP_CAPLAYOUT_CLASS;H_GROUPLAYOUT_CLASS;H_HEADERLAY  
OUT_CLASS;H_NOTELAYOUT_CLASS;H_PAGELAYOUT_CLASS;H_ROWGROUPLAYOUT_CLASS;H_ROWLA  
YOUT_CLASS;H_RUBYLAYOUT_CLASS;H_SUPERTABLEGROUPLAYOUT_CLASS;H_SUPERTABLELAYOUT  
_CLASS;H_TABLELAYOUT_CLASS;H_ENDNOTELAYOUT_CLASS;H_FOOTNOTELAYOUT_CLASS;H_TABLE  
HEADINGLAYOUT_CLASS;H_TOCSUPERTABLELAYOUT_CLASS',0)} See list of classes
```

```
{button ,AL('H_IMPORTWATERMARKGRAPHIC_METHOD_EXSCRIPT',1)} See example
```

Imports a graphic into the watermark layer of a layout object.

Syntax

```
[objectreference].ImportWatermarkGraphic(FilePath, FileFormat, Link, ScratchOutFrame, [FrameStyle],  
[ShowPipesWhenDone])
```

Parameters

FilePath

A String expression, which specifies the directory path and name of the file that is the source of the imported graphic.

FileFormat

A String expression, which specifies the file format for the graphic you are importing. The string expression for each file format is unique and registered with Microsoft Windows 95. The values listed in the table below were valid at the time of publication.

<u>If you are importing this type of graphic...</u>	<u>Use this value for the FileFormat parameter...</u>
---	---

Bitmap	.bmp
CGM	.cgm
Corel Draw	.cdr
Encapsulated Postscript	.eps
Freelance Graphics	.drw
GIF	.gif
HPGL	.plt
JPEG	.jpg
Kodak Photo CD	.pcd
Lotus PIC	.pic
PCX	.pcx
TIFF	.tif
Windows Metafile	.wmf
Word Perfect Graphic	.wpg
Word Pro Draw	.sdw
Word Pro Equation	.tex

Link

An Integer value of -1 or 0, indicating whether the imported graphic receives updates from the original (-1), or remains independent of the original (0). You can use the LotusScript constants of True (-1) and False (0) as the value for this parameter.

ScratchOutFrame

An Integer value, indicating whether you want to draw the new graphic frame by hand or let Word Pro draw the frame based on a frame style. If you want to draw the frame yourself, use a True value (-1) for this parameter. If you want Word Pro to draw the frame based on an existing style, use a False value (0).

FrameStyle

A String expression, specifying the frame style you want to use for the imported graphic's frame. Optional parameter. If the imported graphic is an equation and you do not specify a frame style, Word Pro uses the default equation frame style. All other imported graphics are placed in the default GraphicOle frame style, unless you specify another frame style using this parameter.

ShowPipesWhenDone

An Integer value, which determines what object has the focus after the method is called. Applies only to graphics imported into watermark layers of frame layout objects. If you want the frame to be selected after the method is called, use a True value in this parameter. If you want the content of the frame to have the focus, use a False value. Optional parameter. Default is True.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

This method must be called from the layer property of the layout object in which you want the watermark to appear. For example, you might use the following lines of code to import a watermark graphic into the watermark layer of a page layout:

```
.Page.Layout.Layer.ImportWatermarkGraphic("c:\wm.gif", ".gif", 0, 0)
```

Use the LayerName property of the layout object to determine if a watermark layer already exists. If it does not, use the CreateLayer method of the layout object to create it.

{button ,AL(^H_LAYERNAME_PROPERTY_MEMDEF;H_WATERMARKSPATH_PROPERTY_MEMDEF;H_CREATE_LAYER_METHOD_MEMDEF',0)} [See related topics](#)

Word Pro: InternetExtraFile method

{button ,AL('H_APPLICATIONWINDOW_CLASS',0)} [See list of classes](#)

Syntax

[objectreference].InternetExtraFile(Filename, DeleteFile)

Parameters

Filename

Data type is String.

DeleteFile

Data Type is Integer.

Return value

Integer.

Usage

Word Pro: IsCaseExact method

{button ,AL(^H_SMARTFILL_CLASS',0)} [See list of classes](#)

Syntax

[objectreference].IsCaseExact(ListNum)

Parameters

ListNum

Data type is integer.

Return value

Integer.

Usage

Word Pro: LeastRecentVersion method

{button ,AL(^H_VERSIONMANAGER_CLASS',0)} [See list of classes](#)

Syntax

[objectreference].LeastRecentVersion()

Parameters

None.

Return value

Integer.

Usage

Word Pro: LinkContainers method

```
{button ,AL(^H_BASECONTAINER_CLASS;H_CELLCONTAINER_CLASS;H_FRAMECONTAINER_CLASS;H_NOTECONTAINER_CLASS;H_RUBYCONTAINER_CLASS;H_DROPDOWNCONTAINER_CLASS;H_PAGECONTAINER_CLASS;H_SUBPAGECONTAINER_CLASS;H_SUPERPAGECONTAINER_CLASS;H_ROWCONTAINER_CLASS;H_SUPERTABLECONTAINER_CLASS;H_TABLECONTAINER_CLASS;H_PARALLELCOLSCONTAINER_CLASS;H_TABLEONLYCONT_CLASS',0)} See list of classes
```

Links the contents of selected container objects.

Syntax

[objectreference].LinkContainers()

Parameters

Return value

The return value for this method will always be -1 or 0. When testing the return value, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Equivalent to choosing Frame - Link Frame Contents. This method is only valid when called from frame container objects.

Word Pro: NewItemByPosition method

{button ,AL('H_MENUITEM_CLASS',0)} [See list of classes](#)

Allows you to create a new MenuItem object by specifying its position within the parent menu.

Syntax

[objectreference].NewItemByPosition(Caption, [p2,] [Position])

Parameters

Caption

A String parameter representing the name of the new menu item.

p2

An optional Variant parameter that initializes the Action property. This can be set to a subroutine name or a WMCommand ID. Default is 0.

Position

An optional Integer value which specifies the position at which the new menu item should appear. The topmost menu item corresponds to a position value of 0. If this parameter is omitted, the new menu item is placed at the end of its parent MenuItem object.

Return value

This method returns the new MenuItem object.

Usage

Allows you to insert a new menu item by specifying a caption and a numeric position.

You can specify a caption for the new menu item in the Caption parameter and an action for the menu item to perform in the p2 parameter. The p2 parameter is a Variant, so you can assign it a String subroutine name or an integer WMCommand ID.

When specifying a subroutine name in the p2 parameter, use the following syntax if the subroutine is contained within the current document:

```
"!SubroutineName"
```

Use the following syntax for the p2 parameter if the subroutine is contained within a separate file:

```
"Filename!SubroutineName"
```

Word Pro: NextCycleAlign method

{button ,AL(^H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

Syntax

[objectreference].NextCycleAlign([Forward])

Parameters

Forward

Data type is Integer. Optional parameter.

Return value

Integer.

Usage

Word Pro: NextCycleAttribute method

{button ,AL(^H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

Syntax

[objectreference].NextCycleAttribute([Forward])

Parameters

Forward

Data type is Integer. Optional parameter.

Return value

Integer.

Usage

Word Pro: NextCycleBullet method

{button ,AL(^H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

Syntax

[objectreference].NextCycleBullet([Forward])

Parameters

Forward

Data type is Integer. Optional parameter.

Return value

Integer.

Usage

Word Pro: NextCycleFontSize method

{button ,AL(^H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

Syntax

[objectreference].NextCycleFontSize([Forward])

Parameters

Forward

Data type is Integer. Optional parameter.

Return value

Integer.

Usage

Word Pro: NextCycleFont method

{button ,AL(^H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

Syntax

[objectreference].NextCycleFont([Forward])

Parameters

Forward

Data type is Integer. Optional parameter.

Return value

Integer.

Usage

Word Pro: NextCycleIndent method

{button ,AL(^H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

Syntax

[objectreference].NextCycleIndent([Forward])

Parameters

Forward

Data type is Integer. Optional parameter.

Return value

Integer.

Usage

Word Pro: NextCycleNumber method

{button ,AL(^H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

Syntax

[objectreference].NextCycleNumber([Forward])

Parameters

Forward

Data type is Integer. Optional parameter.

Return value

Integer.

Usage

Word Pro: NextCycleStyle method

{button ,AL('H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL('H_NEXTCYCLESTYLE_METHOD_EXSCRIPT',1)} [See example](#)

Syntax

[objectreference].NextCycleStyle([Forward])

Parameters

Forward

Data type is Integer. Optional parameter.

Return value

Integer.

Usage

Word Pro: OpenDocumentFromNotes method

{button ,AL('H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

Syntax

[ObjectReference].OpenDocumentFromNotes([UID], [Attached], [Field], [Database], [Server], [FileType], [Password])

Parameters

UID

A String expression which specifies Universal Identifier (UNID) for the Notes document to which you want to attach the active Word Pro document. The UNID can be found in the UniversalID property of the NotesDocument object in the Notes object model.

Attached

A String expression which specifies a name for the file you are attaching to the Notes document. The original active document retains its own name.

Field

A String expression which specifies the name of a Rich Text Field in the specified Notes document. The active Word Pro document will be saved as an attachment within the field you name.

Database

A String expression which specifies the name and path of the Notes database which contains the document you specified in the UID parameter.

Server

A String expression which specifies the name of the Notes server which contains the database you specified in the Database parameter.

FileType

An optional String value that indicates the file type in which the document will be saved. A null string saves the document as a Word Pro file. Some of the usual file types are listed in the table below, but each user's list of available file types is derived from the list of text filters installed during the Word Pro installation.

DCA/RFT	Lotus Manuscript 2.x	MS Word for Windows 1.0
DIF	Lotus Organizer 1.x	MS Word for Windows 2.0
DisplayWrite	Lotus Word Pro	MS Word for Windows 6.0
HTML	Lotus Word Pro SmartMaster	MS Word for Windows95 7.0
Lotus 1-2-3	MS Excel	MS WordPad 1.0
Lotus 1-2-3 for OS/2	MS Excel 3.0	OfficeWriter 4,5,6
Lotus 1-2-3 R3	MS Excel 4.0	Rich Text Format(RTF)
Lotus 1-2-3 R4,5	MS Excel 5.0	SAMNA Word
Lotus 1-2-3 R6	MS Excel 7.0	WordPerfect 5.0
Lotus Ami Pro	MS Windows Write 3.x	WordPerfect 5.1
Lotus Ami Pro 3.x Macro	MS Word for DOS 3,4,5,6	WordPerfect 6.x
Lotus Ami Pro 3.x Styles	MS Word for OS/2	WordStar 2000 R3

Password

Optional String parameter indicating the password for the document.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Word Pro: RemoveIndexEntry method

{button ,AL(^H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

Syntax

[objectreference].RemoveIndexEntry([All])

Parameters

All

Data type is Integer

Return value

Integer

Usage

Word Pro: RemoveList method

{button ,AL(^H_SMARTFILL_CLASS',0)} [See list of classes](#)

Syntax

[objectreference].RemoveList(ListName)

Parameters

Listname

Data type is String

Return value

Integer

Usage

Word Pro: RemoveTOCEntry method

{button ,AL(^H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

Syntax

[objectreference].RemoveTOCEntry([All])

Parameters

All

Data type is Integer

Return value

Integer

Usage

Word Pro: Repaint method

{button ,AL(^H_DOCWINDOW_CLASS',0)} [See list of classes](#)

This language element has not yet been defined. You can check for a newer Help file on the Lotus web site.

Syntax

[objectreference].Repaint()

Parameters

None

Return value

Integer

Usage

Word Pro: ReplaceContents method

{button ,AL(^H_CLICKHERE_CLASS',0)} [See list of classes](#)

Syntax

[objectreference].ReplaceContents(p1)

Parameters

p1

Data type is String.

Return value

Integer

Usage

Word Pro: ResetNumberOpts method

{button ,AL('H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

Syntax

[objectreference].ResetNumberOpts(ResetFlags, Style)

Parameters

ResetFlags

Data type is Variant, which allows the value of this parameter to be one of the constants listed below or its numeric equivalent (in parentheses). There is no default value.

<u>Value</u>	<u>Effect</u>
\$LwpNumberResetnone (2555)	
\$LwpNumberResetondivision (2349)	
\$LwpNumberResetonlevel (2350)	
\$LwpNumberResetonsection (2348)	
\$LwpNumberResetonspecificstyle (2351)	

Style

Data type is String.

Return value

Integer.

Usage

Word Pro: ResetPrinting method

{button ,AL(^H_PRINTMANAGER_CLASS',0)} [See list of classes](#)

Syntax

[objectreference].ResetPrinting()

Parameters

None.

Return value

Integer.

Usage

Word Pro: RestorePreviousView method

{button ,AL(^H_WINVIEWPREFS_CLASS',0)} [See list of classes](#)

{button ,AL(^H_RESTOREPREVIOUSVIEW_METHOD_EXSCRIPT',1)} [See example](#)

Restores the previously selected view type and zoom level.

Syntax

[objectreference].RestorePreviousView()

Parameters

None.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Word Pro: SaveMergeDataFile method

{button ,AL(^H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

Syntax

[objectreference].SaveMergeDataFile()

Parameters

None

Return value

Integer.

Usage

Word Pro: SetAllMargins method

```
{button ,AL(^H_LAYOUT_CLASS;H_CELLGROUPLAYOUT_CLASS;H_CELLLAYOUT_CLASS;H_CONNECTEDLAY  
OUT_CLASS;H_COLUMNGROUPLAYOUT_CLASS;H_FOOTERLAYOUT_CLASS;H_FRAMELAYOUT_CLASS;H  
_FRAMEGROUPLAYOUT_CLASS;H_DROP_CAPLAYOUT_CLASS;H_GROUPLAYOUT_CLASS;H_HEADERLAY  
OUT_CLASS;H_NOTELAYOUT_CLASS;H_PAGELAYOUT_CLASS;H_ROWGROUPLAYOUT_CLASS;H_ROWLA  
YOUT_CLASS;H_RUBYLAYOUT_CLASS;H_SUPERTABLEGROUPLAYOUT_CLASS;H_SUPERTABLELAYOUT  
_CLASS;H_TABLELAYOUT_CLASS;H_ENDNOTELAYOUT_CLASS;H_FOOTNOTELAYOUT_CLASS;H_TABLE  
HEADINGLAYOUT_CLASS;H_TOCSUPERTABLELAYOUT_CLASS',0)} See list of classes
```

```
{button ,AL(^H_SETALLMARGINS_METHOD_EXSCRIPT',1)} See example
```

Allows you to simultaneously set all margin and external margin values for a Layout object.

Syntax

[objectreference].SetAllMargins(flag, [l], [r], [t], [b], [lx], [rx], [tx], [bx])

Parameters

flag

The value of this Variant parameter must be one of the hexadecimal values below or a combination of the values.

<u>Value</u>	<u>Effect</u>
&H01	Sets the left margin value.
&H02	Sets the right margin value.
&H04	Sets the top margin value.
&H08	Sets the bottom margin value.
&H10	Sets the left external margin value.
&H20	Sets the right external margin value.
&H40	Sets the top external margin value.
&H80	Sets the bottom external margin value.

l

Data type is Long. Optional parameter which specifies the left margin value. The unit of measurement used for this property is Twips. There are 1440 twips per inch.

r

Data type is Long. Optional parameter which specifies the right margin value. The unit of measurement used for this property is Twips. There are 1440 Twips per inch.

t

Data type is Long. Optional parameter which specifies the top margin value. The unit of measurement used for this property is Twips. There are 1440 Twips per inch.

b

Data type is Long. Optional parameter which specifies the bottom margin value. The unit of measurement used for this property is Twips. There are 1440 Twips per inch.

lx

Data type is Long. Optional parameter which specifies the left external margin value. The unit of measurement used for this property is Twips. There are 1440 Twips per inch.

rx

Data type is Long. Optional parameter which specifies the right external margin value. The unit of measurement used for this property is Twips. There are 1440 Twips per inch.

tx

Data type is Long. Optional parameter which specifies the top external margin value. The unit of measurement used for this property is Twips. There are 1440 Twips per inch.

bx

Data type is Long. Optional parameter which specifies the bottom external margin value. The unit of measurement

used for this property is Twips. There are 1440 Twips per inch.

Return value

The return value for this method will always be -1 or 0. When testing the return value, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

You must provide values for any margins which are selected by the flag parameter. For example, if you set the flag parameter to &H01, then you must provide a value in the left margin parameter or the method will fail. However, since the l parameter is optional, no run-time error will occur.

Word Pro: SetCustomNumber method

{button ,AL('H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL('H_SETCUSTOMNUMBER_METHOD_EXSCRIPT',1)} [See example](#)

Syntax

[objectreference].SetCustomNumber(Restart, RestartSpecificStyleName, Include, TextBefore, Type, RepeatChar, StartAt, TextAfter)

Parameters

Restart

Data type is Variant, which allows the value of this parameter to be one of the constants listed below or its numeric equivalent (in parentheses). There is no default value.

<u>Value</u>	<u>Effect</u>
\$LwpNumberResetnone (2555)	
\$LwpNumberResetondivision (2349)	
\$LwpNumberResetonlevel (2350)	
\$LwpNumberResetonsection (2348)	
\$LwpNumberResetonspecificstyle (2351)	

RestartSpecificStyleName

Data type is String.

Include

Data type is Variant.

TextBefore

Data type is String.

Type

Data type is Variant.

RepeatChar

Data type is Integer.

StartAt

Data type is Integer.

TextAfter

Data type is String.

Return value

Integer.

Usage

Word Pro: SetIndexInfo method

{button ,AL(^H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

Syntax

[objectreference].SetIndexInfo(IndexName, Indented, IncludeAlphaSeeps)

Parameters

IndexName

Data type is String.

Indented

Data type is Integer.

IncludeAlphaSeeps

Data type is Integer.

Return value

Integer

Usage

Word Pro: SetJapanIndexInfo method

{button ,AL(^H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

This language element is available only in the Asian language versions of Word Pro..

Syntax

Parameters

Return value

Usage

Word Pro: SetNoFields method

{button ,AL(^H_MERGEOPTIONS_CLASS',0)} [See list of classes](#)

Syntax

[objectreference].SetNoFields()

Parameters

None

Return value

None

Usage

Word Pro: SetNumberingLevelInfo method

{button ,AL('H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL('H_SETNUMBERINGLEVELINFO_METHOD_EXSCRIPT',1)} [See example](#)

Syntax

[objectreference]SetNumberingLevelInfo(Name, Level, Restart, RestartSpecificStyleName, TextMask, Include, TextBefore, NumberMask, Type, RepeatChar, StartAt, TextAfter)

Parameters

Name

Data type is String.

Level

Data type is Integer.

Restart

Data type is Variant, which allows the value of this parameter to be one of the constants listed below or its numeric equivalent (in parentheses). There is no default value.

<u>Value</u>	<u>Effect</u>
\$LwpNumberResetnone (2555)	
\$LwpNumberResetondivision (2349)	
\$LwpNumberResetonlevel (2350)	
\$LwpNumberResetonsection (2348)	
\$LwpNumberResetonspecificstyle (2351)	

RestartSpecificStyleName

Data type is String.

TextMask

Data type is Integer.

Include

Data type is Variant.

TextBefore

Data type is String.

NumberMask

Data type is Integer.

Type

Data type is Variant.

RepeatChar

Data type is Integer.

StartAt

Data type is Integer.

TextAfter

Data type is String.

Return value

Integer.

Usage

Word Pro: SetTOCLevelContent method

{button ,AL('H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

Syntax

[objectreference].SetTocLevelContent(TOCName, Index, StyleName, Level, IncludeText, IncludeNumber)

Parameters

TOCName

Data type is String.

Index

Data type is Integer.

StyleName

Data type is String.

Level

Data type is Integer.

IncludeText

Data type is Integer.

IncludeNumber

Data type is Integer.

Return value

Integer.

Usage

Word Pro: SetTOCLevelPageInfo method

{button ,AL('H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

Syntax

[objectreference].SetTocLevelPageInfo(TocName, Index, PageNumber, RightAlignPageNumber, LeaderType)

Parameters

TocName

Data type is String.

Index

Data type is Integer.

PageNumber

Data type is Integer.

RightAlignPageNumber

Data type is Integer.

LeaderType

Data type is Variant.

Return value

Integer.

Usage

Word Pro: SpecialView method

{button ,AL('H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL('H_SPECIALVIEW_METHOD_EXSCRIPT',1)} [See example](#)

Syntax

[objectreference].SpecialView(p1)

Parameters

p1

Data type is Integer.

Return value

Integer.

Usage

{button ,AL('H_CLEAR SPLITS_METHOD_MEMDEF',0)} [See related topics](#)

Word Pro: SplitDivision method

{button ,AL('H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL('H_SPLITDIVISION_METHOD_EXSCRIPT',1)} [See example](#)

Syntax

[objectreference].SplitDivision()

Parameters

None.

Return value

Integer.

Usage

Word Pro: SplitWindow method

{button ,AL('H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL('H_SPLITWINDOW_METHOD_EXSCRIPT',1)} [See example](#)

Syntax

[objectreference].SplitWindow(p1)

Parameters

p1

Data type is Integer

Return value

Integer.

Usage

{button ,AL('H_CLEARSPLOTS_METHOD_MEMDEF',0)} [See related topics](#)

Word Pro: UnLinkContainers method

```
{button ,AL(^H_BASECONTAINER_CLASS;H_CELLCONTAINER_CLASS;H_FRAMECONTAINER_CLASS;H_NOTECONTAINER_CLASS;H_RUBYCONTAINER_CLASS;H_DROPDOWNCONTAINER_CLASS;H_PAGECONTAINER_CLASS;H_SUBPAGECONTAINER_CLASS;H_SUPERPAGECONTAINER_CLASS;H_ROWCONTAINER_CLASS;H_SUPERTABLECONTAINER_CLASS;H_TABLECONTAINER_CLASS;H_PARALLELCOLSCONTAINER_CLASS;H_TABLEONLYCONT_CLASS',0)} See list of classes
```

Unlinks the contents of the selected frame container objects.

Syntax

[objectreference].UnLinkContainers()

Parameters

Return value

The return value for this method will always be -1 or 0. When testing the return value, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Equivalent to choosing Frame - Unlink. The UnlinkContainers method is only valid when called from frame container objects.

Word Pro: Unlink method

{button ,AL(^H_DIVISION_CLASS;H_TEXTDOCUMENT_CLASS',0)} [See list of classes](#)

Syntax

[objectreference].Unlink()

Parameters

None

Return value

Integer

Usage

Word Pro: UpdatePowerFieldsOnNew method

{button ,AL(^H_DIVISION_CLASS;H_TEXTDOCUMENT_CLASS',0)} [See list of classes](#)

Syntax

[objectreference].UpdatePowerFieldsOnNew()

Parameters

None

Return value

Integer

Usage

Word Pro: UpdatePrinterBins method

{button ,AL(^H_PRINTMANAGER_CLASS',0)} [See list of classes](#)

Syntax

[objectreference].UpdatePrinterBins()

Parameters

None

Return value

Integer

Usage

Word Pro: UpdateTabs method

{button ,AL(^H_SECTIONTABS_CLASS',0)} [See list of classes](#)

Syntax

[objectreference].UpdateTabs()

Parameters

None.

Return value

Integer

Usage

Word Pro: UpdateUI method

{button ,AL(^H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

Syntax

[objectreference].UpdateUI()

Parameters

None

Return value

Integer

Usage

Word Pro: WordCount method

{button ,AL('H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL('H_WORDCOUNT_METHOD_EXSCRIPT',1)} [See example](#)

Syntax

[objectreference].WordCount()

Parameters

None

Return value

Integer

Usage

Example: AfidClassName property

'This example creates a drawing, then prints the Afid class name to the'
'LotusScript output panel

```
.CreateGraphic "WordProDraw", False  
Print .Graphic.AfidClassName
```

Example: AllowListEdit property

'This example creates a Click Here block which displays a popup list
'then allows the user to create new list items, and creates some items for the list.

```
Dim ClickHereName as String
ClickHereName = .InsertClickHere()
.Division.Foundry.ClickHeres(ClickHereName).Action = 11 ' Popup list
.Division.Foundry.ClickHeres(ClickHereName).AllowListEdit = True
' set 'No' and 'Yes' as items in the list.
.Division.Foundry.ClickHeres(ClickHereName).SetNamedProperty "LIST2", "No"
.Division.Foundry.ClickHeres(ClickHereName).SetNamedProperty "LIST1", "Yes"
```

Example: AllowListMultiValues property

'This example creates a Click Here block which displays a popup list
'then sets list to allow selection of multiple items, and creates some items for the
list.

```
Dim ClickHereName as String
```

```
ClickHereName = .InsertClickHere()
```

```
.Division.Foundry.ClickHeres(ClickHereName).Action = 11 ' Popup list
```

```
.Division.Foundry.ClickHeres(ClickHereName).AllowListMultiValues = True
```

```
' set 'No' and 'Yes' as items in the list.
```

```
.Division.Foundry.ClickHeres(ClickHereName).SetNamedProperty "LIST2", "No"
```

```
.Division.Foundry.ClickHeres(ClickHereName).SetNamedProperty "LIST1", "Yes"
```

Example: AnyOLEDEDELinks property

'This example uses the AnyOLEDEDELinks property to display a message telling
'the user if there are links in the current document.

```
If .ActiveDocument.DivisionInfo.AnyOleDdeLinks = True then
    MessageBox "This document contains a link.", MB_OK, "Example Script"
Else
    MessageBox "This document does not contain a link.", MB_OK, "Example Script"
End If
```

Example: ApplykSplitInfinitives property

'This example sets the grammar check option for flagging split infinitives
'to 2 in a row.

```
.ApplicationWindow.UserInterfacePrefs.GrammarOptions.ApplykSplitInfinitives = 1
```

Example: AsciiCodePage property

'This example sets options for importing or exporting of ASCII files.

'The code page used is DOS - ACII Code page 437.

.ApplicationWindow.Filter.IsAsciiCRLF = False

.ApplicationWindow.Filter.IsAsciiKeepStyle = 0

.ApplicationWindow.Filter.**AsciiCodePage** = 437

Example: BackgroundSpellingOn property

'This example asks the user whether to disable spell checking in the background,
'and then sets the appropriate option.

```
stat = MessageBox ("Do you want to disable background spell check?", 36, "Example  
Script")
```

```
If stat = 6 Then ' user said yes
```

```
    .Application.Preferences.BackgroundSpellingOn = False
```

```
Else
```

```
    .Application.Preferences.BackgroundSpellingOn = True
```

```
End If
```

Example: BookletPrinting property

'This example asks the user whether to print the document as a booklet,
'then sets the appropriate option, and prints the document.

```
stat = MessageBox ("Do you want to print as a booklet?", 36, "Example Script")
If stat = 6 Then ' user said yes
    .ActiveDocument.PrintSettings.BookletPrinting = True
    ' must print front to back for booklet printing
    .ActiveDocument.PrintSettings.PageOrder = $LtsPageOrderFrontToBack
Else
    .ActiveDocument.PrintSettings.BookletPrinting = False
End If
.Print
```

Example: Center property

'This example creates a watermark in the current layout, then centers
'the watermark within the layout.

```
.Page.Layout.RightPage.CreateLayer  
.Page.Layout.RightPage.Layer.ImportWatermarkGraphic .ApplicationWindow.UserInterfacePr  
efs.WatermarksPath & "_draft.wmf", "", False, False  
.Page.Layout.RightPage.Layer.Center = True  
.Page.Layout.RightPage.Layer.Tile = False
```

Example: ChangeKeyboardToLanguage property

'This example asks the user whether to automatically change the keyboard
'layout when the Word Pro language changes, and then sets the appropriate option.

```
stat = MessageBox ("Do you want to change the keyboard layout to match the language?",  
36, "Example Script")
```

```
If stat = 6 Then ' user said yes
```

```
    .Application.Preferences.ChangeKeyboardToLanguage = True
```

```
Else
```

```
    .Application.Preferences.ChangeKeyboardToLanguage = False
```

```
End If
```

Example: ChangeTextToMatchkeyboard property

'This example asks the user whether to change the text language when the
'user changes the keyboard layout, and then sets the appropriate option.

```
stat = MessageBox ("Do you want to change the text language to match the keyboard?",  
36, "Example Script")
```

```
If stat = 6 Then ' user said yes
```

```
    .Application.Preferences.ChangeTextToMatchkeyboard = True
```

```
Else
```

```
    .Application.Preferences.ChangeTextToMatchkeyboard = False
```

```
End If
```

Example: DdeEnabled property

'This example asks the user whether to disable DDE linking,
'and then sets the appropriate option.

```
stat = MessageBox ("Do you want to disable DDE linking?", 36, "Example Script")
If stat = 6 Then ' user said yes
    .Application.Preferences.DdeEnabled = False
Else
    .Application.Preferences.DdeEnabled = True
End If
```

Example: DiagonalLines property

'This example creates a table, and places diagonal lines in every cell.

```
.CreateTable False, "Default Table", 3, 2
.SelectEntireTable
.Table.TableLine.DiagonalLines.LeftBorder.Pattern = $LtsBorderPatternSolid
.Table.TableLine.DiagonalLines.LeftBorder.WidthInTwips = 10
'Make the line black
.Table.TableLine.DiagonalLines.LeftBorder.Color.Red = 0
.Table.TableLine.DiagonalLines.LeftBorder.Color.Blue = 0
.Table.TableLine.DiagonalLines.LeftBorder.Color.Green = 0
.Table.TableLine.DiagonalLines.LeftBorder.Color.Override = $LwpColorOverrideRgb
'Make the line left to right
.Table.TableLine.DiagonalLines.LinePlacement = &H1
'Effect the change
.Table.TableLine.ChgLineStyle $LwpTableLineStyleCustom
```

'Example: FirstDivision property

'This example inserts a new division into the document, and then displays

'the names of the first and last divisions in a message box.

```
.ApplicationWindow.SectionTabs.AddNewSectionTabs
```

```
FirstDiv = .ActiveDocument.Divisions(.ActiveDocument.FirstDivision).DivisionInfo.Name
```

```
LastDiv = .ActiveDocument.Divisions(.ActiveDocument.LastDivision).DivisionInfo.Name
```

```
MessageBox "The first division in this document is " & FirstDiv & _
```

```
    " and the last division is " & LastDiv & ".", MB_OK, "Example Script"
```


Example: FontMatching property

'This example asks the user whether to disable advanced font matching,
'and then sets the appropriate option.

```
stat = MessageBox ("Do you want to disable font matching?", 36, "Example Script")
If stat = 6 Then ' user said yes
    .Application.Preferences.FontMatching = False
Else
    .Application.Preferences.FontMatching = True
End If
```

```
'Example: GraphicPaths property
'This example displays the list of graphics paths in a message box.

AllPaths = ""
Forall GraphicPath In .ApplicationWindow.UserInterfacePrefs.GraphicPaths
    AllPaths = AllPaths & GraphicPath & ", "
End Forall
MessageBox "The graphic pathnames are: " & AllPaths, MB_OK, "Example Script"
```

```
'Example: GraphicPath property
'This example lets the user change the graphics path found in user preferences.
'If the user changes the number, the new path is saved.
```

```
GPath = .ApplicationWindow.UserInterfacePrefs.GraphicPath
NewPath = Inputbox ("Type a new graphics path:", "Example Script", GPath)
If NewPath <> "" Then ' user didn't cancel
    .ApplicationWindow.UserInterfacePrefs.InsertPath 678, NewPath
End If
```

Example: LastDivision property

'This example inserts a new division into the document, and then displays
'the names of the first and last divisions in a message box.

```
.ApplicationWindow.SectionTabs.AddNewSectionTabs  
FirstDiv = .ActiveDocument.Divisions(.ActiveDocument.FirstDivision).DivisionInfo.Name  
LastDiv = .ActiveDocument.Divisions(.ActiveDocument.LastDivision).DivisionInfo.Name  
MessageBox "The first division in this document is " & FirstDiv & _  
    " and the last division is " & LastDiv & ".", MB_OK, "Example Script"
```

Example: LayerName property

'This example checks the LayerName property to determine if a watermark layer
'already exists. The result of this check determines if the script should
'create a new layer, or delete the contents of the existing one.

```
If .Layout.LayerName = "" Then ' if no name, we need to create the layer
```

```
    .Layout.CreateLayer
```

```
Else ' we have one already; get rid of current contents
```

```
    .Layout.Layer.DeleteContents
```

```
End If
```

```
wmpath = .ApplicationWindow.UserInterfacePrefs.WatermarksPath
```

```
' place the watermark graphic
```

```
.Layout.Layer.ImportWatermarkGraphic wmpath & "_draft.wmf", "", False, False
```

Example: Layer property

'This example creates a new layer if one does not already exist, and then
'imports a watermark into the layer.

```
If .Layout.LayerName = "" Then ' if no name, we need to create the layer
```

```
    .Layout.CreateLayer
```

```
Else ' we have one already; get rid of current contents
```

```
    .Layout.Layer.DeleteContents
```

```
End If
```

```
wmpath = .ApplicationWindow.UserInterfacePrefs.WatermarksPath
```

```
' place the watermark graphic
```

```
.Layout.Layer.ImportWatermarkGraphic wmpath & "\_draft.wmf", "", False, False
```

```
'Example: MenuPaths property
'This example displays the list of menu paths in a message box.

AllPaths = ""
Forall MenuPath In .ApplicationWindow.UserInterfacePrefs.MenuPaths
    AllPaths = AllPaths & MenuPath & ", "
End Forall
MessageBox "The Menu pathnames are: " & AllPaths, MB_OK, "Example Script"
```

```
'Example: MenuPath property
'This example lets the user change the menu path found in user preferences.
'If the user changes the path, the new path is saved.

MPath = .ApplicationWindow.UserInterfacePrefs.MenuPath
NewPath = Inputbox ("Type a new menu path:", "Example Script", MPath)
If NewPath <> "" Then ' user didn't cancel
    .ApplicationWindow.UserInterfacePrefs.MenuPath = NewPath
End If
```



```
'Example: OCXDesignMode property
'This example asks the user whether design mode for OCX objects should be
'enabled, and then sets the appropriate option.

stat = MessageBox ("Do you want to enable design mode?", 36, "Example Script")
If stat = 6 Then ' user said yes
    .ActiveDocument.DocControl.OCXDesignMode = True
Else
    .ActiveDocument.DocControl.OCXDesignMode = False
End If
```

Example: OLEEnabled property

'This example asks the user whether to disable OLE drag and drop
'and then sets the appropriate option.

```
stat = MessageBox ("Do you want to disable OLE drag and drop?", 36, "Example Script")
```

```
If stat = 6 Then ' user said yes
```

```
    .Application.Preferences.OLEEnabled = False
```

```
Else
```

```
    .Application.Preferences.OLEEnabled = True
```

```
End If
```

Example: ParagraphHasDropCap property

'This example determines if the current paragraph has a drop cap.

'If not, a drop cap is created for the paragraph.

```
If .Text.ParagraphHasDropCap = False Then
```

```
    .CreateDropCap 3, 1
```

```
Else
```

```
    MessageBox "This paragraph already has a drop cap.", MB_OK, "Example Script"
```

```
End If
```

Example: ParagraphHasText property

'This example creates a drop cap if the current paragraph contains text
'and does not already have a drop cap.

```
If .Text.ParagraphHasDropCap = False And .Text.ParagraphHasText = True Then
    .CreateDropCap 3, 1
Else
    If .Text.ParagraphHasText = True Then
        MessageBox "This paragraph already has a drop cap.", MB_OK, "Example Script"
    Else
        MessageBox "This is an empty paragraph.", MB_OK, "Example Script"
    End If
End If
```

Example: Recursive property

'This example asks the user if files referenced by links in this document
'should also be exported to HTML:, then sets the appropriate option.

```
stat = MessageBox ("Do you want to include linked files in the HTML export?", 36,  
"Example Script")
```

```
If stat = 6 Then ' user said yes
```

```
    .ApplicationWindow.UserInterfacePrefs.HTMLOptions.Recursive = True
```

```
Else
```

```
    .ApplicationWindow.UserInterfacePrefs.HTMLOptions.Recursive = False
```

```
End If
```

Example: RemoveHyperlink method

'This example inserts a Hyperlink into the current document
'at the insertion point. Then it moves the insertion point
'into the hyperlink and runs the RemoveHyperlink method
'which removes the code but leaves the "My Linked Text"

```
.InsertClickHereLink "http://www.lotus.com", "", "", "", "Lotus Development Corporation"
```

```
MessageBox "Click OK to delete the Hyperlink", MB_OK, "Example Script"
```

'move the insertion point into the Hyperlink

```
.Type "[Left][Left][Left][Left]"
```

```
.RemoveHyperlink
```

Example: RetainNameOfImportedFile property

'This example asks the user whether to keep the name of imported files,
'and then sets the appropriate option.

```
stat = MessageBox ("Do you want to use the name of imported files?", 36, "Example  
Script")
```

```
If stat = 6 Then ' user said yes
```

```
    .ApplicationWindow.UserInterfacePrefs.RetainNameOfImportedFile = True
```

```
Else
```

```
    .ApplicationWindow.UserInterfacePrefs.RetainNameOfImportedFile = False
```

```
End If
```

Example: SetupForCropping method

'This example imports a graphic, then crops it.

'get the graphic path and import the graphic.

```
PicFile = .ApplicationWindow.UserInterfacePrefs.GraphicPath & "\_wpdon.gif"
```

```
.ImportGraphic PicFile, ".gif", False, False, "Default Graphic/OLE"
```

' get ready to crop

```
.Layout.SetupForCropping
```

'change the size of the frame holding the graphic

```
.Layout.Width = 1479
```

```
.Layout.Height = 1539
```

'and set up the offset for the graphic within the frame

```
.Layout.XOffset = -570
```

```
.Layout.YOffset = 285
```


Example: ShowBubbleHelp property

'This example asks the user whether to display bubble help for SmartIcons
'and then sets the appropriate option.

```
stat = MessageBox ("Do you want to display Bubble Help?", 36, "Example Script")
If stat = 6 Then ' user said yes
    .ApplicationWindow.UserInterfacePrefs.ShowBubbleHelp = 1
Else
    .ApplicationWindow.UserInterfacePrefs.ShowBubbleHelp = 0
End If
```

Example: ShowFileNew property

'This example asks the user whether to display the File-New dialog box
'when the new file SmartIcon is clicked, and then sets the appropriate option.

```
stat = MessageBox ("Do you want to show the File-New dialog box?", 36, "Example  
Script")
```

```
If stat = 6 Then ' user said yes
```

```
    .ApplicationWindow.UserInterfacePrefs.ShowFileNew = 1
```

```
Else
```

```
    .ApplicationWindow.UserInterfacePrefs.ShowFileNew = 0
```

```
End If
```

Example: ShowMenuCustomization property

'This example asks the user whether to allow use of the menu customization
'tool, and then sets the appropriate option.

```
stat = MessageBox ("Do you want to enable the menu customization option?", 36,  
"Example Script")
```

```
If stat = 6 Then ' user said yes
```

```
    .ApplicationWindow.UserInterfacePrefs.ShowMenuCustomization = True
```

```
Else
```

```
    .ApplicationWindow.UserInterfacePrefs.ShowMenuCustomization = False
```

```
End If
```

Example: ShowMenuWarningMessage property

'This example asks the user whether to display a warning message when changing
'the default menu set, and then sets the appropriate option.

```
stat = MessageBox ("Do you want to display an alert when changing menus?", 36,  
"Example Script")
```

```
If stat = 6 Then ' user said yes
```

```
    .ApplicationWindow.UserInterfacePrefs.ShowMenuWarningMessage = True
```

```
Else
```

```
    .ApplicationWindow.UserInterfacePrefs.ShowMenuWarningMessage = False
```

```
End If
```

Example: Tile property

'This example creates a watermark in the current layout, then tiles the
'watermark graphic within the layout.

```
.Layout.CreateLayer  
.Layout.Layer.ImportWatermarkGraphic .ApplicationWindow.UserInterfacePrefs.WatermarksP  
ath & "_draft.wmf", "", False, False  
.Layout.Layer.Center = False  
.Layout.Layer.Tile = True
```

Example: UserDefinedFilter property

'This example sets the default file types in the Browse dialog box to Word Pro and Ami Pro files.

```
.ApplicationWindow.UserInterfacePrefs.UserDefinedFilter = "*.lwp;*.sam"
```

```
'Turn off working type option so the filter takes effect.
```

```
.ApplicationWindow.UserInterfacePrefs.WorkingType = False
```

Example: WaterMarksPath property

'This example creates a new layer if one does not already exist, and then
'imports a watermark into the layer.

```
If .Layout.LayerName = "" Then ' if no name, we need to create the layer
```

```
    .Layout.CreateLayer
```

```
Else ' we have one already; get rid of current contents
```

```
    .Layout.Layer.DeleteContents
```

```
End If
```

```
wmpath = .ApplicationWindow.UserInterfacePrefs.WatermarksPath
```

```
' place the watermark graphic
```

```
.Layout.Layer.ImportWatermarkGraphic wmpath & "\_draft.wmf", "", False, False
```

'Example: ApplyDoubleNegative property

'This example asks the user whether to check for double negatives in

'grammar check, and then sets the appropriate option.

```
stat = MessageBox ("Do you want to check for double negatives?", 36, "Example Script")
```

```
If stat = 6 Then ' user said yes
```

```
    .ApplicationWindow.UserInterfacePrefs.GrammarOptions.ApplyDoubleNegative = True
```

```
Else
```

```
    .ApplicationWindow.UserInterfacePrefs.GrammarOptions.ApplyDoubleNegative = False
```

```
End If
```


'Example: ApplyDoubleWordCheck property

'This example asks the user whether to check for double words in

'grammar check, and then sets the appropriate option.

```
stat = MessageBox ("Do you want to check for repeated words?", 36, "Example Script")
```

```
If stat = 6 Then ' user said yes
```

```
    .ApplicationWindow.UserInterfacePrefs.GrammarOptions.ApplyDoubleWordCheck = True
```

```
Else
```

```
    .ApplicationWindow.UserInterfacePrefs.GrammarOptions.ApplyDoubleWordCheck = False
```

```
End If
```

```
'Example: ApplyExtraPrepositionCheck property
'This example asks the user whether to check for superfluous prepositions in
'grammar check, and then sets the appropriate option.

stat = MessageBox ("Do you want to check for extra prepositions?", 36, "Example
Script")
If stat = 6 Then ' user said yes
    .ApplicationWindow.UserInterfacePrefs.GrammarOptions.ApplyExtraPrepositionCheck =
True
Else
    .ApplicationWindow.UserInterfacePrefs.GrammarOptions.ApplyExtraPrepositionCheck =
False
End If
```

```
'Example: ApplyFormatErrors property
'This example asks the user whether to check for errors in document formatting in
'grammar check, and then sets the appropriate option.

stat = MessageBox ("Do you want to check for formatting errors?", 36, "Example
Script")
If stat = 6 Then ' user said yes
    .ApplicationWindow.UserInterfacePrefs.GrammarOptions.ApplyFormatErrors = True
Else
    .ApplicationWindow.UserInterfacePrefs.GrammarOptions.ApplyFormatErrors = False
End If
```

'Example: ApplyGenderExpressions property

'This example asks the user whether to check for gender specific expressions in
'grammar check, and then sets the appropriate option.

```
stat = MessageBox ("Do you want to check for gender expressions?", 36, "Example  
Script")
```

```
If stat = 6 Then ' user said yes
```

```
    .ApplicationWindow.UserInterfacePrefs.GrammarOptions.ApplyGenderExpressions = True
```

```
Else
```

```
    .ApplicationWindow.UserInterfacePrefs.GrammarOptions.ApplyGenderExpressions =
```

```
False
```

```
End If
```

```
'Example: ApplyHomonyms property
'This example asks the user whether to check for correct usage of homonyms in
'grammar check, and then sets the appropriate option.

stat = MessageBox ("Do you want to check for homonym errors?", 36, "Example Script")
If stat = 6 Then ' user said yes
    .ApplicationWindow.UserInterfacePrefs.GrammarOptions.ApplyHomonyms = True
Else
    .ApplicationWindow.UserInterfacePrefs.GrammarOptions.ApplyHomonyms = False
End If
```

'Example: ApplyIncorrectPlural property

'This example asks the user whether to check for correct usage of plurals in
'grammar check, and then sets the appropriate option.

```
stat = MessageBox ("Do you want to check for plural errors?", 36, "Example Script")
```

```
If stat = 6 Then ' user said yes
```

```
    .ApplicationWindow.UserInterfacePrefs.GrammarOptions.ApplyIncorrectPlural = True
```

```
Else
```

```
    .ApplicationWindow.UserInterfacePrefs.GrammarOptions.ApplyIncorrectPlural = False
```

```
End If
```

```
'Example: ApplyInformalExpressions property
'This example asks the user whether to check for overuse of informal
'expressions in grammar check, and then sets the appropriate option.

stat = MessageBox ("Do you want to check for informal expressions?", 36, "Example
Script")
If stat = 6 Then ' user said yes
    .ApplicationWindow.UserInterfacePrefs.GrammarOptions.ApplyInformalExpressions =
True
Else
    .ApplicationWindow.UserInterfacePrefs.GrammarOptions.ApplyInformalExpressions =
False
End If
```

```
'Example: ApplyJargonWords property
'This example asks the user whether to check for overuse of jargon words in
'grammar check, and then sets the appropriate option.

stat = MessageBox ("Do you want to check for jargon words?", 36, "Example Script")
If stat = 6 Then ' user said yes
    .ApplicationWindow.UserInterfacePrefs.GrammarOptions.ApplyJargonWords = True
Else
    .ApplicationWindow.UserInterfacePrefs.GrammarOptions.ApplyJargonWords = False
End If
```



```
'Example: ApplyMassVsCount property
'This example asks the user whether to check for correct use of mass and count
'modifiers in grammar check, and then sets the appropriate option.

stat = MessageBox ("Do you want to check for correct use of less and fewer?", 36,
"Example Script")
If stat = 6 Then ' user said yes
    .ApplicationWindow.UserInterfacePrefs.GrammarOptions.ApplyMassVsCount = True
Else
    .ApplicationWindow.UserInterfacePrefs.GrammarOptions.ApplyMassVsCount = False
End If
```

```
'Example: ApplyMisspelledExpressions property
'This example asks the user whether to check for correct spelling of expressions in
'grammar check, and then sets the appropriate option.

stat = MessageBox ("Do you want to check for misspelled expressions?", 36, "Example
Script")
If stat = 6 Then ' user said yes
    .ApplicationWindow.UserInterfacePrefs.GrammarOptions.ApplyMisspelledExpressions =
True
Else
    .ApplicationWindow.UserInterfacePrefs.GrammarOptions.ApplyMisspelledExpressions =
False
End If
```

```
'Example: ApplyMisspelledForeignExpressions property
'This example asks the user whether to check for correctly spelled foreign
'expressions in grammar check, and then sets the appropriate option.

stat = MessageBox ("Do you want to check for correct foreign spelling?", 36, "Example
Script")
If stat = 6 Then ' user said yes
    .ApplicationWindow.UserInterfacePrefs.GrammarOptions.ApplyMisspelledForeignExpress
ions = True
Else
    .ApplicationWindow.UserInterfacePrefs.GrammarOptions.ApplyMisspelledForeignExpress
ions = False
End If
```

'Example: ApplyMisusedWords property

'This example asks the user whether to check for improper use of words in

'grammar check, and then sets the appropriate option.

```
stat = MessageBox ("Do you want to check for misused words?", 36, "Example Script")
If stat = 6 Then ' user said yes
    .ApplicationWindow.UserInterfacePrefs.GrammarOptions.ApplyMisusedWords = True
Else
    .ApplicationWindow.UserInterfacePrefs.GrammarOptions.ApplyMisusedWords = False
End If
```

'Example: ApplyNonStandardExpression property
'This example asks the user whether to check for correct use of expressions in
'grammar check, and then sets the appropriate option.

```
stat = MessageBox ("Do you want to check for expression errors?", 36, "Example  
Script")  
If stat = 6 Then ' user said yes  
    .ApplicationWindow.UserInterfacePrefs.GrammarOptions.ApplyNonStandardExpression =  
True  
Else  
    .ApplicationWindow.UserInterfacePrefs.GrammarOptions.ApplyNonStandardExpression =  
False  
End If
```

'Example: ApplyNonStandardModifiers property

'This example asks the user whether to check for correct use of modifiers in

'grammar check, and then sets the appropriate option.

```
stat = MessageBox ("Do you want to check for non-standard modifiers?", 36, "Example  
Script")
```

```
If stat = 6 Then ' user said yes
```

```
    .ApplicationWindow.UserInterfacePrefs.GrammarOptions.ApplyNonStandardModifiers =  
True
```

```
Else
```

```
    .ApplicationWindow.UserInterfacePrefs.GrammarOptions.ApplyNonStandardModifiers =  
False
```

```
End If
```

```
'Example: ApplyNoudModifierOrderCheck property
'This example asks the user whether to check for correct order of noun
'modifiers in grammar check, and then sets the appropriate option.

stat = MessageBox ("Do you want to check for correct modifier order?", 36, "Example
Script")
If stat = 6 Then ' user said yes
    .ApplicationWindow.UserInterfacePrefs.GrammarOptions.ApplyNoudModifierOrderCheck =
True
Else
    .ApplicationWindow.UserInterfacePrefs.GrammarOptions.ApplyNoudModifierOrderCheck =
False
End If
```

'Example: ApplyNounConsistency property

'This example asks the user whether to check for correct use of noun phrases in
'grammar check, and then sets the appropriate option.

```
stat = MessageBox ("Do you want to check for noun consistency?", 36, "Example Script")
```

```
If stat = 6 Then ' user said yes
```

```
    .ApplicationWindow.UserInterfacePrefs.GrammarOptions.ApplyNounConsistency = True
```

```
Else
```

```
    .ApplicationWindow.UserInterfacePrefs.GrammarOptions.ApplyNounConsistency = False
```

```
End If
```


'Example: ApplyOpenClosedSpelling property

'This example asks the user whether to check for correct spacing and spelling

'of word pairs in grammar check, and then sets the appropriate option.

```
stat = MessageBox ("Do you want to check for correct spelling?", 36, "Example Script")
```

```
If stat = 6 Then ' user said yes
```

```
    .ApplicationWindow.UserInterfacePrefs.GrammarOptions.ApplyOpenClosedSpelling =  
True
```

```
Else
```

```
    .ApplicationWindow.UserInterfacePrefs.GrammarOptions.ApplyOpenClosedSpelling =  
False
```

```
End If
```

'Example: ApplyPrepositionalPhrases property

'This example sets the option for flagging consecutive prepositional

'phrase errors in the document to 3 prepositional phrases in a row.

.ApplicationWindow.UserInterfacePrefs.GrammarOptions.**ApplyPrepositionalPhrases** = 1

'Example: ApplyPretentiousWords property

'This example asks the user whether to check for pretentious words in

'grammar check, and then sets the appropriate option.

```
stat = MessageBox ("Do you want to check for pretentious words?", 36, "Example  
Script")
```

```
If stat = 6 Then ' user said yes
```

```
    .ApplicationWindow.UserInterfacePrefs.GrammarOptions.ApplyPretentiousWords = True
```

```
Else
```

```
    .ApplicationWindow.UserInterfacePrefs.GrammarOptions.ApplyPretentiousWords = False
```

```
End If
```

'Example: ApplyPronounErrors property

'This example asks the user whether to check for errors in pronoun usage in
'grammar check, and then sets the appropriate option.

```
stat = MessageBox ("Do you want to check for pronoun errors?", 36, "Example Script")
If stat = 6 Then ' user said yes
    .ApplicationWindow.UserInterfacePrefs.GrammarOptions.ApplyPronounErrors = True
Else
    .ApplicationWindow.UserInterfacePrefs.GrammarOptions.ApplyPronounErrors = False
End If
```

'Example: ApplyRedundantExpressions property
'This example asks the user whether to check for phrases that are redundant terms in
'grammar check, and then sets the appropriate option.

```
stat = MessageBox ("Do you want to check for redundant phrases?", 36, "Example  
Script")  
If stat = 6 Then ' user said yes  
    .ApplicationWindow.UserInterfacePrefs.GrammarOptions.ApplyRedundantExpressions =  
True  
Else  
    .ApplicationWindow.UserInterfacePrefs.GrammarOptions.ApplyRedundantExpressions =  
False  
End If
```

'Example: ApplySexistExpressions property

'This example asks the user whether to check for sexist terms in

'grammar check, and then sets the appropriate option.

```
stat = MessageBox ("Do you want to check for sexist expressions?", 36, "Example  
Script")
```

```
If stat = 6 Then ' user said yes
```

```
    .ApplicationWindow.UserInterfacePrefs.GrammarOptions.ApplySexistExpressions = True
```

```
Else
```

```
    .ApplicationWindow.UserInterfacePrefs.GrammarOptions.ApplySexistExpressions =
```

```
False
```

```
End If
```

```
'Example: ApplySubjectVerbAgreement property
'This example asks the user whether to check for agreement between subjects
'and verbs in grammar check, and then sets the appropriate option.

stat = MessageBox ("Do you want to check for subject/verb agreement?", 36, "Example
Script")
If stat = 6 Then ' user said yes
    .ApplicationWindow.UserInterfacePrefs.GrammarOptions.ApplySubjectVerbAgreement =
True
Else
    .ApplicationWindow.UserInterfacePrefs.GrammarOptions.ApplySubjectVerbAgreement =
False
End If
```

'Example: ApplyVagueQuantifiers property

'This example asks the user whether to check for vague use of numeric quantifiers in
'grammar check, and then sets the appropriate option.

```
stat = MessageBox ("Do you want to check for vague quantifiers?", 36, "Example  
Script")
```

```
If stat = 6 Then ' user said yes
```

```
    .ApplicationWindow.UserInterfacePrefs.GrammarOptions.ApplyVagueQuantifiers = True
```

```
Else
```

```
    .ApplicationWindow.UserInterfacePrefs.GrammarOptions.ApplyVagueQuantifiers = False
```

```
End If
```



```
'Example: ApplyVerbGroupConsistency property
'This example asks the user whether to check for verb group consistency in
'grammar check, and then sets the appropriate option.

stat = MessageBox ("Do you want to check for correct verb usage?", 36, "Example
Script")
If stat = 6 Then ' user said yes
    .ApplicationWindow.UserInterfacePrefs.GrammarOptions.ApplyVerbGroupConsistency =
True
Else
    .ApplicationWindow.UserInterfacePrefs.GrammarOptions.ApplyVerbGroupConsistency =
False
End If
```

```
'Example: ApplyWeakModifiers property
'This example asks the user whether to check for weak adjectives and adverbs in
'grammar check, and then sets the appropriate option.

stat = MessageBox ("Do you want to check for weak modifiers?", 36, "Example Script")
If stat = 6 Then ' user said yes
    .ApplicationWindow.UserInterfacePrefs.GrammarOptions.ApplyWeakModifiers = True
Else
    .ApplicationWindow.UserInterfacePrefs.GrammarOptions.ApplyWeakModifiers = False
End If
```

```
'Example: AppViewPrefs property  
' This example displays the red, green, and blue values of the margin color in  
' the LotusScript output panel
```

```
Print .AppViewPrefs.MarginColor.Red  
Print .AppViewPrefs.MarginColor.Green  
Print .AppViewPrefs.MarginColor.Blue
```

```
'Example: AtBeginningOfLine property
'This example inserts text into the current document, then determines if
'the caret is at the beginning of the line.

.Text.InsertText "Now is the time for all good men to come to the aid of their
party.", False
If .Text.AtBeginningOfLine = True Then
    MessageBox "The text is at the beginning of the line", MB_OK, "Example Script"
Else
    MessageBox "The text is not at the beginning of the line", MB_OK, "Example Script"
End If
.Text.Backward $LwpNavigateObjectTypeSentence, 1
If .Text.AtBeginningOfLine = True Then
    MessageBox "The text is at the beginning of the line", MB_OK, "Example Script"
Else
    MessageBox "The text is not at the beginning of the line", MB_OK, "Example Script"
End If
```

```
'Example: AtBeginning property
'This example inserts text into the current document, then tests to see
'if the insertion point is at the beginning of the text stream.

.Text.InsertText "Now is the time for all good men to come to the aid of their
party.", True
If .Text.AtBeginning = True Then
    MessageBox "The text is at the beginning of the stream", MB_OK, "Example Script"
Else
    MessageBox "The text is not at the beginning of the stream", MB_OK, "Example
Script"
End If
.Text.Backward $LwpNavigateObjectTypeParagraph, 1
If .Text.AtBeginning = True Then
    MessageBox "The text is at the beginning of the stream", MB_OK, "Example Script"
Else
    MessageBox "The text is not at the beginning of the stream", MB_OK, "Example
Script"
End If
```

```
'Example: AtEndOfLine property
'This example inserts text into the current document, then determines if
'the caret is at the end of the line.

.Text.InsertText "Now is the time for all good men to come to the aid of their
party.", False
If .Text.AtEndOfLine = True Then
    MessageBox "The caret is at the end of the line", MB_OK, "Example Script"
Else
    MessageBox "The caret is not at the end of the line", MB_OK, "Example Script"
End If
.Text.Backward $LwpNavigateObjectTypeSentence, 1
If .Text.AtEndOfLine = True Then
    MessageBox "The caret is at the end of the line", MB_OK, "Example Script"
Else
    MessageBox "The caret is not at the end of the line", MB_OK, "Example Script"
End If
```

```
'Example: AtEnd property
'This example inserts text into the current document, then tests to see
'if the insertion point is at the end of the text stream.

.Text.InsertText "Now is the time for all good men to come to the aid of their
party.", True
If .Text.AtEnd = True Then
    MessageBox "The caret is at the end of the stream", MB_OK, "Example Script"
Else
    MessageBox "The caret is not at the end of the stream", MB_OK, "Example Script"
End If
.Text.Backward $LwpNavigateObjectTypeParagraph, 1
If .Text.AtEnd = True Then
    MessageBox "The caret is at the end of the stream", MB_OK, "Example Script"
Else
    MessageBox "The caret is not at the end of the stream", MB_OK, "Example Script"
End If
```

'Example: Attributes property

'This example enables the HighlightMode attribute, types text with the mode
'on, then disables the attribute.

.Text.**Attributes**.HighlightMode = True

.Type "Now is the time for all good men to come to the aid of their party."

.Text.**Attributes**.HighlightMode = False


```
'Example: AttrStyleName property
'This example determines the hierarchy of the current paragraph style's
'attributes attribute.  If attributes are inherited, the user is given a
'chance to make it local.

If .Text.ParagraphStyle.AttrStyleName = "" Then 'this style's attributes are local
    MsgBox .Text.ParagraphStyle.Name & " stores its attributes locally.", MB_OK,
    "Example Script"
Else ' Give user a chance to make it local
    Stat = MsgBox (.Text.ParagraphStyle.Name & " inherits its attributes from "
    & .Text.ParagraphStyle.AttrStyleName & ". Do you want to make it local?", 36, "Example
    Script")
    If stat = 6 Then 'user said yes
        'get the current hierarchy, and add attributes to the local attrs.
        StyleAttrs = .Text.ParagraphStyle.Definition + &H4
        .SetStyle $LwpStyleTypeParagraph, .Text.ParagraphStyle.Name, StyleAttrs
        ' set the attribute style name to the empty string.
        .Text.ParagraphStyle.AttrStyleName = ""
    End If
End If
```

```
'Example: AuthorName property
With WordPro.ActiveDocument
MsgTxt = "Current Word Pro Doc is " & .FullName
MsgTxt = MsgTxt & ", the author is " & .Docinfo.AuthorName
Msgbox MsgTxt,64,"Word Pro Information"
MsgTxt = "It was created on " & .DocInfo.CreationDateString & " at "
& .DocInfo.CreationTimeString
Msgbox MsgTxt,64,"Word Pro Information"
End With
```

```
'Example: AutoBackup property
'This example asks the user whether to back up documents automatically,
'and then sets the appropriate option.

stat = MessageBox ("Do you want to back up documents?", 36, "Example Script")
If stat = 6 Then ' user said yes
    .ApplicationWindow.UserInterfacePrefs.AutoBackup = True
Else
    .ApplicationWindow.UserInterfacePrefs.AutoBackup = False
End If
```

```
'Example: AutoHyphenate property
'This example sets hyphenation options for a new document.

.NewDocument
For i = 1 To 20 ' put some text in to demonstrate
    .type "testing hyphenation "
Next
' turn on auto hyphenation
.ActiveDocument.DivisionOptions.HyphenationOptions.AutoHyphenate = True
.Text.Attributes.NoHyphenate = False ' turn on local hyphenation
MessageBox "Click OK to disable hyphenation", MB_OK, "Example Script"
.ActiveDocument.DivisionOptions.HyphenationOptions.AutoHyphenate = False
```

'Example: AutoSaveMinutes property

'This example asks the user whether to enable auto timed save. If auto timed

'save is enabled, the interval is set to 15 minutes.

```
stat = MessageBox ("Do you want to use auto timed save?", 36, "Example Script")
```

```
If stat = 6 Then ' user said yes
```

```
    .ApplicationWindow.UserInterfacePrefs.AutoSave = True
```

```
    .ApplicationWindow.UserInterfacePrefs.AutoSaveMinutes = 15
```

```
Else
```

```
    .ApplicationWindow.UserInterfacePrefs.AutoSave = False
```

```
End If
```

'Example: AutoSave property

'This example asks the user whether to enable auto timed save. If auto timed

'save is enabled, the interval is set to 15 minutes.

```
stat = MessageBox ("Do you want to use auto timed save?", 36, "Example Script")
```

```
If stat = 6 Then ' user said yes
```

```
    .ApplicationWindow.UserInterfacePrefs.AutoSave = True
```

```
    .ApplicationWindow.UserInterfacePrefs.AutoSaveMinutes = 15
```

```
Else
```

```
    .ApplicationWindow.UserInterfacePrefs.AutoSave = False
```

```
End If
```

'Example: AutoVersion property

'This example sets the document to automatically create a new version
'each time it is opened.

.ActiveDocument.DocControl.**AutoVersion** = \$LwpAutoVersionOnfileopen

'Example: BackColor property

'This example creates a frame, and sets the background to blue, and the
'background pattern to 50% gray scale. The pattern color is then set
'to white.

```
.NewFrame 1521, 2151, 3492, 1812, "Default Frame"  
.Frame.Layout.Background.Color.Red = 0  
.Frame.Layout.Background.Color.Blue = 255  
.Frame.Layout.Background.Color.Green = 0  
.Frame.Layout.Background.Color.Override = $LwpColorOverrideRgb  
.Frame.Layout.Background.Pattern = $LtsFillGray5  
.Frame.Layout.Background.BackColor.Red = 255  
.Frame.Layout.Background.BackColor.Blue = 255  
.Frame.Layout.Background.BackColor.Green = 255  
.Frame.Layout.Background.BackColor.Override = $LwpColorOverrideRgb
```


'Example: BackgroundPrintingOn property

'This example asks the user whether to disable background printing,

'and then sets the appropriate option.

```
stat = MessageBox ("Do you want to disable background printing?", 36, "Example  
Script")
```

```
If stat = 6 Then ' user said yes
```

```
    .Application.Preferences.BackgroundPrintingOn = False
```

```
Else
```

```
    .Application.Preferences.BackgroundPrintingOn = True
```

```
End If
```

'Example: Background property

'This example creates a table, then sets the table's background color and
'patthen. It then sets the current cell's background color

```
.CreateTable False, "Default Table", 2, 2
'Set table's background color to blue
.SuperTableContainer.Layout.Background.Color.Red = 0
.SuperTableContainer.Layout.Background.Color.Blue = 255
.SuperTableContainer.Layout.Background.Color.Green = 0
.SuperTableContainer.Layout.Background.Color.Override = $LwpColorOverrideRgb
' set the background pattern to gradient fill
.SuperTableContainer.Layout.Background.Pattern = $LtsFillLeftRightGrad
' set the pattern color to yellow
.SuperTableContainer.Layout.Background.BackColor.Red = 255
.SuperTableContainer.Layout.Background.BackColor.Blue = 0
.SuperTableContainer.Layout.Background.BackColor.Green = 255
.SuperTableContainer.Layout.Background.BackColor.Override = $LwpColorOverrideRgb
'set the table cell's background color to red.
.Table.TableFill.Background.Pattern = $LtsFillSolid
.Table.TableFill.Background.Color.Red = 255
.Table.TableFill.Background.Color.Blue = 0
.Table.TableFill.Background.Color.Green = 0
.Table.TableFill.Background.Color.Override = $LwpColorOverrideRgb
.Table.TableFill.FillStyle = $LwpTableFillStyleAll
```

```
'Example: Backspace method
' This example inserts 5 words into the current document and then backspaces
' 5 characters.
' RUNTIME DEPENDENCIES: You must have a document open for this script to work.

Dim WordNumber as Integer
For WordNumber = 1 To 5
    .Text.InsertText "Word" & Format$(WordNumber) & " "
Next
MessageBox "Click OK to backspace 5 characters.",MB_OK,"Example Script"
.Text.BackSpace (5)
```

```
'Example: BackupPath property
'This example lets the user change the backup path found in user preferences.
'If the user changes the path, the new path is saved.
```

```
BPath = .ApplicationWindow.UserInterfacePrefs.BackupPath
NewPath = Inputbox ("Type a new backup path:", "Example Script", BPath)
If NewPath <> "" Then ' user didn't cancel
    .ApplicationWindow.UserInterfacePrefs.InsertPath 2085, NewPath
End If
```

```
'Example: Backward method
' This example inserts 5 sentences with 5 words each into the current
' document. The cursor location is then moved backward 2 sentences and then
' moved forward one word.
' RUNTIME DEPENDENCIES: You must have a document open for this script to work.
```

```
Dim SentenceNumber as Integer
Dim WordNumber as Integer
For SentenceNumber = 1 To 5
    For WordNumber = 1 To 5
        .Text.InsertText "Word" & Format$(WordNumber) & " "
    Next
    .Type (". ")
Next
MessageBox "Click OK to backup 2 sentences.",MB_OK,"Example Script"
.Text.Backward $LwpNavigateObjectTypeSentence, 2
MessageBox "Click OK to go forward 1 word.",MB_OK,"Example Script"
.Text.Forward $LwpNavigateObjectTypeWord, 1
```

Word Pro: ApplySensitiveExp property

{button ,AL('H_GRAMMAR_CLASS',0)} [See list of classes](#)

(Read-write) A flag that turns the specific Grammar Check rule on or off.

Data Type

Integer

Syntax

appliesensitiveexpvalue = [objectreference].ApplySensitiveExp

[objectreference].ApplySensitiveExp = appliesensitiveexpvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Use this property when you are setting the corresponding rule for proofing the document in Grammar Check. Equivalent to choosing Edit - Check Grammar, clicking Options, and selecting the appropriate rule in the "Rule type" field on the Rules panel.

Note This rule is not applicable in every language. Some Apply properties can only be found in specific language versions of Word Pro.

Word Pro: ApplySexistExpressions property

{button ,AL('H_GRAMMAR_CLASS',0)} [See list of classes](#)

{button ,AL('H_APPLYSEXISTEXPRESSIONS_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) A flag that turns the Grammar Check rule on or off for checking sexist expressions.

Data Type

[Integer](#)

Syntax

applysexistexpressionsvalue = [objectreference].ApplySexistExpressions

[objectreference].ApplySexistExpressions = applysexistexpressionsvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Use this property when you are setting the corresponding rule for proofing the document in Grammar Check.

Equivalent to choosing Edit - Check Grammar, clicking Options, and selecting "Sexist expressions" in the "Rule type" field on the Rules panel.

This rule flags expressions that are considered to be sexist or stereotypical, based on gender. The error messages may offer alternatives or suggest rephrasing the sentence. For example, the rule flags the sentence, "Say hello to the little woman," and suggest the words, "wife," "partner," or "spouse," as alternatives.

Note This rule is not applicable in every language. Some Apply properties can only be found in specific language versions of Word Pro.

Word Pro: ApplySpellStandard property

{button ,AL(^H_GRAMMAR_CLASS',0)} [See list of classes](#)

(Read-write) A flag that turns the specific Grammar Check rule on or off.

Data Type

Integer

Syntax

applyspellstandardvalue = [objectreference].ApplySpellStandard

[objectreference].ApplySpellStandard = applyspellstandardvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Use this property when you are setting the corresponding rule for proofing the document in Grammar Check. Equivalent to choosing Edit - Check Grammar, clicking Options, and selecting the appropriate rule in the "Rule type" field on the Rules panel.

Note This rule is not applicable in every language. This property can only be found in specific language versions of Word Pro, for example, French.

Word Pro: ApplyStockPhrase property

{button ,AL('H_GRAMMAR_CLASS',0)} [See list of classes](#)

{button ,AL('H_APPLYSTOCKPHRASE_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) A flag that turns the Grammar Check rule on or off for checking commonly used phrases.

Data Type

[Integer](#)

Syntax

applystockphrasevalue = [objectreference].ApplyStockPhrase

[objectreference].ApplyStockPhrase = applystockphrasevalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Use this property when you are setting the corresponding rule for proofing the document in Grammar Check.

Equivalent to choosing Edit - Check Grammar, clicking Options, and selecting "Stock phrases" in the "Rule type" field on the Rules panel.

This rule flags stock phrases that may often be deleted without changing the meaning or emphasis of a sentence. Unlike clichés, these expressions are not worn-out metaphors or conventional descriptive phrases. They are introductory or parenthetical expressions often used as filler (for example, "it goes without saying that.." or "in fact").

Note This rule is not applicable in every language. Some Apply properties can only be found in specific language versions of Word Pro.

Word Pro: ApplyStyleParameters property

{button ,AL('H_GRAMMAR_CLASS',0)} [See list of classes](#)

(Read-write) A flag that turns the specific Grammar Check rule on or off.

Data Type

Integer

Syntax

applystyleparametersvalue = [objectreference].ApplyStyleParameters

[objectreference].ApplyStyleParameters = applystyleparametersvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Use this property when you are setting the corresponding rule for proofing the document in Grammar Check. Equivalent to choosing Edit - Check Grammar, clicking Options, and selecting the appropriate rule in the "Rule type" field on the Rules panel.

Note This rule is not applicable in every language. Some Apply properties can only be found in specific language versions of Word Pro.

Word Pro: ApplySubjectVerbAgreement property

{button ,AL('H_GRAMMAR_CLASS',0)} [See list of classes](#)

{button ,AL('H_APPLYSUBJECTVERBAGREEMENT_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) A flag that turns the Grammar Check rule on or off for checking agreement between subjects and verbs.

Data Type

[Integer](#)

Syntax

applysubjectverbagreementvalue = [objectreference].ApplySubjectVerbAgreement

[objectreference].ApplySubjectVerbAgreement = applysubjectverbagreementvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Use this property when you are setting the corresponding rule for proofing the document in Grammar Check.

Equivalent to choosing Edit - Check Grammar, clicking Options, and selecting "Subject-verb agreement errors" in the "Rule type" field on the Rules panel.

This rule flags errors of agreement between verbs and their subjects. The rule checks to see whether verbs contain the right number (singular or plural) and the right person (first, second, or third) to agree with their subjects. For example, the rule flags the sentence, "The index are full of errors," because the subject, "index," is singular and does not agree with the plural verb, "are."

Note This rule is not applicable in every language. Some Apply properties can only be found in specific language versions of Word Pro.

Word Pro: ApplySwedishGender property

{button ,AL(^H_GRAMMAR_CLASS',0)} [See list of classes](#)

(Read-write) A flag that turns the specific Grammar Check rule on or off.

Data Type

Integer

Syntax

applyswedishgendervalue = [objectreference].ApplySwedishGender

[objectreference].ApplySwedishGender = applyswedishgendervalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Use this property when you are setting the corresponding rule for proofing the document in Grammar Check. Equivalent to choosing Edit - Check Grammar, clicking Options, and selecting the appropriate rule in the "Rule type" field on the Rules panel.

Note This rule is not applicable in every language. Some Apply properties can only be found in specific language versions of Word Pro.

Word Pro: ApplySwedishNegation property

{button ,AL('H_GRAMMAR_CLASS',0)} [See list of classes](#)

(Read-write) A flag that turns the specific Grammar Check rule on or off.

Data Type

Integer

Syntax

applyswedishnegationvalue = [objectreference].ApplySwedishNegation

[objectreference].ApplySwedishNegation = applyswedishnegationvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Use this property when you are setting the corresponding rule for proofing the document in Grammar Check. Equivalent to choosing Edit - Check Grammar, clicking Options, and selecting the appropriate rule in the "Rule type" field on the Rules panel.

Note This rule is not applicable in every language. Some Apply properties can only be found in specific language versions of Word Pro.

Word Pro: ApplySwedishUsage property

{button ,AL('H_GRAMMAR_CLASS',0)} [See list of classes](#)

(Read-write) A flag that turns the specific Grammar Check rule on or off.

Data Type

Integer

Syntax

applyswedishusagevalue = [objectreference].ApplySwedishUsage

[objectreference].AppIswedishUsage = applyswedishusagevalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Use this property when you are setting the corresponding rule for proofing the document in Grammar Check. Equivalent to choosing Edit - Check Grammar, clicking Options, and selecting the appropriate rule in the "Rule type" field on the Rules panel.

Note This rule is not applicable in every language. Some Apply properties can only be found in specific language versions of Word Pro.

Word Pro: ApplyTrite property

{button ,AL('H_GRAMMAR_CLASS',0)} [See list of classes](#)

(Read-write) A flag that turns the specific Grammar Check rule on or off.

Data Type

Integer

Syntax

applytritevalue = [objectreference].ApplyTrite

[objectreference].ApplyTrite = applytritevalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Use this property when you are setting the corresponding rule for proofing the document in Grammar Check. Equivalent to choosing Edit - Check Grammar, clicking Options, and selecting the appropriate rule in the "Rule type" field on the Rules panel.

Note This rule is not applicable in every language. Some Apply properties can only be found in specific language versions of Word Pro.

Word Pro: ApplyTwoGender property

{button ,AL('H_GRAMMAR_CLASS',0)} [See list of classes](#)

(Read-write) A flag that turns the specific Grammar Check rule on or off.

Data Type

Integer

Syntax

applytwogendervalue = [objectreference].ApplyTwoGender

[objectreference].ApplyTwoGender = applytwogendervalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Use this property when you are setting the corresponding rule for proofing the document in Grammar Check. Equivalent to choosing Edit - Check Grammar, clicking Options, and selecting the appropriate rule in the "Rule type" field on the Rules panel.

Note This rule is not applicable in every language. Some Apply properties can only be found in specific language versions of Word Pro.every language.

Word Pro: ApplyTypicalMisspell property

{button ,AL('H_GRAMMAR_CLASS',0)} [See list of classes](#)

(Read-write) A flag that turns the specific Grammar Check rule on or off.

Data Type

Integer

Syntax

applytypicalmisspellvalue = [objectreference].ApplyTypicalMisspell

[objectreference].ApplyTypicalMisspell = applytypicalmisspellvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Use this property when you are setting the corresponding rule for proofing the document in Grammar Check. Equivalent to choosing Edit - Check Grammar, clicking Options, and selecting the appropriate rule in the "Rule type" field on the Rules panel.

Note This rule is not applicable in every language. Some Apply properties can only be found in specific language versions of Word Pro.

Word Pro: ApplyUnGrammaticalExpressions property

{button ,AL('H_GRAMMAR_CLASS',0)} [See list of classes](#)

{button ,AL('H_APPLYUNGRAMMATICALEXPRESSIONS_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) A flag that turns the Grammar Check rule on or off for checking nongrammatical words and expressions.

Data Type

[Integer](#)

Syntax

applyungrammaticalexpressionsvalue = [objectreference].ApplyUnGrammaticalExpressions

[objectreference].ApplyUnGrammaticalExpressions = applyungrammaticalexpressionsvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Use this property when you are setting the corresponding rule for proofing the document in Grammar Check.

Equivalent to choosing Edit - Check Grammar, clicking Options, and selecting "Ungrammatical expressions" in the "Rule type" field on the Rules panel.

This rule flags phrases considered ungrammatical in standard English, although they may occur in regional dialects (for example, "seeing as how" instead of "since").

Note This rule is not applicable in every language. Some Apply properties can only be found in specific language versions of Word Pro.

Word Pro: ApplyVagueQuantifiers property

{button ,AL('H_GRAMMAR_CLASS',0)} [See list of classes](#)

{button ,AL('H_APPLYVAGUEQUANTIFIERS_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) A flag that turns the Grammar Check rule for checking vague quantifiers on or off .

Data Type

[Integer](#)

Syntax

applyvaguequantifiersvalue = [objectreference].ApplyVagueQuantifiers [objectreference].ApplyVagueQuantifiers = applyvaguequantifiersvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Use this property when you are setting the corresponding rule for proofing the document in Grammar Check. Equivalent to choosing Edit - Check Grammar, clicking Options, and selecting "Vague quantifiers" in the "Rule type" field on the Rules panel.

This rule flags vague, wordy, or informal quantifiers (words or phrases that specify number or amount). For example, the expression, "lots of," can be replaced by a less colloquial term, such as "much," or "many," or a more precise indication of the amount.

Note This rule is not applicable in every language. Some Apply properties can only be found in specific language versions of Word Pro.

Word Pro: ApplyVerbGroupConsistency property

{button ,AL('H_GRAMMAR_CLASS',0)} [See list of classes](#)

{button ,AL('H_APPLYVERBGROUPCONSISTENCY_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) A flag that turns the Grammar Check rule on or off for checking verb group consistency.

Data Type

[Integer](#)

Syntax

applyverbgroupconsistencyvalue = [objectreference].ApplyVerbGroupConsistency

[objectreference].ApplyVerbGroupConsistency = applyverbgroupconsistencyvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Use this property when you are setting the corresponding rule for proofing the document in Grammar Check.

Equivalent to choosing Edit - Check Grammar, clicking Options, and selecting "Verb group consistency errors" in the "Rule type" field on the Rules panel.

This rule checks the forms of verbs. It identifies errors in the use of the present, the past, and the past participle, as well as errors in the choice of helping verbs. For example, the rule flags the sentence, "We could not have drove home that night." It notes that "drove" is the simple past form of the verb and explains that "have" should be followed by the past participle, "driven."

Note This rule is not applicable in every language. Some Apply properties can only be found in specific language versions of Word Pro.

Word Pro: ApplyWeakModifiers property

{button ,AL('H_GRAMMAR_CLASS',0)} [See list of classes](#)

{button ,AL('H_APPLYWEAKMODIFIERS_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) A flag that turns the Grammar Check rule on or off for checking weak modifiers.

Data Type

[Integer](#)

Syntax

applyweakmodifiersvalue = [objectreference].ApplyWeakModifiers

[objectreference].ApplyWeakModifiers = applyweakmodifiersvaluev

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Use this property when you are setting the corresponding rule for proofing the document in Grammar Check. Equivalent to choosing Edit - Check Grammar, clicking Options, and selecting "Weak modifiers" in the "Rule type" field on the Rules panel.

This rule flags overused or colloquial modifiers (adjectives or adverbs that limit or modify the sense of another word). Expressions, such as "funny," "pretty well," or "nice," can often be replaced by more specific expressions.

Note This rule is not applicable in every language. Some Apply properties can only be found in specific language versions of Word Pro.

Word Pro: ApplyWordChoice property

{button ,AL('H_GRAMMAR_CLASS',0)} [See list of classes](#)

(Read-write) A flag that turns the specific Grammar Check rule on or off.

Data Type

Integer

Syntax

applywordchoicevalue = [objectreference].ApplyWordChoice

[objectreference].ApplyWordChoice = applywordchoicevalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Use this property when you are setting the corresponding rule for proofing the document in Grammar Check. Equivalent to choosing Edit - Check Grammar, clicking Options, and selecting the appropriate rule in the "Rule type" field on the Rules panel.

Note This rule is not applicable in every language. Some Apply properties can only be found in specific language versions of Word Pro.

Word Pro: ApplyWordCompoundingCheck property

{button ,AL('H_GRAMMAR_CLASS',0)} [See list of classes](#)

{button ,AL('H_APPLYWORDCOMPOUNDINGCHECK_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) A flag that turns the Grammar Check rule on or off for checking word compounding errors.

Data Type

[Integer](#)

Syntax

applywordcompoundingcheckvalue = [objectreference].ApplyWordCompoundingCheck

[objectreference].ApplyWordCompoundingCheck = applywordcompoundingcheckvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Use this property when you are setting the corresponding rule for proofing the document in Grammar Check. Equivalent to choosing Edit - Check Grammar, clicking Options, and selecting "Compounding errors" in the "Rule type" field on the Rules panel.

This rule flags compounds with a missing or superfluous linking element or hyphen (for example, words such as "blue-print" that should be written as one word, "blueprint").

Note This rule is not applicable in every language. Some Apply properties can only be found in specific language versions of Word Pro.

Word Pro: ApplyWordConfusion property

{button ,AL('H_GRAMMAR_CLASS',0)} [See list of classes](#)

(Read-write) A flag that turns the specific Grammar Check rule on or off.

Data Type

Integer

Syntax

applywordconfusionvalue = [objectreference].ApplyWordConfusion

[objectreference].ApplyWordConfusion = applywordconfusionvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Use this property when you are setting the corresponding rule for proofing the document in Grammar Check. Equivalent to choosing Edit - Check Grammar, clicking Options, and selecting the appropriate rule in the "Rule type" field on the Rules panel.

Note This rule is not applicable in every language. Some Apply properties can only be found in specific language versions of Word Pro.

Word Pro: ApplyWordForm property

{button ,AL('H_GRAMMAR_CLASS',0)} [See list of classes](#)

(Read-write)

This language element has not yet been defined. You can check for a newer Help file on the Lotus web site.

Data Type

Integer.

Syntax

applywordformvalue = [objectreference].ApplyWordForm

[objectreference].ApplyWordForm = applywordformvalue

Legal values

The legal values for this property are -1 and 0 but you may use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Word Pro: ApplyWordGender property

{button ,AL(^H_GRAMMAR_CLASS',0)} [See list of classes](#)

(Read-write) A flag that turns the specific Grammar Check rule on or off.

Data Type

Integer

Syntax

applywordgendervalue = [objectreference].ApplyWordGender

[objectreference].ApplyWordGender = applywordgendervalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Use this property when you are setting the corresponding rule for proofing the document in Grammar Check. Equivalent to choosing Edit - Check Grammar, clicking Options, and selecting the appropriate rule in the "Rule type" field on the Rules panel.

Note This rule is not applicable in every language. Some Apply properties can only be found in specific language versions of Word Pro.

Word Pro: ApplyWordParts property

{button ,AL('H_GRAMMAR_CLASS',0)} [See list of classes](#)

(Read-write) A flag that turns the specific Grammar Check rule on or off.

Data Type

Integer

Syntax

applywordpartsvalue = [objectreference].ApplyWordParts

[objectreference].ApplyWordParts = applywordpartsvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Use this property when you are setting the corresponding rule for proofing the document in Grammar Check. Equivalent to choosing Edit - Check Grammar, clicking Options, and selecting the appropriate rule in the "Rule type" field on the Rules panel.

Note This rule is not applicable in every language. Some Apply properties can only be found in specific language versions of Word Pro.

Word Pro: ApplyWordyPhraseCheck property

{button ,AL('H_GRAMMAR_CLASS',0)} [See list of classes](#)

{button ,AL('H_APPLYWORDYPHRASECHECK_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) A flag that turns the Grammar Check rule on or off for checking phrases that may be wordy and/or awkward.

Data Type

[Integer](#)

Syntax

applywordyphrasecheckvalue = [objectreference].ApplyWordyPhraseCheck

[objectreference].ApplyWordyPhraseCheck = applywordyphrasecheckvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Use this property when you are setting the corresponding rule for proofing the document in Grammar Check. Equivalent to choosing Edit - Check Grammar, clicking Options, and selecting "Wordy expressions" in the "Rule type" field on the Rules panel.

This rule flags vague or wordy expressions that can be replaced by simpler, more direct expressions to make the writing clearer. For example, the phrase, "in all probability," may be replaced by the adverb, "probably."

Note This rule is not applicable in every language. Some Apply properties can only be found in specific language versions of Word Pro.

Word Pro: AppViewPrefs property

{button ,AL('H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL('H_APPVIEWPREFS_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-only) An instance of the AppViewPrefs class. The object in this property represents the view preferences for a session of Word Pro.

Data Type

[AppViewPrefs](#)

Syntax

appviewprefsvalue = [objectreference].AppViewPrefs

Legal values

Always contains an instance of the AppViewPrefs class.

Usage

Use the object in this property to manipulate the color of margins, window panes, selection borders, spelling errors, and the currently selected spelling error.

Word Pro: AreDisabledIconsGrayed property

{button ,AL(^H_ICONBARMANAGER_CLASS',0)} [See list of classes](#)

(Write-only) Indicates whether an icon is enabled or disabled. If disabled, the icon appears grayed. Before an icon can be enabled or disabled, you must first select it, using either the SelectStandardIcon or SelectCustomIcon method.

Data Type

Data type is [Integer](#)

Syntax

[objectreference].AreDisabledIconsGrayed = aredisablediconsgrayedvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

If the constant is True (-1), the icon is enabled and is not grayed. If the constant is False (0), the icon is disabled and is grayed.

Word Pro: Ascent property

{button ,AL('H_FONT_CLASS',0)} [See list of classes](#)

(Read-only) Represents the ascent characteristic of a font in points and fractions of points.

Data Type

Single

Syntax

ascentvalue = [objectreference].Ascent

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

The ascent of a font is the distance from the baseline to the top of the characters.

There are 72 points per inch.

Word Pro: AsciiCodePage property

{button ,AL('H_FILTER_CLASS',0)} [See list of classes](#)

{button ,AL('H_ASCII_CODEPAGE_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write)

Data Type

Integer.

Syntax

[objectreference].AsciiCodePage = asciicodepagevalue

asciicodepagevalue = [objectreference].AsciiCodePage

Legal values**Usage**

Word Pro: AtBeginningOfLine property

{button ,AL('H_CLICKHERE_CLASS;H_TEXTMARKER_CLASS;H_TEXT_CLASS',0)} [See list of classes](#)

{button ,AL('H_ATBEGINNINGOFLINE_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-only)

Data Type

[Integer](#)

Syntax

atbeginningoflinevalue = [objectreference].AtBeginningOfLine

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Word Pro: AtBeginningOfObject property

{button ,AL(^H_CLICKHERE_CLASS;H_TEXTMARKER_CLASS;H_TEXT_CLASS',0)} [See list of classes](#)

(Read-only)

Data Type

Integer

Syntax

atbeginningofobjectvalue = [objectreference].AtBeginningOfObject

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Word Pro: AtBeginningOfParagraph property

{button ,AL(^H_CLICKHERE_CLASS;H_TEXTMARKER_CLASS;H_TEXT_CLASS',0)} [See list of classes](#)

(Read-only)

Data Type

Integer

Syntax

atbeginningofparagraphvalue = [objectreference].AtBeginningOfParagraph

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Word Pro: AtBeginningOfStream property

{button ,AL(^H_GRAPHIC_CLASS',0)} [See list of classes](#)

(Read-only)

Data Type

Integer

Syntax

atbeginningofstreamvalue = [objectreference].AtBeginningOfStream

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Word Pro: AtBeginningOfWord property

{button ,AL(^H_CLICKHERE_CLASS;H_TEXTMARKER_CLASS;H_TEXT_CLASS',0)} [See list of classes](#)

(Read-only)

Data Type

Integer

Syntax

atbeginningofwordvalue = [objectreference].AtBeginningOfWord

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Word Pro: AtBeginning property

{button ,AL(^H_CLICKHERE_CLASS;H_TEXTMARKER_CLASS;H_TEXT_CLASS',0)} [See list of classes](#)

{button ,AL(^H_ATBEGINNING_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-only) Indicates whether or not the insertion point is at the beginning of a text stream.

Data Type

[Integer](#)

Syntax

atbeginningvalue = [objectreference].AtBeginning

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Word Pro: AtEndOfLine property

{button ,AL('H_CLICKHERE_CLASS;H_TEXT_CLASS',0)} [See list of classes](#)

{button ,AL('H_ATENDOFLINE_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-only)

Data Type

[Integer](#)

Syntax

atendoflinevalue = [objectreference].AtEndOfLine

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Word Pro: AtEndOfObject property

{button ,AL(^H_CLICKHERE_CLASS;H_TEXTMARKER_CLASS;H_TEXT_CLASS',0)} [See list of classes](#)

(Read-only)

Data Type

Integer

Syntax

atendofobjectvalue = [objectreference].AtEndOfObject

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Word Pro: AtEndOfParagraph property

{button ,AL(^H_CLICKHERE_CLASS;H_TEXTMARKER_CLASS;H_TEXT_CLASS',0)} [See list of classes](#)

(Read-only)

Data Type

Integer

Syntax

atendofparagraphvalue = [objectreference].AtEndOfParagraph

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Word Pro: AtEndOfStream property

{button ,AL(^H_GRAPHIC_CLASS',0)} [See list of classes](#)

(Read-only)

Data Type

[Integer](#)

Syntax

atendofstreamvalue = [objectreference].AtEndOfStream()

Legal values

True False

Usage

Word Pro: AtEndOfWord property

{button ,AL(^H_CLICKHERE_CLASS;H_TEXTMARKER_CLASS;H_TEXT_CLASS',0)} [See list of classes](#)

(Read-only)

Data Type

Integer

Syntax

atendofwordvalue = [objectreference].AtEndOfWord

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Word Pro: AtEnd property

{button ,AL('H_CLICKHERE_CLASS;H_TEXTMARKER_CLASS;H_TEXT_CLASS',0)} [See list of classes](#)

{button ,AL('H_ATEND_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-only)

Data Type

[Integer](#)

Syntax

atendvalue = [objectreference].AtEnd

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Word Pro: Attributes property

{button ,AL('H_CLICKHERE_CLASS;H_TEXTMARKER_CLASS;H_TEXT_CLASS',0)} [See list of classes](#)

{button ,AL('H_ATTRIBUTES_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-only)

Data Type

[Attributes](#)

Syntax

attributesvalue = [objectreference].Attributes

Legal values

Always contains an instance of the Attributes class.

Usage

Word Pro: AttrStyleName property

{button ,AL('H_CHARACTERSTYLE_CLASS;H_PARAGRAPHSTYLE_CLASS',0)} [See list of classes](#)

{button ,AL('H_ATTRSTYLENAME_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) The name of the attribute style from which you get the attribute information for a font. Word Pro uses the same style name for attributes as for a character, or group of characters. Bold, Italic, Underline, and so on, are part of this style.

Data Type

String

Syntax

attrstylevalue = [objectreference].AttrStyleName

[objectreference].AttrStyleName = attrstylevalue

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Word Pro: AuthorName property

{button ,AL('H_DOCINFO_CLASS;H_VERSION_CLASS',0)} [See list of classes](#)

{button ,AL('H_AUTHORNAME_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-only)

[DocInfo]

Indicates the name of the assigned editor who created the document.

[Version]

Data Type

String

Syntax

authornamevalue = [objectreference].AuthorName

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

[DocInfo]

Equivalent to the "User name" box on the Personal panel of the Word Pro Preferences dialog box.

Word Pro: AutoBackup property

{button ,AL('H_USERINTERFACEPREFS_CLASS',0)} [See list of classes](#)

{button ,AL('H_AUTOBACKUP_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Indicates if automatic backup of documents is enabled.

Data Type

[Integer\(bool\)](#)

Syntax

autobackupvalue = [objectreference].AutoBackup

[objectreference].AutoBackup = autobackupvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values. Default is False (0).

Usage

Equivalent to "File saving options" on the General panel of the Word Pro Preferences dialog box. If this property is set to False, Word Pro does not automatically back up documents.

Word Pro: AutoHyphenate property

{button ,AL('H_HYPHENATIONOPTIONS_CLASS',0)} [See list of classes](#)

{button ,AL('H_AUTOHYPHENATE_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Determines whether automatic hyphenation is enabled.

Data Type

[Integer](#)

Syntax

autohyphenatevalue = [objectreference].AutoHyphenate

[objectreference].AutoHyphenate = autohyphenatevalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

This property is equivalent to the "Auto hyphenation" option in the Document Properties dialog box.

Word Pro: AutomaticLink property

{button ,AL('H_GRAPHIC_CLASS;H_OLEOBJECT_CLASS',0)} [See list of classes](#)

(Read-write)

Note This property is not supported for the class OLEObject within OS/2.

Data Type

Integer

Syntax

automaticlinkvalue = [objectreference].AutomaticLink

[objectreference].AutomaticLink = automaticlinkvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Word Pro: AutoRunMacro property

{button ,AL('H_TEXTDOCUMENT_CLASS',0)} [See list of classes](#)

(Read-only)

Data Type

[AutoRunMacro](#)

Syntax

autorunmacrovalue = [objectreference].AutoRunMacro

Legal values

Always contains an instance of the AutoRunMacro class.

Usage

Word Pro: AutoSaveMinutes property

{button ,AL(`H_USERINTERFACEPREFS_CLASS',0)} [See list of classes](#)

{button ,AL(`H_AUTOSAVEMINUTES_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Stores the automatic time save interval in minutes.

Data Type

[Integer](#)

Syntax

autosaveminutesvalue = [objectreference].AutoSaveMinutes

[objectreference].AutoSaveMinutes = autosaveminutesvalue

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Word Pro uses this property only if the AutoSave property is set to True. Equivalent to the "minutes" box on the General panel of the Word Pro Preferences dialog box. Default for AutoSaveMinutes is 10.

{button ,AL(`H_AUTOSAVE_PROPERTY_MEMDEF',0)} [See related topics](#)

Word Pro: AutoSave property

{button ,AL(`H_USERINTERFACEPREFS_CLASS',0)} [See list of classes](#)

{button ,AL(`H_AUTOSAVE_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Indicates if automatic time save is enabled.

Data Type

[Integer\(bool\)](#)

Syntax

autosavevalue = [objectreference].AutoSave

[objectreference].AutoSave = autosavevalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values. Default is True (-1).

Usage

Equivalent to "File saving options" on the General panel of the Word Pro Preferences dialog box. If the this property is set to False, Word Pro does not automatically time save documents.

{button ,AL(`H_AUTOSAVEMINUTES_PROPERTY_MEMDEF',0)} [See related topics](#)

Word Pro: AutoVersion property

{button ,AL('H_DOCCONTROL_CLASS',0)} [See list of classes](#)

{button ,AL('H_AUTOVERSION_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Determines when a version is automatically created. This property is usually set to occur when the file opens.

Data Type

Data type is [Variant](#), which allows the value of this property to be one of the constants listed below or its numeric equivalent (in parentheses).

Syntax

autoversionvalue = [objectreference].AutoVersion

[objectreference].AutoVersion = autoversionvalue

Legal values

<u>Value</u>	<u>Effect</u>
\$LwpAutoVersionEveryday (29)	Creates a new version and reuses that version until the file is opened on the following day. If you leave a file open until the next day, Word Pro does not create a new version until you close and reopen the file.
\$LwpAutoVersionEveryweek (30)	Creates a new version and reuses that version until the file is opened after the following Monday. If you leave a file open until Monday, Word Pro does not create a new version until you close and reopen the file. In Word Pro, Monday is the first day of the week.
\$LwpAutoVersionNone (27)	Turns off the AutoVersion property.
\$LwpAutoVersionOnfileopen (28)	Creates a new version every time someone opens the file.
\$LwpAutoVersionOnfilesave (31)	Creates a new version every time someone saves the file.

Usage

Equivalent to choosing File - Versions, selecting "Auto versioning," and specifying when auto versioning should occur (every day, every week, and so on).

Word Pro: BackColorIndex property

{button ,AL(^H_BACKGROUND_CLASS;H_FONT_CLASS;H_BORDER_CLASS',0)} [See list of classes](#)

(Read-write)

Data Type

Integer.

Syntax

[objectreference].BackColorIndex = backcolorindexvalue

backcolorindexvalue = [objectreference].BackColorIndex

Legal values**Usage**

Word Pro: BackColor property

{button ,AL(`H_BACKGROUND_CLASS;H_FONT_CLASS',0)} [See list of classes](#)

{button ,AL(`H_BACKCOLOR_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-only)

Data Type

[Color](#)

Syntax

backcolorvalue = [objectreference].BackColor

Legal values

Always contains an instance of the Color class.

Usage

{button ,AL(`H_COLOR_PROPERTY_MEMDEF;H_PATTERN_PROPERTY_MEMDEF',0)} [See related topics](#)

Word Pro: BackgroundPrintingOn property

{button ,AL('H_PREFERENCES_CLASS',0)} [See list of classes](#)

{button ,AL('H_BACKGROUNDPRINTINGON_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write)

Data Type

[Integer](#)

Syntax

backgroundprintingonvalue = [objectreference].BackgroundPrintingOn

[objectreference].BackgroundPrintingOn = backgroundprintingonvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Word Pro: BackgroundSpellingOn property

{button ,AL('H_PREFERENCES_CLASS',0)} [See list of classes](#)

{button ,AL('H_BACKGROUNDSPELLINGON_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write)

Data Type

Integer.

Syntax

[objectreference].BackgroundSpellingOn = backgroundspellingonvalue

backgroundspellingonvalue = [objectreference].BackgroundSpellingOn

Legal values

Usage

Word Pro: Background property

{button ,AL(^H_AMIKAKE_CLASS;H_LAYOUT_CLASS;H_CELLGROUPLAYOUT_CLASS;H_CELLLAYOUT_CLASS;H_CONNECTEDLAYOUT_CLASS;H_COLUMNGROUPLAYOUT_CLASS;H_FOOTERLAYOUT_CLASS;H_FRAMELAYOUT_CLASS;H_FRAMEGROUPLAYOUT_CLASS;H_DROPPLAYOUT_CLASS;H_GROUPLAYOUT_CLASS;H_HEADERLAYOUT_CLASS;H_NOTELAYOUT_CLASS;H_PAGELAYOUT_CLASS;H_ROWGROUPLAYOUT_CLASS;H_ROWLAYOUT_CLASS;H_RUBYLAYOUT_CLASS;H_SUPERTABLEGROUPLAYOUT_CLASS;H_SUPERTABLELAYOUT_CLASS;H_TABLELAYOUT_CLASS;H_ENDNOTELAYOUT_CLASS;H_FOOTNOTELAYOUT_CLASS;H_TABLEHEADINGLAYOUT_CLASS;H_TOCSUPERTABLELAYOUT_CLASS;H_PARAGRAPHBACKGROUND_CLASS;H_TABLEFILL_CLASS',0)} [See list of classes](#)

{button ,AL(^H_BACKGROUND_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-only) Indicates the current background settings for a specific layout object.

[TableFill]

(Read-write) Allows you to set the foreground color, background color and pattern of a specific table object.

[Layout]

Allows you to set the foreground color, background color and pattern of a specific layout object.

Data Type

Background

Syntax

backgroundvalue = [objectreference].Background

[objectreference].Background = backgroundvalue

Legal values

Always contains an instance of the Background class.

Usage

Amikake - If you are using an English language version of Word Pro, this property as a member of Amikake is not available.

TableFill - Equivalent to choosing Table - Table Properties, selecting Table cell in the "Properties for" box, clicking Options on the Table Cell Lines & Colors panel, and selecting the desired options from the "Fill," "Background color," "Pattern," and "Pattern color" boxes.

Layout - Equivalent to opening the layout object InfoBox and selecting the desired options from the "Background color," "Pattern," and "Pattern color" boxes in the Lines & Colors panel.

Word Pro: BackupPaths property

{button ,AL(^H_USERINTERFACEPREFS_CLASS,0)} [See list of classes](#)

(Read-only) Stores multiple backup paths (drive and directory) for Word Pro documents.

Data Type

[StringCollection](#)

Syntax

backuppathsvalue = [objectreference].BackupPaths

Legal values

Always contains an instance of the StringCollection class.

Usage

Equivalent to the "Backups" field on the Locations panel of the Word Pro Preferences dialog box. Although you can enter multiple paths in this field, Word Pro only uses the first one listed. The first path listed corresponds to the BackupPath property.

Word Pro: BackupPath property

{button ,AL(^H_USERINTERFACEPREFS_CLASS',0)} [See list of classes](#)

{button ,AL(^H_BACKUPPATH_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Stores the default backup path (drive and directory) for Word Pro documents.

Data Type

[String](#)

Syntax

backuppathvalue = [objectreference].BackupPath

[objectreference].BackupPath = backuppathvalue

Legal values

Usage

Equivalent to the "Backups" field on the Locations panel of the Word Pro Preferences dialog box.

Word Pro: Bags property

{button ,AL('H_FOUNDRY_CLASS',0)} [See list of classes](#)

(Read-only) An object created from the BagCollection class. This object provides access to Bag objects.

Data Type

[BagCollection](#)

Syntax

bagsvalue = [objectreference].Bags

Legal values

Always contains an instance of the BagCollection class.

Usage

When accessed through the Foundry property on a Division object, the collection object in this property provides access to all the Bag objects contained in that Division object. When accessed through the AppFoundry property on the WPAApplication object, the collection object in this property provides access to all the Bag objects contained in the Word Pro Clipboard.

When accessed through the TempFoundry property on the WPAApplication object, the collection object in this property provides access to all the Bag objects placed in TempFoundry by WordPro or a script.

When accessed through the Foundry property on the WPAApplication object, the collection object in this property provides access to all the Bag objects contained in the currently active Division object.

This property is not used in the Foundry property on the TextDocument object.

For more information about collection classes, see [Overview: Word Pro LotusScript Collection Classes](#).

Word Pro: BaseLineOffset property

{button ,AL(^H_ATTRIBUTES_CLASS;H_LAYOUT_CLASS;H_CELLGROUPLAYOUT_CLASS;H_CELLAYOUT_CLASS;H_CONNECTEDLAYOUT_CLASS;H_COLUMNGROUPLAYOUT_CLASS;H_FOOTERLAYOUT_CLASS;H_FRAMELAYOUT_CLASS;H_FRAMEGROUPLAYOUT_CLASS;H_DROPPLAYOUT_CLASS;H_GROUPLAYOUT_CLASS;H_HEADERLAYOUT_CLASS;H_NOTELAYOUT_CLASS;H_PAGELAYOUT_CLASS;H_ROWGROUPLAYOUT_CLASS;H_ROWLAYOUT_CLASS;H_RUBYLAYOUT_CLASS;H_SUPERTABLEGROUPLAYOUT_CLASS;H_SUPERTABLELAYOUT_CLASS;H_TABLELAYOUT_CLASS;H_ENDNOTELAYOUT_CLASS;H_FOOTNOTELAYOUT_CLASS;H_TABLEHEADINGLAYOUT_CLASS;H_TOCSUPERTABLELAYOUT_CLASS',0)} [See list of classes](#)

(Read-write) Allows you to set or retrieve a layout object's position, relative to the text baseline.

Data Type

[Long](#)

Syntax

baselineoffsetvalue = [objectreference].BaseLineOffset

[objectreference].BaseLineOffset = baselineoffsetvalue

Legal values

Data type is Long but the unit of measurement used for this property is [Twips](#). There are 1440 Twips per inch.

Usage

[FrameLayout]

Use this property when the frame layout object's placement is set to "In Text."

Word Pro: BaseTables property

{button ,AL('H_FOUNDRY_CLASS',0)} [See list of classes](#)

(Read-only)

Data Type

TableCollection

Syntax

basetablesvalue = [objectreference].BaseTables

Legal values

A member of the TableCollection Class.

Usage

Word Pro: BaseTable property

{button ,AL('H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL('H_BASetable_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-only)

An instance of one of the following classes:

FootnoteTable

Glossary

ParallelColumns

TableHeading

Table

This is a current context property. The content of this property is determined by context of the Word Pro focus.

Data Type

[BaseTable](#)

Syntax

basetablevalue = [objectreference].BaseTable

Legal values

Data type is BaseTable which allows this property to contain any object created from one of BaseTable's derived classes, including FootnoteTable, Glossary, ParallelColumns, TableHeading, and Table.

Usage

In Word Pro, tables take many different forms and have many different uses. In addition to the standard Table object, there are Glossary objects, TableHeading objects, FootnoteTable objects, and ParallelColumns objects. Each of these objects is unique and serves a different purpose. But the classes for these objects are all derived from the same BaseTable class. By using that BaseTable class as the data type for the BaseTable property, Word Pro allows the BaseTable property to contain any object that is created from one of BaseTable's derived classes.

Whatever object is stored in BaseTable is determined by the context of Word Pro's focus. If you have a document that has a table, a footnote, a glossary, and a table heading, Word Pro gives you access to all of those objects through the BaseTable property. However, only one of those objects can be found in BaseTable at any given time. If the focus (usually your cursor) is on the Table object, that Table object is contained in the BaseTable property. Similarly, if the focus is on a FootnoteTable object, then the BaseTable property contains that FootnoteTable object. When your focus is on a page and not in a table, this property contains the currently active ParallelColumns object.

Word Pro: BaseURL property

{button ,AL('H_TEXTDOCUMENT_CLASS',0)} [See list of classes](#)

(Read-write)

Data Type

String.

Syntax

[objectreference].BaseURL = baseurlvalue

baseurlvalue =- [objectreference].BaseURL

Legal values

Usage

Word Pro: BetweenLines property

{button ,AL('H_PARAGRAPHBORDER_CLASS','0')} [See list of classes](#)

{button ,AL('H_BETWEENLINES_PROPERTY_EXSCRIPT','1')} [See example](#)

(Read-only)

This language element has not yet been defined. You can check for a newer Help file on the Lotus web site.

Data Type

BetweenLines

Syntax

betweenlinesvalue = [objectreference].BetweenLines

Legal values

Always contains an instance of the BetweenLines class.

Usage

Word Pro: BinNames property

{button ,AL(^H_PRINTMANAGER_CLASS',0)} [See list of classes](#)

(Read-only)

Data Type

[StringCollection](#)

Syntax

binnamesvalue = [objectreference].BinNames

Legal values

Always contains an instance of the StringCollection class.

Usage

Word Pro: BinName property

```
{button ,AL(^H_LAYOUT_CLASS;H_CELLGROUPLAYOUT_CLASS;H_CELLLAYOUT_CLASS;H_CONNECTEDLAY  
OUT_CLASS;H_COLUMNGROUPLAYOUT_CLASS;H_FOOTERLAYOUT_CLASS;H_FRAMELAYOUT_CLASS;H  
_FRAMEGROUPLAYOUT_CLASS;H_DROP_CAPLAYOUT_CLASS;H_GROUPLAYOUT_CLASS;H_HEADERLAY  
OUT_CLASS;H_NOTELAYOUT_CLASS;H_PAGELAYOUT_CLASS;H_ROWGROUPLAYOUT_CLASS;H_ROWLA  
YOUT_CLASS;H_RUBYLAYOUT_CLASS;H_SUPERTABLEGROUPLAYOUT_CLASS;H_SUPERTABLELAYOUT  
_CLASS;H_TABLELAYOUT_CLASS;H_ENDNOTELAYOUT_CLASS;H_FOOTNOTELAYOUT_CLASS;H_TABLE  
HEADINGLAYOUT_CLASS;H_TOCSUPERTABLELAYOUT_CLASS',0)} See list of classes
```

(Read-write) The name of the bin from which the paper is taken when printing a page layout.

Data Type

String

Syntax

binnamevalue = [objectreference].BinName

[objectreference].BinName = binnamevalue

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

The BinName property is used only for page layouts. Equivalent to the "Printer bin" setting on the Size & Margins panel of the Page layout InfoBox.

If you set this property to an invalid bin name, Word Pro uses the current printer's default bin selection. You can check for valid BinName values by accessing the [BinNames](#) collection.

Word Pro: BlockPaint property

{button ,AL(^H_DOCWINDOW_CLASS',0)} [See list of classes](#)

(Write-only)

Data Type

[Integer](#)

Syntax

[objectreference].BlockPaint = blockpaintvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Word Pro: Blue property

{button ,AL('H_COLOR_CLASS',0)} [See list of classes](#)

{button ,AL('H_BLUE_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) The blue component of a color.

Data Type

Integer

Syntax

bluevalue = [objectreference].Blue

[objectreference].Blue = bluevalue

Legal values

The value of this property can range from 0 - 255.

Usage

Use this property to access the current level of blue in a specific object's color. For example, if you want to change the blue value of a frame's background color, you can use the following statement:

```
.Frame.Layout.Background.Color.Blue = 128
```

Word Pro: BodyOnly property

{button ,AL('H_INDENT_CLASS;H_RELATIVEINDENT_CLASS',0)} [See list of classes](#)

{button ,AL('H_BODYONLY_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Specifies whether an indent should be ignored when not in the body of the document.

Data Type

Integer

Syntax

bodyonlyvalue = [objectreference].BodyOnly

[objectreference].BodyOnly = bodyonlyvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Equivalent to the "Ignore indent when not in main body of document" box in the Indent Options dialog box.

Word Pro: Bold property

{button ,AL('H_FONT_CLASS',0)} [See list of classes](#)

{button ,AL('H_BOLD_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write)

Data Type

Integer

Syntax

boldvalue = [objectreference].Bold

[objectreference].Bold = boldvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Word Pro: BookletPrinting property

{button ,AL('H_PRINTSETTINGS_CLASS',0)} [See list of classes](#)

{button ,AL('H_BOOKLETPRINTING_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Indicates whether or not booklet printing is enabled for the current document.

Data Type

[Integer](#)

Syntax

bookletprintingvalue = [objectreference].BookletPrinting

[objectreference].BookletPrinting = bookletprintingvalue

Legal values

Data type is Integer. The legal values for this property are -1 and 0 but you may use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

When setting this property, the PageOrder property must be set "front to back." . Equivalent to the "As booklet" option in the Word Pro Print Options dialog box. For more information on booklet printing, see the Word Pro Help topics which relate to printing.

Word Pro: BookmarkManager property

{button ,AL('H_DIVISION_CLASS;H_FOUNDRY_CLASS;H_TEXTDOCUMENT_CLASS',0)} [See list of classes](#)

(Read-only)

Provides access to a BookmarkManager object. You should access this property from the Division or TextDocument object which contains the bookmarks you want to access.

Data Type

[BookmarkManager](#)

Syntax

SET *bookmarkmanagervariable* = [*objectreference*].BookmarkManager

Legal values

Always contains an instance of the BookmarkManager class.

Usage

When called from a Division object:

Provides access to the BookmarkManager object which manages the bookmarks for a division.

```
.Division.BookmarkManager
```

Calls the BookmarkManager from the currently active division.

When called from the Foundry object on a Division:

In addition to the bookmarks in the division, this BookmarkManager object manages any bookmarks which are present in the Clipboard (AppFoundry) as well as Word Pro's internal clipboard (TempFoundry). This is the only way to access bookmarks in a clipboard area.

```
.Division.Foundry.BookmarkManager
```

Calls the BookmarkManager from the currently active division.

Word Pro: BookmarksByMarkerName property

{button ,AL('H_BOOKMARKMANAGER_CLASS',0)} [See list of classes](#)

(Read-only) An object created from the BookmarkCollection class. This object provides access to Bookmark objects in a document or division.

Data Type

[BookmarkCollection](#)

Syntax

bookmarksbymarkernamevalue = [objectreference].BookmarksByMarkerName

Legal values

Always contains an instance of the BookmarkCollection class.

Usage

The Bookmark objects stored in this collection are indexed by their internal (hexadecimal) names which cannot be changed.

Word Pro: Bookmarks property

{button ,AL(^H_BOOKMARKMANAGER_CLASS',0)} [See list of classes](#)

(Read-only) An object created from the BookmarkCollection class. This object provides access to Bookmark objects in a document or division.

Data Type

[BookmarkCollection](#)

Syntax

bookmarksvalue = [objectreference].Bookmarks

Legal values

Always contains an instance of the BookmarkCollection class.

Usage

The Bookmark objects stored in this collection are shown in Word Pro in the Bookmarks dialog box. They are indexed by their user-defined name.

In order to access all bookmarks in a document, you must iterate the collection of divisions within the document to access the Bookmarks property of each division's BookmarkManager object.

Word Pro: BorderLines property

{button ,AL(^H_CHARACTERBORDER_CLASS;H_FOOTNOTESEPOPT_CLASS;H_FOOTNOTECONTSEP_CLASS;H_FOOTNOTESEPARATOR_CLASS;H_LAYOUT_CLASS;H_CELLGROUPLAYOUT_CLASS;H_CELLLAYOUT_CLASS;H_CONNECTEDLAYOUT_CLASS;H_COLUMNGROUPLAYOUT_CLASS;H_FOOTERLAYOUT_CLASS;H_FRAMELAYOUT_CLASS;H_FRAMEGROUPLAYOUT_CLASS;H_DROPCAPLAYOUT_CLASS;H_GROUPLAYOUT_CLASS;H_HEADERLAYOUT_CLASS;H_NOTELAYOUT_CLASS;H_PAGELAYOUT_CLASS;H_ROWGROUPLAYOUT_CLASS;H_ROWLAYOUT_CLASS;H_RUBYLAYOUT_CLASS;H_SUPERTABLEGROUPLAYOUT_CLASS;H_SUPERTABLELAYOUT_CLASS;H_TABLELAYOUT_CLASS;H_ENDNOTELAYOUT_CLASS;H_FOOTNOTE_LAYOUT_CLASS;H_TABLEHEADINGLAYOUT_CLASS;H_TOCSUPERTABLELAYOUT_CLASS;H_PARAGRAPHBORDER_CLASS;H_TABLELINE_CLASS',0)} [See list of classes](#)

(Read-only) Sets or returns the BorderLines object for a specific object.

Data Type

[BorderLines](#)

Syntax

borderlinesvalue = [objectreference].BorderLines

Legal values

Always contains an instance of the BorderLines class.

Usage

Use this property to access the type, color, or width of the border lines in a specific object.

Word Pro: BorderOffset property

```
{button ,AL(`H_LAYOUT_CLASS;H_CELLGROUPLAYOUT_CLASS;H_CELLLAYOUT_CLASS;H_CONNECTEDLAY  
OUT_CLASS;H_COLUMNGROUPLAYOUT_CLASS;H_FOOTERLAYOUT_CLASS;H_FRAMELAYOUT_CLASS;H  
_FRAMEGROUPLAYOUT_CLASS;H_DROP_CAP_LAYOUT_CLASS;H_GROUPLAYOUT_CLASS;H_HEADERLAY  
OUT_CLASS;H_NOTELAYOUT_CLASS;H_PAGELAYOUT_CLASS;H_ROWGROUPLAYOUT_CLASS;H_ROWLA  
YOUT_CLASS;H_RUBY_LAYOUT_CLASS;H_SUPERTABLEGROUPLAYOUT_CLASS;H_SUPERTABLELAYOUT  
_CLASS;H_TABLELAYOUT_CLASS;H_ENDNOTELAYOUT_CLASS;H_FOOTNOTELAYOUT_CLASS;H_TABLE  
HEADINGLAYOUT_CLASS;H_TOCSUPERTABLELAYOUT_CLASS',0)} See list of classes
```

```
{button ,AL(`H_BORDEROFFSET_PROPERTY_EXSCRIPT',1)} See example
```

(Read-write) Allows you to change the distance between border lines and a layout object.

Data Type

Long

Syntax

borderoffsetvalue = [objectreference].BorderOffset

[objectreference].BorderOffset = borderoffsetvalue

Legal values

Data type is Long but the unit of measurement used for this property is Twips. There are 1440 Twips per inch.

Usage

This is the distance between border lines and the location specified in the "Line placement" box in the Lines panel of the InfoBox.

Set this property in conjunction with the LineLocation property. If you specify a custom location, that value is stored here. Otherwise, Word Pro calculates this value based on your margins.

For example, if you have lines specified for the middle of the margins and margins of one inch, the value of this property is 720 twips (1/2 inch). Later, if you change the margins to 1/2 inch, WordPro automatically recalculates the value of this property, coming up with 360 twips (1/4 inch) so that the line location is still in the middle.

```
{button ,AL(`H_LINELOCATION_PROPERTY_MEMDEF',0)} See related topics
```

Word Pro: BorderStyleName property

{button ,AL('H_PARAGRAPHSTYLE_CLASS',0)} [See list of classes](#)

(Read-write)

Data Type

String

Syntax

borderstylevalue = [objectreference].BorderStyleName

[objectreference].BorderStyleName = borderstylevalue

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Word Pro: BottomBorder property

{button ,AL(^H_BORDERLINES_CLASS;H_GUTTER_CLASS;H_BETWEENLINES_CLASS',0)} [See list of classes](#)

(Read-only) Allows you to access an object's bottom border object.

Data Type

[Border](#)

Syntax

bottombordervalue = [objectreference].BottomBorder

Legal values

Always contains an instance of the Border class.

Usage

You can also use the AllBorders property to simultaneously access an object's BottomBorder, LeftBorder, RightBorder, and TopBorder objects.

Word Pro: BottomExternalMargin property

```
{button ,AL('H_LAYOUT_CLASS;H_CELLGROUPLAYOUT_CLASS;H_CELLLAYOUT_CLASS;H_CONNECTEDLAY  
OUT_CLASS;H_COLUMNGROUPLAYOUT_CLASS;H_FOOTERLAYOUT_CLASS;H_FRAMELAYOUT_CLASS;H  
_FRAMEGROUPLAYOUT_CLASS;H_DROP_CAPLAYOUT_CLASS;H_GROUPLAYOUT_CLASS;H_HEADERLAY  
OUT_CLASS;H_NOTELAYOUT_CLASS;H_PAGELAYOUT_CLASS;H_ROWGROUPLAYOUT_CLASS;H_ROWLA  
YOUT_CLASS;H_RUBYLAYOUT_CLASS;H_SUPERTABLEGROUPLAYOUT_CLASS;H_SUPERTABLELAYOUT  
_CLASS;H_TABLELAYOUT_CLASS;H_ENDNOTELAYOUT_CLASS;H_FOOTNOTELAYOUT_CLASS;H_TABLE  
HEADINGLAYOUT_CLASS;H_TOCSUPERTABLELAYOUT_CLASS',0)} See list of classes
```

```
{button ,AL('H_BOTTOMEXTERNALMARGIN_PROPERTY_EXSCRIPT',1)} See example
```

(Read-write) Allows you to set the amount of margin space located beneath a layout object.

Data Type

Long

Syntax

bottomexternalmarginvalue = [objectreference].BottomExternalMargin

[objectreference].BottomExternalMargin = bottomexternalmarginvalue

Legal values

Data type is Long but the unit of measurement used for this property is Twips. There are 1440 Twips per inch.

Usage

[FrameLayout]

This property cannot be set individually for frame layout objects within Word Pro. It is combined with all external margin values in the "Padding around border" setting on the Size & Margins panel of the InfoBox.

Word Pro: BreaksStyleName property

{button ,AL('H_PARAGRAPHSTYLE_CLASS',0)} [See list of classes](#)

(Read-write)

Data Type

String

Syntax

breaksstylevalue = [objectreference].BreaksStyleName

[objectreference].BreaksStyleName = breaksstylevalue

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Word Pro: Breaks property

{button ,AL(^H_CLICKHERE_CLASS;H_TEXTMARKER_CLASS;H_PARAGRAPHSTYLE_CLASS;H_TEXT_CLASS',
0)} [See list of classes](#)

{button ,AL(^H_BREAKS_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-only)

Data Type

[Breaks](#)

Syntax

breaksvalue = [objectreference].Breaks

Legal values

Always contains an instance of the Breaks class.

Usage

Word Pro: BulletFonts property

{button ,AL(^H_USERINTERFACEPREFS_CLASS',0)} [See list of classes](#)

(Read-write) A list of the names of decorative fonts, separated by commas, and used to identify possible bullets during Format Check.

Data Type

String

Syntax

bulletfontvalue = [objectreference].BulletFonts

[objectreference].BulletFonts = bulletfontvalue

Legal values

Usage

When you run Format Check on a document and Word Pro finds a single character surrounded by white space, Word Pro tests to see if the character is a possible bullet. If the character is in this list, Word Pro assumes that it could be a bullet and runs further tests to confirm. You can use the SetArrayProp method to add or delete fonts from the list, or you can use the GetArrayProp method to read the fonts in the list.

Word Pro: BulletStyleName property

{button ,AL('H_PARAGRAPHSTYLE_CLASS',0)} [See list of classes](#)

(Read-write)

Data Type

String

Syntax

bulletstylevalue = [objectreference].BulletStyleName

[objectreference].BulletStyleName = bulletstylevalue

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Word Pro: Bullet property

{button ,AL(^H_CLICKHERE_CLASS;H_TEXTMARKER_CLASS;H_PARAGRAPHSTYLE_CLASS;H_TEXT_CLASS',
0)} [See list of classes](#)

(Read-only)

Data Type

[Bullet](#)

Syntax

bulletvalue = [objectreference].Bullet

Legal values

Always contains an instance of the Bullet class.

Usage

Word Pro: CanCreatePreviewBitmap property

{button ,AL(^H_DIVISION_CLASS;H_TEXTDOCUMENT_CLASS',0)} [See list of classes](#)

(Read-only)

Data Type

Integer.

Syntax

cancreatepreviewbitmapvalue = [objectreference].CanCreatePreviewBitmap

Legal values

Usage

Word Pro: CanEditProperty property

{button ,AL(^H_MAILROUTING_CLASS',0)} [See list of classes](#)

(Read-only) An internal flag that indicates if the route can be edited.

Data Type

[Integer](#)

Syntax

caneditpropertyvalue = [objectreference].CanEditProperty

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

The document must be in the middle of the route at a preordained stop for this property to function.

Word Pro: CanEmbed property

{button ,AL('H_CONTENT_CLASS;H_GRAPHICOLEBJECT_CLASS;H_GRAPHIC_CLASS;H_OLEOBJECT_CLASSES;H_SUPERTABLE_CLASS;H_BASETABLE_CLASS;H_FOOTNOTETABLE_CLASS;H_PARALLELCOLUMNS_CLASS;H_GLOSSARY_CLASS;H_TABLEHEADING_CLASS;H_TABLE_CLASS;H_TEXT_CLASS';0)} [See list of classes](#)

(Read-only) Indicates whether or not another object can be embedded into the current object.

Data Type

[Integer](#)

Syntax

canembedvalue = [objectreference].CanEmbed

Legal values

The legal values for this property are -1 and 0. If you prefer, you may use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Indicates whether or not another object can be embedded into the current object. For example, a Graphic object contains a False value in this property, since no other objects can be inserted into it. A Text object contains a True value in this property, since other objects can be inserted into it.

Word Pro: CanWePrint property

{button ,AL(^H_PRINTMANAGER_CLASS',0)} [See list of classes](#)

(Read-only)

Data Type

Integer

Syntax

canweprintvalue = [objectreference].CanWePrint

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Word Pro: Caption property

{button ,AL('H_WINDOW_CLASS;H_APPLICATIONWINDOW_CLASS;H_STATUSBAR_CLASS;H_DOCWINDOW_CLASS;H_MENUITEM_CLASS',0)} [See list of classes](#)

{button ,AL('H_CAPTION_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write)

[MenuItem]

The name of a menu item that displays on a Word Pro menu.

[Window]

Text that displays on the title bar of a window.

[ApplicationWindow]

The name of the application window that displays in the title bar.

Data Type

String

Syntax

captionvalue = [objectreference].Caption

[objectreference].Caption = captionvalue

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Not valid for StatusBar.

[MenuItem]

Alters the name of a menu item that displays on a Word Pro menu. For example, you can write a script that changes a menu item's caption when an event is triggered in Word Pro.

To display an accelerator keystroke caption, such as the Ctrl+S accelerator keystroke for Save, you must manually create a string. You can right-align the accelerator keystroke caption in the menu with a special tab character that you create by using this right alignment variable:

```
Spacer$ = Chr$(8) .
```

You can see how to use this variable in the following example (shows you how to write a Word Pro Save menu caption and its accelerator key stroke caption, Ctrl+S):

```
MyMenuName$ = "&Save" & Spacer$ & "Ctrl+S"
```

[ApplicationWindow]

Sets or obtains the caption of the active application window. Not implemented for this class in Word Pro.

Word Pro: Case property

{button ,AL(^H_FONT_CLASS',0)} [See list of classes](#)

(Read-write)

Data Type

Variant (Enumerated)

Case

Syntax

casevalue = [objectreference].Case

[objectreference].Case = casevalue

Legal values

\$LtsCaseLower (1056964842)

\$LtsCaseMixed (1056964843)

\$LtsCaseUpper (1056964841)

\$LwpCaseDontcare (114)

\$LwpCaseInitcaps (115)

\$LwpCaseStyle (116)

Usage

Word Pro: CellEngines property

{button ,AL('H_FOUNDRY_CLASS',0)} [See list of classes](#)

(Read-only) An object created from the CellCollection class. This object provides access to CellEngine objects.

Data Type

[CellCollection](#)

Syntax

cellenginesvalue = [objectreference].CellEngines

Legal values

Always contains an instance of the CellCollection class.

Usage

When accessed through the Foundry property on a Division object, the collection object in this property provides access to all the CellEngine objects contained in that Division object.

When accessed through the AppFoundry property on the WPAApplication object, the collection object in this property provides access to all the CellEngine objects contained in the Word Pro Clipboard.

When accessed through the TempFoundry property on the WPAApplication object, the collection object in this property provides access to all the CellEngine objects placed in TempFoundry by WordPro or a script.

When accessed through the Foundry property on the WPAApplication object, the collection object in this property provides access to all the CellEngine objects contained in the currently active Division object.

This property is not used in the Foundry property on the TextDocument object.

For more information about collection classes, see [Overview: Word Pro LotusScript Collection Classes](#).

Word Pro: CellEngine property

{button ,AL(^H_TABLE_CLASS',0)} [See list of classes](#)

(Read-only) Allows you to access the CellEngine object within a table.

Data Type

[CellEngine](#)

Syntax

cellenginevalue = [objectreference].Cell Engine

Legal values

Always contains an instance of the CellEngine class.

Usage

Word Pro: CellLayoutStyles property

{button ,AL(^H_FOUNDRY_CLASS',0)} [See list of classes](#)

(Read-only) An object created from the CellLayoutCollection class. This object provides access to CellLayout objects that are used as cell styles. If there are no cell styles defined for a document, the collection object in this property will be empty.

Data Type

[CellLayoutCollection](#)

Syntax

cellLayoutStylesvalue = [objectreference].CellLayoutStyles

Legal values

Always contains an instance of the CellLayoutCollection class.

Usage

When accessed through the Foundry property on a Division object, the collection object in this property provides access to all the CellLayout objects used as cell styles in that Division object.

When accessed through the AppFoundry property on the WPAApplication object, the collection object in this property provides access to all the CellLayout objects used as styles contained in the Word Pro Clipboard.

When accessed through the TempFoundry property on the WPAApplication object, the collection object in this property provides access to all the CellLayout objects placed in TempFoundry by WordPro or a script.

When accessed through the Foundry property on the WPAApplication object, the collection object in this property provides access to all the CellLayout objects used as cell styles in the currently active Division object.

This property is not used in the Foundry property on the TextDocument object.

For more information about collection classes, see [Overview: Word Pro LotusScript Collection Classes](#).

Word Pro: CellLayouts property

{button ,AL(^H_FOUNDRY_CLASS;H_BASETABLE_CLASS;H_FOOTNOTETABLE_CLASS;H_PARALLELCOLUMNS_CLASS;H_GLOSSARY_CLASS;H_TABLEHEADING_CLASS;H_TABLE_CLASS',0)} [See list of classes](#)

(Read-only) An object created from the CellLayoutCollection class. This object provides access to all CellLayout objects, including those used as cell styles.

Data Type

[CellLayoutCollection](#)

Syntax

celllayoutsvalue = [objectreference].CellLayouts

Legal values

Always contains an instance of the CellLayoutCollection class.

Usage

When accessed through the Foundry property on a Division object, the collection object in this property provides access to all the CellLayout objects contained in that Division object.

When accessed through the AppFoundry property on the WPAApplication object, the collection object in this property provides access to all the CellLayout objects contained in the Word Pro Clipboard.

When accessed through the TempFoundry property on the WPAApplication object, the collection object in this property provides access to all the CellLayout objects placed in TempFoundry by WordPro or a script.

When accessed through the Foundry property on the WPAApplication object, the collection object in this property provides access to all the CellLayout objects contained in the currently active Division object.

This property is not used in the Foundry property on the TextDocument object.

For more information about collection classes, see [Overview: Word Pro LotusScript Collection Classes](#).

Word Pro: CellLayout property

{button ,AL('H_CELLCONTAINER_CLASS',0)} [See list of classes](#)

(Read-only) Returns the cell layout object from a cell container.

Data Type

[CellLayout](#)

Syntax

celllayoutvalue = [objectreference].CellLayout

Legal values

Always contains an instance of the CellLayout class.

Usage

Word Pro: CellStyleName property

{button ,AL('H_PREFERENCES_CLASS',0)} [See list of classes](#)

(Read-write)

Data Type

String

Syntax

cellstylevalue = [objectreference].CellStyleName

[objectreference].CellStyleName = cellstylevalue

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Word Pro: Cell property

{button ,AL(^H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL(^H_CELL_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-only) An instance of the CellContainer class. This is a current context property that only contains an object when the focus of Word Pro includes a cell in a table. If there is no cell in the focus, this property is empty.

Data Type

[CellContainer](#)

Syntax

propertycellvalue = [objectreference].Cell

Legal values

An instance of the CellContainer class.

Usage

When the focus includes a cell in a table, this property contains the CellContainer object that groups together the objects that comprise the cell with the focus. You can use this property to access the Layout or other objects related to that cell.

Word Pro: Center property

```
{button ,AL('H_LAYOUT_CLASS;H_CELLGROUPLAYOUT_CLASS;H_CELLLAYOUT_CLASS;H_CONNECTEDLAY  
OUT_CLASS;H_COLUMNGROUPLAYOUT_CLASS;H_FOOTERLAYOUT_CLASS;H_FRAMELAYOUT_CLASS;H  
_FRAMEGROUPLAYOUT_CLASS;H_DROP_CAPLAYOUT_CLASS;H_GROUPLAYOUT_CLASS;H_HEADERLAY  
OUT_CLASS;H_NOTELAYOUT_CLASS;H_PAGELAYOUT_CLASS;H_ROWGROUPLAYOUT_CLASS;H_ROWLA  
YOUT_CLASS;H_RUBYLAYOUT_CLASS;H_SUPERTABLEGROUPLAYOUT_CLASS;H_SUPERTABLELAYOUT  
_CLASS;H_TABLELAYOUT_CLASS;H_ENDNOTELAYOUT_CLASS;H_FOOTNOTELAYOUT_CLASS;H_TABLE  
HEADINGLAYOUT_CLASS;H_TOCSUPERTABLELAYOUT_CLASS',0)} See list of classes
```

```
{button ,AL('H_CENTER_PROPERTY_EXSCRIPT',1)} See example
```

(Read-write) Indicates whether or not a graphic object is centered horizontally and vertically within a layout object.

Data Type

[Integer](#)

Syntax

[objectreference].Center = centervalue

centervalue = [objectreference].Center

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Equivalent to the "Placement" setting on the Watermark panel of the InfoBox.

Word Pro: Changed property

{button ,AL('H_DOCUMENT_CLASS;H_TEXTDOCUMENT_CLASS;H_DIVISIONINFO_CLASS',0)} [See list of classes](#)

{button ,AL('H_CHANGED_PROPERTY_EXSCRIPT',1)} [See example](#)

[Document]

(Read-write) A flag for Word Pro to use when determining whether a document has changed since the last time it was saved.

[DivisionInfo]

(Write-only)

Data Type

[Integer](#)

Syntax

changedvalue = [objectreference].Changed

[objectreference].Changed = changedvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

[Document]

If you set this property to False, Word Pro will not prompt you to save the document when it is closed. You will lose any changes that you have made since the last time the document was saved.

If you set this property to True, Word Pro will prompt you to save the document when it is closed, even if no changes have been made.

Word Pro: ChangeKeyboardToLanguage property

{button ,AL('H_PREFERENCES_CLASS',0)} [See list of classes](#)

{button ,AL('H_CHANGEKEYBOARDTOLANGUAGE_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write)

Data Type

Integer.

Syntax

[objectreference].ChangeKeyboardToLanguage = changekeyboardtolanguagevalue

changekeyboardtolanguagevalue = [objectreference].ChangeKeyboardToLanguage

Legal values

Usage

Word Pro: ChangeTextToMatchkeyboard property

{button ,AL('H_PREFERENCES_CLASS',0)} [See list of classes](#)

{button ,AL('H_CHANGETEXTTOMATCHKEYBOARD_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write)

Data Type

Integer

Syntax

[objectreference].ChangeTextToMatchKeyboard = changetexttomatchkeyboardvalue

changetexttomatchkeyboardvalue = [objectreference].ChangeTextToMatchKeyboard

Legal values

Usage

Word Pro: CharacterBorderName property

{button ,AL('H_PARAGRAPHSTYLE_CLASS',0)} [See list of classes](#)

(Read-write)

Data Type

String

Syntax

characterbordernamevalue = [objectreference].CharacterBorderName

[objectreference].CharacterBorderName = characterbordernamevalue

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Word Pro: CharacterBorder property

{button ,AL(^H_CHARACTERSTYLE_CLASS;H_CLICKHERE_CLASS;H_TEXTMARKER_CLASS;H_PARAGRAPHS
STYLE_CLASS;H_TEXT_CLASS',0)} [See list of classes](#)

(Read-only)

Data Type

[CharacterBorder](#)

Syntax

characterbordervalue = [objectreference].CharacterBorder

Legal values

Always contains an instance of the CharacterBorder class.

Usage

Word Pro: CharacterSetName property

{button ,AL('H_OPTIONS_CLASS',0)} [See list of classes](#)

(Read-write)

Data Type

String

Syntax

charactersetNamevalue = [objectreference].CharacterSetName

[objectreference].CharacterSetName = charactersetNamevalue

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Word Pro: CharacterSet property

{button ,AL(^H_PREFERENCES_CLASS;H_FINDANDREPLACE_CLASS',0)} [See list of classes](#)

(Read-only) Enables you to use special characters to expand a find and replace search. Depends on the language set in Word Pro Preferences.

Data Type

[CharacterSet](#)

Syntax

charactersetvalue = [objectreference].CharacterSet

Legal values

Always contains an instance of the CharacterSet class.

Usage

Data type is String in the specific language. Equivalent to choosing Edit - Find & Replace Text, clicking Options, and selecting an option in the "Special characters help" list box. You can then enter these values in the "Find" and "Replace with" boxes on the Find & Replace bar. The values are:

<u>Value</u>	<u>Effect</u>
^? Any one character	Finds and replaces any one character. Can be used with other text to find and replace variations of a word.
^* Any characters	Finds and replaces zero or more characters in a word. Can be used with other text to find and replace variations of a word.
^+ Phrase	Finds and replaces zero or more characters across multiple words. Can be used with other text to find and replace phrases.
^p To end of paragraph	Finds and replaces zero or more characters across multiple words to the end of a paragraph.
^t Tab character	Finds and replaces spaces with a tab character.
^r Return character	Finds and replaces a hard return.
^^ ^character	Finds and replaces the ^ character.

Word Pro: CharacterStyleName property

{button ,AL(^H_CLICKHERE_CLASS;H_TEXTMARKER_CLASS;H_TEXT_CLASS',0)} [See list of classes](#)

(Read-write) The name of the character style at the insertion point.

Data Type

String

Syntax

characterstylevalue = [objectreference].CharacterStyleName

[objectreference].CharacterStyleName = characterstylevalue

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Word Pro: CharacterStyles property

{button ,AL('H_FOUNDRY_CLASS',0)} [See list of classes](#)

(Read-only) An object created from the CharacterStyleCollection class. This object provides access to CharacterStyle objects.

Data Type

[CharacterStyleCollection](#)

Syntax

characterstylesvalue = [objectreference].CharacterStyles

Legal values

Always contains an instance of the CharacterStyleCollection class.

Usage

When accessed through the Foundry property on a Division object, the collection object in this property provides access to all the CharacterStyle objects contained in that Division object.

When accessed through the AppFoundry property on the WPAApplication object, the collection object in this property provides access to all the CharacterStyle objects contained in the Word Pro Clipboard.

When accessed through the TempFoundry property on the WPAApplication object, the collection object in this property provides access to all the CharacterStyle objects placed in TempFoundry by WordPro or a script.

When accessed through the Foundry property on the WPAApplication object, the collection object in this property provides access to all the CharacterStyle objects contained in the currently active Division object.

This property is not used in the Foundry property on the TextDocument object.

For more information about collection classes, see [Overview: Word Pro LotusScript Collection Classes](#).

Word Pro: CharacterStyle property

{button ,AL(^H_CLICKHERE_CLASS;H_TEXTMARKER_CLASS;H_TEXT_CLASS',0)} [See list of classes](#)

(Read-only)

Data Type

[CharacterStyle](#)

Syntax

characterstylevalue = [objectreference].CharacterStyle

Legal values

Always contains an instance of the CharacterStyle class.

Usage

Word Pro: CharSet property

{button ,AL('H_CHARACTERSET_CLASS',0)} [See list of classes](#)

(Read-write)

Data Type

Variant (Enumerated)

CharSet

Syntax

charsetvalue = [objectreference].CharSet

[objectreference].CharSet = charsetvalue

Legal values

\$LwpCharSetAnsi (119)

\$LwpCharSetAnsi1250 (120)

\$LwpCharSetAnsi1251 (121)

\$LwpCharSetAnsi1252 (122)

\$LwpCharSetAnsi1253 (123)

\$LwpCharSetAnsi1254 (124)

\$LwpCharSetArabic (134)

\$LwpCharSetAscii (118)

\$LwpCharSetBig5 (126)

\$LwpCharSetCp850 (139)

\$LwpCharSetCp852 (2380)

\$LwpCharSetCp857 (2383)

\$LwpCharSetCp865 (2384)

\$LwpCharSetCp866 (2381)

\$LwpCharSetCp869 (2382)

\$LwpCharSetCp874 (2343)

\$LwpCharSetCp932 (125)

\$LwpCharSetCp949 (127)

\$LwpCharSetCyrillic (133)

\$LwpCharSetGreek (135)

\$LwpCharSetHebrew (136)

\$LwpCharSetIsolatin1 (129)

\$LwpCharSetIsolatin2 (130)

\$LwpCharSetIsolatin3 (131)

\$LwpCharSetIsolatin4 (132)

\$LwpCharSetIsolatin5 (137)

\$LwpCharSetIsolatin6 (138)

\$LwpCharSetUnicode (128)

Usage

Word Pro: Checked property

{button ,AL('H_MENUITEM_CLASS',0)} [See list of classes](#)

{button ,AL('H_CHECKED_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Indicates if a menu item is checked or not checked.

Data Type

Integer

Syntax

checkedvalue = [objectreference].Checked

[objectreference].Checked = checkedvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

To check a menu item, set this property value to True. To uncheck a menu item, set the property value to False.

Setting the Checked property is available only for custom created menu items. You cannot set the Checked property for predefined Word Pro menu items, because Word Pro dynamically sets this property for predefined items, based on the current context.

Word Pro: CheckForScriptEnumError property

{button ,AL('H_USERINTERFACEPREFS_CLASS',0)} [See list of classes](#)

(Read-write) This property is not implemented. Indicates if the Script Editor is set to check enumeration values.

Data Type

Integer (Bool)

Syntax

checkforscriptenumerrorvalue = [objectreference].CheckForScriptEnumError

[objectreference].CheckForScriptEnumError = checkforscriptenumerrorvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Provides a script writer with an additional level of error checking, while developing a script. Default is False (0), which causes the Script Editor to check the validity of property values beyond checking for data type. If the value for the property is changed to True (-1), the Script Editor generates an error if the value of a property does not match one of the enumerated legal values defined for that property. Enumerated legal values always begin with "\$" and always have a numeric equivalent.

Word Pro: CheckForScriptPropertyError property

{button ,AL('H_USERINTERFACEPREFS_CLASS',0)} [See list of classes](#)

(Read-write) Indicates if the Script Editor is set to check property values.

Data Type

Integer (Bool)

Syntax

checkforscriptpropertyerrorvalue = [objectreference].CheckForScriptPropertyError

[objectreference].CheckForScriptPropertyError = checkforscriptpropertyerrorvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Provides a script writer with an additional level of error checking. while developing a script. Default is False (0), which causes Word Pro not to check the value of a property after it is set. If the value for the property is changed to True (-1), the Script Editor checks the value of a property after the property is set. It does this in order to confirm that the property value matches the value called for in the script. If, for some reason the value of the property does not match the value in the script statement, the Script Editor raises the error, 7009 LWPERR SetPropFailed. You can include an OnError routine in your script to handle this error, or simply use this extra level of checking as a debugging tool.

Word Pro: ChildLayouts property

```
{button ,AL('H_LAYOUT_CLASS;H_CELLGROUPLAYOUT_CLASS;H_CELLLAYOUT_CLASS;H_CONNECTEDLAY  
OUT_CLASS;H_COLUMNGROUPLAYOUT_CLASS;H_FOOTERLAYOUT_CLASS;H_FRAMELAYOUT_CLASS;H  
_FRAMEGROUPLAYOUT_CLASS;H_DROP_CAPLAYOUT_CLASS;H_GROUPLAYOUT_CLASS;H_HEADERLAY  
OUT_CLASS;H_NOTELAYOUT_CLASS;H_PAGELAYOUT_CLASS;H_ROWGROUPLAYOUT_CLASS;H_ROWLA  
YOUT_CLASS;H_RUBYLAYOUT_CLASS;H_SUPERTABLEGROUPLAYOUT_CLASS;H_SUPERTABLELAYOUT  
_CLASS;H_TABLELAYOUT_CLASS;H_ENDNOTELAYOUT_CLASS;H_FOOTNOTELAYOUT_CLASS;H_TABLE  
HEADINGLAYOUT_CLASS;H_TOCSUPERTABLELAYOUT_CLASS',0)} See list of classes
```

(Read-only) Returns a collection of a layout object's children.

Data Type

[LayoutCollection](#)

Syntax

childlayoutsvalue = [objectreference].ChildLayouts

Legal values

Always contains an instance of the LayoutCollection class.

Usage

Returns a collection of a layout object's children. For example, a page layout object's children could include header and footer layout objects, as well as certain frames or tables.

Word Pro: CityState property

{button ,AL('H_PREFERENCES_CLASS',0)} [See list of classes](#)

(Read-write)

Data Type

String

Syntax

citystatevalue = [objectreference].CityState

[objectreference].CityState = citystatevalue

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Word Pro: ClassId property

{button ,AL(^H_OLEOBJECT_CLASS',0)} [See list of classes](#)

(Read-only)

Note This property is not supported for the class OLEObject within OS/2.

Data Type

String.

Syntax

classidvalue = [objectreference].ClassID

Legal values**Usage**

Word Pro: ClassName property

{button ,AL('H_DIVISIONINFO_CLASS;H_LAYOUT_CLASS;H_CELLGROUPLAYOUT_CLASS;H_CELLLAYOUT_CLASS;H_CONNECTEDLAYOUT_CLASS;H_COLUMNGROUPLAYOUT_CLASS;H_FOOTERLAYOUT_CLASS;H_FRAMELAYOUT_CLASS;H_FRAMEGROUPLAYOUT_CLASS;H_DROP_CAPLAYOUT_CLASS;H_GROUPLAYOUT_CLASS;H_HEADERLAYOUT_CLASS;H_NOTELAYOUT_CLASS;H_PAGELAYOUT_CLASS;H_ROWGROUPLAYOUT_CLASS;H_ROWLAYOUT_CLASS;H_RUBYLAYOUT_CLASS;H_SUPERTABLEGROUPLAYOUT_CLASS;H_SUPERTABLELAYOUT_CLASS;H_TABLELAYOUT_CLASS;H_ENDNOTELAYOUT_CLASS;H_FOOTNOTELAYOUT_CLASS;H_TABLEHEADINGLAYOUT_CLASS;H_TOCSUPERTABLELAYOUT_CLASS',0)} [See list of classes](#)

{button ,AL('H_CLASSNAME_PROPERTY_EXSCRIPT',1)} [See example](#)

[Layout]

(Read-only) The type of layout or content object from which you called this property.

[DivisionInfo]

(Read-write) The category of a document division.

Data Type

String

Syntax

classnamevariable = [objectreference].ClassName

[objectreference].ClassName = classnamevariable

Legal values

Usage

[Layout]

Check this property to see what type of object you are currently accessing. For example, the ClassName property of a frame layout object contains a string value of Frame.

[DivisionInfo]

This property is equivalent to the "Category" box in the General panel of the Document Properties dialog box.

Word Pro: CleanScreenMode property

{button ,AL(`H_USERINTERFACEPREFS_CLASS',0)} [See list of classes](#)

(Read-write) Indicates whether the current application window is in clean screen mode.

Data Type

[Integer\(Bool\)](#)

Syntax

cleanscreenmodevalue = [objectreference].CleanScreenMode

[objectreference].CleanScreenMode = cleanscreenmodevalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

If tset to False (0), Word Pro is not in clean screen mode. If set to True (-1), Word Pro is in clean screen mode.

Setting this property to True does not enable clean screen mode. Use the ToggleCleanScreen method to enable or disable clean screen mode.

{button ,AL(`H_TOGGLECLEANSCREEN_METHOD_MEMDEF',0)} [See related topics](#)

Word Pro: CleanScreenOnStartUp property

{button ,AL('H_USERINTERFACEPREFS_CLASS',0)} [See list of classes](#)

{button ,AL('H_CLEANSCREENONSTARTUP_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Indicates if clean screen mode is enabled when starting Word Pro.

Data Type

[Integer\(Bool\)](#)

Syntax

cleanscreenonstartupvalue = [objectreference].CleanScreenOnStartUp

[objectreference].CleanScreenOnStartUp = cleanscreenonstartupvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values. Default is False (0).

Usage

Equivalent to the "Load in clean screen view" option on the General panel of the Word Pro Preferences dialog box. If the legal value is False, Word Pro does not automatically load in clean screen view.

Word Pro: ClickHerePrompts property

{button ,AL('H_OPTIONS_CLASS',0)} [See list of classes](#)

{button ,AL('H_CLICKHEREPROMPTS_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Allows you to display unfilled Click Here Block prompts in the document.

Data Type

[Integer](#)

Syntax

clickherepromptsvalue = [objectreference].ClickHerePrompts

[objectreference].ClickHerePrompts = clickherepromptsvalue

Legal values

The legal values for this property are -1 and 0, but you may use the LotusScript constants of True (-1) to display unfilled Click Here Block prompts, and False (0) to hide them, instead of the integer values..

Usage

Equivalent to choosing File - Document Properties - Document and selecting "Show unfilled Click Here Block prompts."

{button ,AL('H_ISVIEWCLICKHEREBLOCKS_PROPERTY_MEMDEF',0)} [See related topics](#)

Word Pro: ClickHeres property

{button ,AL('H_FOUNDRY_CLASS',0)} [See list of classes](#)

{button ,AL('H_CLICKHERES_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-only) An object created from the ClickHereCollection class. This object provides access to ClickHere objects.

Data Type

[ClickHereCollection](#)

Syntax

clickheresvalue = [objectreference].ClickHeres

Legal values

Always contains an instance of the ClickHereCollection class.

Usage

When accessed through the Foundry property on a Division object, the collection object in this property provides access to all the ClickHere objects contained in that Division object.

When accessed through the AppFoundry property on the WPAApplication object, the collection object in this property provides access to all the ClickHere objects contained in the Word Pro Clipboard.

When accessed through the TempFoundry property on the WPAApplication object, the collection object in this property provides access to all the ClickHere objects placed in TempFoundry by WordPro or a script.

When accessed through the Foundry property on the WPAApplication object, the collection object in this property provides access to all the ClickHere objects contained in the currently active Division object.

This property is not used in the Foundry property on the TextDocument object.

For more information about collection classes, see [Overview: Word Pro LotusScript Collection Classes](#).

Word Pro: ClientHeight property

```
{button ,AL(^H_BASECONTAINER_CLASS;H_CELLCONTAINER_CLASS;H_FRAMECONTAINER_CLASS;H_NOTECONTAINER_CLASS;H_RUBYCONTAINER_CLASS;H_DROPDOWNCONTAINER_CLASS;H_PAGECONTAINER_CLASS;H_SUBPAGECONTAINER_CLASS;H_SUPERPAGECONTAINER_CLASS;H_ROWCONTAINER_CLASS;H_SUPERTABLECONTAINER_CLASS;H_TABLECONTAINER_CLASS;H_PARALLELCOLSCONTAINER_CLASS;H_TABLEONLYCONT_CLASS',0)} See list of classes
```

(Read-only) Returns the height of the content area within the margins of a container.

Data Type

Long

Syntax

clientheightvalue = [objectreference].ClientHeight

Legal values

Data type is Long but the unit of measurement used for this property is Twips. There are 1440 Twips per inch.

Usage

Word Pro: ClientWidth property

```
{button ,AL(^H_BASECONTAINER_CLASS;H_CELLCONTAINER_CLASS;H_FRAMECONTAINER_CLASS;H_NOTE  
CONTAINER_CLASS;H_RUBYCONTAINER_CLASS;H_DROPDOWNCONTAINER_CLASS;H_PAGECONTAINER_  
CLASS;H_SUBPAGECONTAINER_CLASS;H_SUPERPAGECONTAINER_CLASS;H_ROWCONTAINER_CLASS;  
H_SUPERTABLECONTAINER_CLASS;H_TABLECONTAINER_CLASS;H_PARALLELCOLSCONTAINER_CLASS  
;H_TABLEONLYCONT_CLASS',0)} See list of classes
```

(Read-only) Returns the width of the content area within the margins of a container.

Data Type

Long

Syntax

clientwidthvalue = [objectreference].ClientWidth

Legal values

Data type is Long but the unit of measurement used for this property is Twips. There are 1440 Twips per inch.

Usage

Word Pro: ClientWndHeight property

{button ,AL(^H_DOCWINDOW_CLASS',0)} [See list of classes](#)

(Read-only)

Data Type

Long

Syntax

clientwndheightvalue = [objectreference].ClientWndHeight

Legal values

Data type is Long but the unit of measurement used for this property is Twips. There are 1440 Twips per inch.

Usage

Word Pro: ClientWndWidth property

{button ,AL(^H_DOCWINDOW_CLASS',0)} [See list of classes](#)

(Read-only)

Data Type

Long

Syntax

clientwndwidthvalue = [objectreference].ClientWndWidth

Legal values

Data type is Long but the unit of measurement used for this property is Twips. There are 1440 Twips per inch.

Usage

Word Pro: CloseDocMacroName property

{button ,AL('H_AUTORUNMACRO_CLASS',0)} [See list of classes](#)

(Read-write)

Data Type

String

Syntax

closedocmacrovalue = [objectreference].CloseDocMacroName

[objectreference].CloseDocMacroName = closedocmacrovalue

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Word Pro: CodePage property

{button ,AL(^H_CLICKHERE_CLASS;H_TEXTMARKER_CLASS;H_TEXT_CLASS',0)} [See list of classes](#)

(Read-write)

Data Type

Integer

Syntax

codepagevalue = [objectreference].CodePage

[objectreference].CodePage = codepagevalue

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Word Pro: Collapsible property

{button ,AL('H_TEXTMARKER_CLASS',0)} [See list of classes](#)

{button ,AL('H_COLLAPSIBLE_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write)

Data Type

[Integer](#)

Syntax

collapsiblevalue = [objectreference].Collapsible

[objectreference].Collapsible = collapsiblevalue

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Word Pro: Collate property

{button ,AL('H_PRINTSETTINGS_CLASS',0)} [See list of classes](#)

{button ,AL('H_COLLATE_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Determines whether an entire copy of the document prints before printing the next copy. This property only has an effect if the document contains multiple pages.

Data Type

[Integer](#)

Syntax

collatevalue = [objectreference].Collate

[objectreference].Collate = collatevalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Determines whether or not a document is collated when printed. Equivalent to the "Collate" option in the Word Pro Print dialog box.

{button ,AL('H_COPIES_PROPERTY_MEMDEF',0)} [See related topics](#)

Word Pro: ColorOverride property

{button ,AL('H_NUMERICFORMATSUBSET_CLASS',0)} [See list of classes](#)

{button ,AL('H_COLOROVERRIDE_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Indicates whether or not to override the font color with the color specified in the appropriate NumericFormatSubset object.

Data Type

[Integer](#)

Syntax

coloroverridevalue = [objectreference].ColorOverride

[objectreference].ColorOverride = coloroverridevalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Equivalent to the "Number color" option in the Edit Format dialog box. You can access this dialog box by clicking the Format Options button on the Number Format panel of the InfoBox for cell layout objects.

If the ColorOverride property is set to True, the font color of a numeric value is overridden with the specified NumericFormatSubset color.

Word Pro: Color property

{button ,AL(^H_BACKGROUND_CLASS;H_DIVISIONINFO_CLASS;H_BORDER_CLASS;H_NOTELAYOUT_CLASS ;H_NUMERICFORMATSUBSET_CLASS;H_SECTION_CLASS;H_INDEXSECTION_CLASS;H_SHADOW_CLASS',0)} [See list of classes](#)

{button ,AL(^H_COLOR_PROPERTY_EXSCRIPT',1)} [See example](#)

[NumericFormatSubset]

(Read-only) The numeric format color of a numeric value within a table cell.

[Shadow]

(Read-only) Returns the color object of a shadow for text, frames, headers, footers, and pages in a document.

Data Type

Color

Syntax

colorvalue = [objectreference].Color

[objectreference].Color = colorvalue

Legal values

Always contains an instance of the Color class.

Usage

[NumericFormatSubset]

Use the color property to change the color of numeric values in a table cell. When used, this color overrides the font color assigned to text within a table cell.

[Shadow]

You can set the properties of a Shadow's Color object for text, frames, tables, headers, footers, and pages in a document. Equivalent to:

- Clicking the right mouse button anywhere in the text, choosing Text Properties, clicking the Lines & Colors tab, and selecting an option in the "Shadow" box.
- Clicking the right mouse button anywhere in the frame, choosing Frame Properties, clicking the Lines & Colors tab, and selecting an option in the "Shadow" box.
- Clicking the right mouse button anywhere in the table, choosing Table Properties, clicking the Lines & Colors tab, and selecting an option in the "Shadow" box.
- Clicking the right mouse button anywhere in the header, choosing Header Properties, clicking the Lines & Colors panel, and selecting an option in the "Shadow" box.
- Clicking the right mouse button anywhere in the footer, choosing Footer Properties, clicking the Lines & Colors tab, and selecting an option in the "Shadow" box.
- Clicking the right mouse button anywhere in the page, choosing Page Properties, clicking the Lines & Colors tab, and selecting an option in the "Shadow" box.

Word Pro: ColumnBalance property

```
{button ,AL('H_LAYOUT_CLASS;H_CELLGROUPLAYOUT_CLASS;H_CELLLAYOUT_CLASS;H_CONNECTEDLAY  
OUT_CLASS;H_COLUMNGROUPLAYOUT_CLASS;H_FOOTERLAYOUT_CLASS;H_FRAMELAYOUT_CLASS;H  
_FRAMEGROUPLAYOUT_CLASS;H_DROP_CAPLAYOUT_CLASS;H_GROUPLAYOUT_CLASS;H_HEADERLAY  
OUT_CLASS;H_NOTELAYOUT_CLASS;H_PAGELAYOUT_CLASS;H_ROWGROUPLAYOUT_CLASS;H_ROWLA  
YOUT_CLASS;H_RUBYLAYOUT_CLASS;H_SUPERTABLEGROUPLAYOUT_CLASS;H_SUPERTABLELAYOUT  
_CLASS;H_TABLELAYOUT_CLASS;H_ENDNOTELAYOUT_CLASS;H_FOOTNOTELAYOUT_CLASS;H_TABLE  
HEADINGLAYOUT_CLASS;H_TOCSUPERTABLELAYOUT_CLASS',0)} See list of classes
```

```
{button ,AL('H_COLUMNBALANCE_PROPERTY_EXSCRIPT',1)} See example
```

(Read-write) Indicates whether or not text is balanced in each column of certain layout objects.

Data Type

[Integer](#)

Syntax

columnbalancevalue = [objectreference].ColumnBalance

[objectreference].ColumnBalance = columnbalancevalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Equivalent to the "Column balance" setting on the Columns panel of the InfoBox for certain layout objects.

Word Pro: ColumnGap property

```
{button ,AL('H_LAYOUT_CLASS;H_CELLGROUPLAYOUT_CLASS;H_CELLLAYOUT_CLASS;H_CONNECTEDLAY  
OUT_CLASS;H_COLUMNGROUPLAYOUT_CLASS;H_FOOTERLAYOUT_CLASS;H_FRAMELAYOUT_CLASS;H  
_FRAMEGROUPLAYOUT_CLASS;H_DROP_CAPLAYOUT_CLASS;H_GROUPLAYOUT_CLASS;H_HEADERLAY  
OUT_CLASS;H_NOTELAYOUT_CLASS;H_PAGELAYOUT_CLASS;H_ROWGROUPLAYOUT_CLASS;H_ROWLA  
YOUT_CLASS;H_RUBYLAYOUT_CLASS;H_SUPERTABLEGROUPLAYOUT_CLASS;H_SUPERTABLELAYOUT  
_CLASS;H_TABLELAYOUT_CLASS;H_ENDNOTELAYOUT_CLASS;H_FOOTNOTELAYOUT_CLASS;H_TABLE  
HEADINGLAYOUT_CLASS;H_TOCSUPERTABLELAYOUT_CLASS',0)} See list of classes
```

```
{button ,AL('H_COLUMNGAP_PROPERTY_EXSCRIPT',1)} See example
```

(Read-write) The distance between one newspaper column and the next newspaper column to the right.

Data Type

Long

Syntax

columngapvalue = [objectreference].ColumnGap

[objectreference].ColumnGap = columngapvalue

Legal values

Data type is Long but the unit of measurement used for this property is Twips. There are 1440 Twips per inch.

Usage

Equivalent to the "Space between columns" setting on the Columns panel of the InfoBox for certain layout objects.

Word Pro: ColumnLayouts property

{button ,AL(^H_BASETABLE_CLASS;H_FOOTNOTETABLE_CLASS;H_PARALLELCOLUMNS_CLASS;H_GLOSSARY_CLASS;H_TABLEHEADING_CLASS;H_TABLE_CLASS',0)} [See list of classes](#)

(Read-only) Provides the names of any column layout objects within a table.

Data Type

[StringCollection](#)

Syntax

columnlayoutsvalue = [objectreference].ColumnLayouts

Legal values

Always contains an instance of the StringCollection class.

Usage

Column layout objects are created when the layout of a column is modified. For example, if you modify the width of the first column in a table, a column layout object is created for only that column. The name of the column layout object is then stored in the ColumnLayouts property. Column layout objects do not exist for columns that were never modified.

Word Pro: ColumnNumber property

{button ,AL(^H_CLICKHERE_CLASS;H_TEXTMARKER_CLASS;H_TEXT_CLASS',0)} [See list of classes](#)

(Read-only) The number of the page column in which the insertion point is located.

Data Type

Integer

Syntax

columnnumbervariable = [objectreference].ColumnNumber

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Does not refer to table columns or parallel columns. Instead, it refers to the standard columns, such as those found in a page layout.

Word Pro: ColumnWidth property

{button ,AL(^H_CLICKHERE_CLASS;H_TEXTMARKER_CLASS;H_TEXT_CLASS',0)} [See list of classes](#)

(Read-only) The width of the column in which the insertion point is located. Does not refer to a parallel or table column.

Data Type

Long

Syntax

columnwidthvalue = [objectreference].ColumnWidth

Legal values

Data type is Long but the unit of measurement used for this property is Twips. There are 1440 Twips per inch.

Usage

Word Pro: Company property

{button ,AL('H_PREFERENCES_CLASS',0)} [See list of classes](#)

(Read-write)

Data Type

String

Syntax

companyvalue = [objectreference].Company

[objectreference].Company = companyvalue

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Word Pro: ConditionType property

```
{button ,AL(`H_LAYOUT_CLASS;H_CELLGROUPLAYOUT_CLASS;H_CELLLAYOUT_CLASS;H_CONNECTEDLAYOUT_CLASS;H_COLUMNGROUPLAYOUT_CLASS;H_FOOTERLAYOUT_CLASS;H_FRAMELAYOUT_CLASS;H_FRAMEGROUPLAYOUT_CLASS;H_DROPDOWNLAYOUT_CLASS;H_GROUPLAYOUT_CLASS;H_HEADERLAYOUT_CLASS;H_NOTELAYOUT_CLASS;H_PAGELAYOUT_CLASS;H_ROWGROUPLAYOUT_CLASS;H_ROWLAYOUT_CLASS;H_RUBYLAYOUT_CLASS;H_SUPERTABLEGROUPLAYOUT_CLASS;H_SUPERTABLELAYOUT_CLASS;H_TABLELAYOUT_CLASS;H_ENDNOTELAYOUT_CLASS;H_FOOTNOTELAYOUT_CLASS;H_TABLEHEADINGLAYOUT_CLASS;H_TOCSUPERTABLELAYOUT_CLASS',0)} See list of classes
```

```
{button ,AL(`H_CONDITIONTYPE_PROPERTY_EXSCRIPT',1)} See example
```

(Read-write) Determines when a specific layout object is used.

Data Type

Data type is [Variant](#), which allows the value of this property to be one of the constants listed below or its numeric equivalent (in parentheses).

Syntax

[objectreference].ConditionType = conditiontypevalue

conditiontypevalue = [objectreference].ConditionType

Legal values

<u>Value</u>	<u>Effect</u>
\$LwpConditionTypeAllbutspecificpage (156)	Inserts a certain layout object in all pages, except a specific page in a document.
\$LwpConditionTypeAllpages (154)	Inserts a certain layout object in all pages of a document.
\$LwpConditionTypeOnlyevenpages (157)	Inserts a certain layout object in even pages of a document.
\$LwpConditionTypeOnlyoddpages (158)	Inserts a certain layout object in odd pages of a document.
\$LwpConditionTypeOnlyspecificpage (155)	Inserts a certain layout object in specific pages of a document.
\$LwpConditionTypeStartatpage (159)	Specifies on which page to insert a certain layout object.

Usage

Use this property in conjunction with other properties, such as PageToUseLayoutOn. If the parent layout is not a page, then a layout uses the parent's ConditionType.

```
{button ,AL(`H_PAGETOUSELAYOUTON_PROPERTY_MEMDEF',0)} See related topics
```

Word Pro: Condition property

{button ,AL('H_USEWHEN_CLASS',0)} [See list of classes](#)

(Read-write)

Data Type

Variant (Enumerated)

ConditionType

Syntax

conditionvalue = [objectreference].Condition

[objectreference].Condition = conditionvalue

Legal values

\$LwpConditionTypeAllbutspecificpage (156)

\$LwpConditionTypeAllpages (154)

\$LwpConditionTypeOnlyevenpages (157)

\$LwpConditionTypeOnlyoddpages (158)

\$LwpConditionTypeOnlyspecificpage (155)

\$LwpConditionTypeStartatpage (159)

Usage

Word Pro: ConnectedLayouts property

{button ,AL('H_FOUNDRY_CLASS',0)} [See list of classes](#)

(Read-only) An object created from the ConnectedLayoutCollection class. This object provides access to ConnectedLayoutCollection objects.

Data Type

[ConnectedLayoutCollection](#)

Syntax

connectedlayoutsvalue = [objectreference].ConnectedLayouts

Legal values

Always contains an instance of the ConnectedLayoutCollection class.

Usage

When accessed through the Foundry property on a Division object, the collection object in this property provides access to all the CellLayout objects contained in that Division object.

When accessed through the AppFoundry property on the WPAApplication object, this collection object provides access to all the CellLayout objects contained in the Word Pro Clipboard.

When accessed through the TempFoundry property on the WPAApplication object, this collection object provides access to all the CellLayout objects placed in TempFoundry by WordPro or a script.

When accessed through the Foundry property on the WPAApplication object, this collection object provides access to all the CellLayout objects contained in the currently active Division object.

This property is not used in the Foundry property on the TextDocument object.

For more information about collection classes, see [Overview: Word Pro LotusScript Collection Classes](#).

Word Pro: Container property

{button ,AL('H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL('H_CONTAINER_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-only) A "catch all" container property that always contains the topmost container object in the focus. Container uses the abstract class BaseContainer as its data type, which allows Container to store any kind of container object.

Data Type

[BaseContainer](#)

Syntax

containervalue = [objectreference].Container

Legal values

An instance of the BaseContainer class.

Usage

Use this property to access the topmost container object's layout, regardless of that container object's contents. For example, you may write a script that allows you to select an object whose background you want to be red. After you select the object (thus setting the focus on that object), you can issue this statement:

```
.Container.Layout.Background.Color.SetRGB 255, 0, 0
```

This statement sets the color of the layout's background to red, regardless of the kind of layout or the kind of container object you selected.

Word Pro: ContentHeight property

```
{button ,AL('H_BASECONTAINER_CLASS;H_CELLCONTAINER_CLASS;H_FRAMECONTAINER_CLASS;H_NOTE  
CONTAINER_CLASS;H_RUBYCONTAINER_CLASS;H_DROPDOWNCONTAINER_CLASS;H_PAGECONTAINER_  
CLASS;H_SUBPAGECONTAINER_CLASS;H_SUPERPAGECONTAINER_CLASS;H_ROWCONTAINER_CLASS;  
H_SUPERTABLECONTAINER_CLASS;H_TABLECONTAINER_CLASS;H_PARALLELCOLSCONTAINER_CLASS  
;H_TABLEONLYCONT_CLASS',0)} See list of classes
```

(Read-only) Returns the height of the content based on its rotation within a container.

Data Type

Long

Syntax

contentheightvalue = [objectreference].ContentHeight

Legal values

Data type is Long but the unit of measurement used for this property is Twips. There are 1440 Twips per inch.

Usage

Word Pro: ContentName property

{button ,AL(^H_BASECONTAINER_CLASS;H_CELLCONTAINER_CLASS;H_FRAMECONTAINER_CLASS;H_NOTECONTAINER_CLASS;H_RUBYCONTAINER_CLASS;H_DROPCAPCONTAINER_CLASS;H_PAGECONTAINER_CLASS;H_SUBPAGECONTAINER_CLASS;H_SUPERPAGECONTAINER_CLASS;H_ROWCONTAINER_CLASS;H_SUPERTABLECONTAINER_CLASS;H_TABLECONTAINER_CLASS;H_PARALLELCOLSCONTAINER_CLASS;H_TABLEONLYCONT_CLASS;H_DIVISIONINFO_CLASS;H_LAYOUT_CLASS;H_CELLGROUPLAYOUT_CLASSES;H_CELLLAYOUT_CLASS;H_CONNECTEDLAYOUT_CLASS;H_COLUMNGROUPLAYOUT_CLASS;H_FOOTERLAYOUT_CLASS;H_FRAMELAYOUT_CLASS;H_FRAMEGROUPLAYOUT_CLASS;H_DROPCAPLAYOUT_CLASSES;H_GROUPLAYOUT_CLASS;H_HEADERLAYOUT_CLASS;H_NOTELAYOUT_CLASS;H_PAGELAYOUT_CLASSES;H_ROWGROUPLAYOUT_CLASS;H_ROWLAYOUT_CLASS;H_RUBYLAYOUT_CLASS;H_SUPERTABLEGROUPLAYOUT_CLASS;H_SUPERTABLELAYOUT_CLASS;H_TABLELAYOUT_CLASS;H_ENDNOTELAYOUT_CLASSES;H_FOOTNOTELAYOUT_CLASS;H_TABLEHEADINGLAYOUT_CLASS;H_TOCSUPERTABLELAYOUT_CLASSES',0)} [See list of classes](#)

(Read-only) Returns the name of the content object in any container.

Data Type

String

Syntax

contentnamevalue = [objectreference].ContentName

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Word Pro: ContentStyleName property

```
{button ,AL(^H_LAYOUT_CLASS;H_CELLGROUPLAYOUT_CLASS;H_CELLLAYOUT_CLASS;H_CONNECTEDLAY  
OUT_CLASS;H_COLUMNGROUPLAYOUT_CLASS;H_FOOTERLAYOUT_CLASS;H_FRAMELAYOUT_CLASS;H  
_FRAMEGROUPLAYOUT_CLASS;H_DROP_CAPLAYOUT_CLASS;H_GROUPLAYOUT_CLASS;H_HEADERLAY  
OUT_CLASS;H_NOTELAYOUT_CLASS;H_PAGELAYOUT_CLASS;H_ROWGROUPLAYOUT_CLASS;H_ROWLA  
YOUT_CLASS;H_RUBYLAYOUT_CLASS;H_SUPERTABLEGROUPLAYOUT_CLASS;H_SUPERTABLELAYOUT  
_CLASS;H_TABLELAYOUT_CLASS;H_ENDNOTELAYOUT_CLASS;H_FOOTNOTELAYOUT_CLASS;H_TABLE  
HEADINGLAYOUT_CLASS;H_TOCSUPERTABLELAYOUT_CLASS',0)} See list of classes
```

(Read-write) The name of the initial paragraph style used by certain layout objects.

Data Type

String

Syntax

contentstylevalue = [objectreference].ContentStyleName

[objectreference].ContentStyleName = contentstylevalue

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Stores the name of the initial paragraph style used by a layout object. This is not the same as the paragraph style assigned to the currently active text object. To access the name of the paragraph style assigned to the current text object, check the text object's ParagraphStyleName property.

Equivalent to the "Initial paragraph style" setting on the Miscellaneous panel of the InfoBox for certain layout objects.

Word Pro: ContentType property

{button ,AL('H_CONTENT_CLASS;H_GRAPHICOLEBJECT_CLASS;H_GRAPHIC_CLASS;H_OLEOBJECT_CLASSES;H_SUPERTABLE_CLASS;H_BASSETABLE_CLASS;H_FOOTNOTETABLE_CLASS;H_PARALLELCOLUMNS_CLASS;H_GLOSSARY_CLASS;H_TABLEHEADING_CLASS;H_TABLE_CLASS;H_TEXT_CLASS',0)} [See list of classes](#)

{button ,AL('H_CONTENTTYPE_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-only) Indicates the type of content object in which the insertion point is located.

Data Type

Variant

Syntax

contenttypevariable = [objectreference].ContentType

Legal values

<u>Value</u>	<u>Effect</u>
\$LwpContentTypeText (161)	The content type is a Text object.
\$LwpContentTypeTableonly (162)	The content type is a TableOnly object.
\$LwpContentTypeGraphic (163)	The content type is a Graphic object.
\$LwpContentTypeOleobject (164)	The content type is a OleObject.
\$LwpContentTypeParallelcolumns (165)	The content type is a ParallelColumns object.
\$LwpContentTypeFootnotetable (166)	The content type is a FootnoteTable object.
\$LwpContenttTypeFormula (167)	The content type is a Formula object.
\$LwpContentTypeFormula (2056)	The content type is a Formula object.

Usage

Verifies a specific type of content object in a layout. For example, you can use this property to determine whether or not a certain frame contains a graphic object.

Word Pro: ContentWidth property

```
{button ,AL(^H_BASECONTAINER_CLASS;H_CELLCONTAINER_CLASS;H_FRAMECONTAINER_CLASS;H_NOTE  
CONTAINER_CLASS;H_RUBYCONTAINER_CLASS;H_DROPDOWNCONTAINER_CLASS;H_PAGECONTAINER_  
CLASS;H_SUBPAGECONTAINER_CLASS;H_SUPERPAGECONTAINER_CLASS;H_ROWCONTAINER_CLASS;  
H_SUPERTABLECONTAINER_CLASS;H_TABLECONTAINER_CLASS;H_PARALLELCOLSCONTAINER_CLASS  
;H_TABLEONLYCONT_CLASS',0)} See list of classes
```

(Read-only) Returns the width of the content based on its rotation within a container.

Data Type

Long

Syntax

contentwidthvalue = [objectreference].ContentWidth

Legal values

Data type is Long but the unit of measurement used for this property is Twips. There are 1440 Twips per inch.

Usage

Word Pro: Content property

```
{button ,AL('H_WPAPPLICATION_CLASS;H_FOOTNOTE_CLASS;H_LAYOUT_CLASS;H_CELLGROUPLAYOUT_C  
LASS;H_CELLLAYOUT_CLASS;H_CONNECTEDLAYOUT_CLASS;H_COLUMNGROUPLAYOUT_CLASS;H_FO  
OTERLAYOUT_CLASS;H_FRAMELAYOUT_CLASS;H_FRAMEGROUPLAYOUT_CLASS;H_DROPPLAYOUT  
_CLASS;H_GROUPLAYOUT_CLASS;H_HEADERLAYOUT_CLASS;H_NOTELAYOUT_CLASS;H_PAGELAYOUT  
_CLASS;H_ROWGROUPLAYOUT_CLASS;H_ROWLAYOUT_CLASS;H_RUBYLAYOUT_CLASS;H_SUPERTABL  
EGROUPLAYOUT_CLASS;H_SUPERTABLELAYOUT_CLASS;H_TABLELAYOUT_CLASS;H_ENDNOTELAYOU  
T_CLASS;H_FOOTNOTELAYOUT_CLASS;H_TABLEHEADINGLAYOUT_CLASS;H_TOCSUPERTABLELAYOUT  
_CLASS',0)} See list of classes
```

```
{button ,AL('H_CONTENT_PROPERTY_EXSCRIPT',1)} See example
```

(Read-only) An instance of one of the content classes. A content class is any class derived from the abstract class called Content. Each content class describes a different kind of content, such as text, a graphic, or a cell formula. This property uses Content as its data type, so it can store any object created from one of these content classes. For more details on the Content class, content classes, and content objects, click the data type below.

Data Type

[Content](#)

Syntax

contentvalue = [objectreference].Content

Legal values

This property can contain an instance of one of the following classes:

FootnoteTable

Formula

Glossary

Graphic

OleObjects

Note OLE is not supported under OS/2.

ParallelColumns

SuperTable

TableHeading

Table

Text

Usage

The type of content object you find in this property depends on the object from which you call the property, and what objects are in the focus when you call the Content property.

WPAplication.Content

If you call the Content property on the WPAplication object, you get the uppermost content object within the focus. Therefore, the content object in this property changes as the focus changes in your document. This content object can be any of the types listed above, under Legal values.

Footnote.Content

If you call the Content property on a Footnote object, you get the content object for that footnote. This content object can be any of the types listed above, under Legal values.

layoutobject.Content

The Layout class has a Content property which is inherited by each layout class. A layout class is any class derived from the Layout class. A layout object is any object created from one of the layout classes. If you call the Content property on a layout object, you get the content object of that layout object. Usually, the type of content object found in the Content property corresponds to the type of object represented by that layout.

For example, the Content property on a CellLayout object may contain a Formula content object. But the Content property on a PageLayout object most likely is a Text content object. However, you must keep in mind that a cell or a page can also contain a graphic or an OLE object, and this affects the type of content object stored in the Content

property. The types of layout objects that have a Content property are:

CellGroupLayout

CellLayout

ConnectedLayout

EndnoteLayout

FooterLayout

FootnoteLayout

FrameLayout

GroupLayout

HeaderLayout

NoteLayout

PageLayout

RowLayout

RubyLayout

SuperTableGroupLayout

SuperTableLayout

TableHeadingLayout

TableLayout

TOCSuperTableLayout

Note that some of these layout objects are stored in properties of other objects, such as CellLayout that can be found in the CurrentCell property on WPApplication, and the CellLayout property on CellContainer. Other layout objects listed above may not be stored in a property, but can still be accessed through the appropriate collection object.

Word Pro: ContextLocked property

{button ,AL(^H_DIVISION_CLASS;H_TEXTDOCUMENT_CLASS',0)} [See list of classes](#)

(Read-only)

Indicates if the user is prohibited from editing at his/her current location.

Data Type

[Integer](#)

Syntax

contextlockedvalue = [objectreference].ContextLocked

Legal values

This property is read-only. The value of this property cannot be set by a script.

Usage

The user may be locked out of editing for several reasons - the document is read-only, the document is locked for annotation only, the document is not the most current version.

Word Pro: ContextMenuOptions property

{button ,AL(^H_APPLICATIONWINDOW_CLASS',0)} [See list of classes](#)

(Read-only) An instance of the ContextMenuOptions class which is menu options in a context sensitive dialog box or bar.

Data Type

[ContextMenuOptions](#)

Syntax

contextmenuoptionsvalue = [objectreference].ContextMenuOptions

Legal values

Always contains an instance of the ContextMenuOptions class.

Usage

Use this property to see if menus or menu items are grayed.

Word Pro: ContinuedFromAlignment property

{button ,AL('H_FOOTNOTEOPTIONS_CLASS',0)} [See list of classes](#)

{button ,AL('H_CONTINUEDFROMALIGNMENT_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Sets or returns the alignment for a footnote "continued from" message.

Data Type

Data type is [Variant](#), which allows the value of this property to be one of the constants listed below or its numeric equivalent (in parentheses).

Syntax

continuedfromalignmentvalue = [objectreference].ContinuedFromAlignment

[objectreference].ContinuedFromAlignment = continuedfromalignmentvalue

Legal values

<u>Value</u>	<u>Effect</u>
\$LtsAlignmentHorizCenter (1056964611)	The footnote is centered horizontally.
\$LtsAlignmentJustify (1056964613)	Not used.
\$LtsAlignmentLeft (1056964609)	The footnote is left-aligned.
\$LtsAlignmentRight (1056964610)	The footnote is right-aligned.
\$LtsAlignmentSmart (1056964612)	Not used.
\$LwpAlignmentTypeAlignRevert (8)	Not used.
\$LwpAlignmentTypeJustifyall (5)	Not used.
\$LwpAlignmentTypeNumericleft (6)	Not used.
\$LwpAlignmentTypeNumericright (7)	Not used.

Usage

Equivalent to the "continued from" Alignment setting on the Continued Messages panel of the Footnote and Endnote Options dialog box.

Word Pro: Contents property

{button ,AL('H_DOCINFOFIELD_CLASS;H_FOUNDRY_CLASS',0)} [See list of classes](#)

(Read-write) Sets or returns the value of a DocInfo field.

Data Type

String

Syntax

contentsvalue = [objectreference].Contents

[objectreference].Contents = contentsvalue

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

When you add a DocInfo field, you assign a name and a value to it. The content is the value of a DocInfo field and is always a String value. For example, if you add a DocInfo field and assign the name "Client" and the value "Active," then the Contents property value would be "Active".

To change the value of the "Client" DocInfo field from "Active" to "Inactive," you assign the string value, "Inactive," to the Contents property.

Word Pro: Contents property

{button ,AL('H_FOUNDRY_CLASS',0)} [See list of classes](#)

(Read-only)

An object created from the ContentCollection class. This collection object provides access to a variety of Content objects.

Data Type

[ContentCollection](#)

Syntax

contentsvalue = [objectreference].Contents

[objectreference].Contents = contentsvalue

Legal values

When this property is populated, it contains an instance of the ContentCollection class. You cannot set the value of this property but you can access the objects contained within this collection object.

Usage

When accessed through the Foundry property on a Division object, the collection object in this property provides access to all the Content objects contained in that Division object.

When accessed through the AppFoundry property on the WPAApplication object, this collection object provides access to all the Content objects contained in the Word Pro Clipboard.

When accessed through the TempFoundry property on the WPAApplication object, this collection object provides access to all the Content objects placed in TempFoundry by WordPro or a script.

When accessed through the Foundry property on the WPAApplication object, this collection object provides access to all the Content objects contained in the currently active Division object.

This property is not used in the Foundry property on the TextDocument object.

For more information about collection classes, see [Overview: WordPro LotusScript CollectionClasses](#).

```
'Example: BaseTable property
' This example creates a table, then prints the default column width of the
' table to the LotusScript output panel.
.CreateTable
TwipColWid = .BaseTable.DefColWidth
Print "The default column width is " & TwipColWid/1440 & " inches."
```

```
'Example: BeginChange method
' This example creates a table with 5 columns and 5 rows. Several background
' table cell properties are changed all at one using the BeginChange and
' EndChange methods.
' RUNTIME DEPENDENCIES: You must have a document open for this script to work.
```

```
Dim ExampleTable As Table
Dim ExampleCell As CellLayout

.CreateTable False, "Default Table", 5,5
Set ExampleTable = .Table
ExampleTable.CellLayout(1,0).GotoLayout

Set ExampleCell = ExampleTable.CellLayout(1,0)
```

.BeginChange

```
ExampleCell.Content.InsertText "Hello"
With ExampleCell.BackGround
    .Pattern = $LtsFillSolid
    .Color.Red = 255
    .Color.Blue = 194
    .Color.Green = 255
    .Color.Override = $LwpColorOverrideRgb
    .BackColor.Red = 65
    .BackColor.Blue = 176
    .BackColor.Green = 0
End With

.EndChange
```

```
'Example: BetweenLines property
'This example inserts some text into the current document, then places
'a red line between the paragraphs.

' Insert some text, then select it.
.Text.InsertText "This is an example paragraph.", True
.Text.InsertText "This is another example paragraph.", True
.Text.Backward $LwpNavigateObjectTypeParagraph , 2
' Set up the borders
.Text.ParagraphBorder.Betweenlines.LinePlacement = &Hf
.Text.ParagraphBorder.Betweenlines.AllBorders.Pattern = $LtsBorderPatternSolid
.Text.ParagraphBorder.Betweenlines.AllBorders.WidthInTwips = 20 ' 1 point line
' make the line red
.Text.ParagraphBorder.Betweenlines.AllBorders.Color.Red = 255
.Text.ParagraphBorder.Betweenlines.AllBorders.Color.Blue = 0
.Text.ParagraphBorder.Betweenlines.AllBorders.Color.Green = 0
.Text.ParagraphBorder.Betweenlines.AllBorders.Color.Override = $LwpColorOverrideRgb
'Put 1/10 inch between the text and the line
.Text.ParagraphBorder.MarginBetween = 144
'Make the line as long as the paragraph
.Text.ParagraphBorder.TypeBetween = $LwpParaBorderWidthText
```

'Example: BinNameFromNumber method

'This example retrieves the name of the first sheet feeder bin, then

'displays a message box with the printer name and the bin name.

```
PrinterName = .ActiveDocument.PrintManager.PrinterName
```

```
BinName = .ActiveDocument.PrintManager.BinNameFromNumber(1)
```

```
MessageBox "The first available bin for the " & PrinterName & " printer is " & Binname  
& ".", MB_OK, "Example Script"
```

'Example: Blue property

'This example sets the color of text to blue, then types some text.

```
.Text.Font.FontColor.Red = 0
```

```
.Text.Font.FontColor.Blue = 255
```

```
.Text.Font.FontColor.Green = 0
```

```
.Text.Font.FontColor.Override = $LwpColorOverrideRgb
```

```
.Type "This is blue text.[Enter]"
```


'Example: BodyOnly property

'This example types some text, indents it 1/2 inch. Because the BodyOnly property
'is set to True, the indentation will only occur if the text is in the main
'body of the text.

.Text.InsertText "This text will be indented 1/2 inch."

.Text.Indent.All = 720 ' 720 twips = 1/2 inch.

.Text.Indent.**BodyOnly** = True

'Example: Bold method

' This example first inserts sample text in the current document and selects
' the paragraph. The script then uses the Bold method to toggle the bold
' attribute.

' RUNTIME DEPENDENCIES: You must have a document open with selected text
' for this script to work.

.Text.InsertText "This is some sample text."
.SelectParagraph

.Bold

```
'Example: Bold property
'This example enables the Bold attribute, types text, and disables the attribute.
.Text.Font.Bold = True
.Text.InsertText "This is bolded text."
.Text.Font.Bold = False
```

'Example: BorderOffset property

'This example places lines around the page, then sets the location of the
'lines to the middle of the margins, and the offset to 1/2 inch.

```
.Page.Layout.RightPage.BorderLines.LinePlacement = &Hf  
.Page.Layout.RightPage.BorderLines.AllBorders.Pattern = $LwpBorderPatternWavy  
.Page.Layout.RightPage.BorderLines.AllBorders.WidthInTwips = 80  
.Page.Layout.RightPage.BorderOffset = 720  
.Page.Layout.RightPage.LineLocation = 1
```

'Example: BottomExternalMargin property

'This example creates a frame, then changes the padding around the frame's
'border to 1/4 inch (360 twips).

```
.NewFrame 4320, 4320, 3387, 1992, "Default Frame"
```

```
.Frame.Layout.LeftExternalMargin = 360
```

```
.Frame.Layout.TopExternalMargin = 360
```

```
.Frame.Layout.RightExternalMargin = 360
```

```
.Frame.Layout.BottomExternalMargin = 360
```

```
'Example: Breaks property
'This example sets options for page breaks before and after paragraphs.
.Type "This is some text in the on the first page.[Enter]"
.Text.Breaks.IsPageBreakBefore = True
.Type "There is a page break before this paragraph[Enter]"
.Text.Breaks.IsPageBreakBefore = False
.Text.Breaks.IsPageBreakAfter = True
.Type "There is a page break after this paragraph[Enter]"
.Text.Breaks.IsPageBreakAfter = False
.Type "This is text on the third page."
```

```
'Example: BringFrameToFrontOne method
' This example creates two frames and changes the order of the layering
' for the two frames.
' RUNTIME DEPENDENCIES: You must have a document open for this script to work.

.NewFrame 3285, 1200, 1575, 1830
.Frame.Layout.Background.Color.Red = 82
.Frame.Layout.Background.Color.Blue = 239
.Frame.Layout.Background.Color.Green = 145
.Frame.Layout.Background.Override = $LwpColorOverrideRgb
.Frame.Anchor $LwpAnchorWhereLayout, $LwpConditionTypeOnlyspecificpage,
$LwpRelativeTypeLytParent

.Deselect

.NewFrame 5285, 2200, 1575, 1830
.Frame.Anchor $LwpAnchorWhereLayout, $LwpConditionTypeOnlyspecificpage,
$LwpRelativeTypeLytParent
.Frame.Layout.Background.Color.Red = 182
.Frame.Layout.Background.Color.Blue = 139
.Frame.Layout.Background.Color.Green = 45
.Frame.Layout.Background.Override = $LwpColorOverrideRgb

MessageBox "Click OK to send frame to back. ",MB_OK,"Example Script"
.SendFrameToBack

MessageBox "Click OK to bring frame to front. ",MB_OK,"Example Script"
.BringFrameToFrontOne
```

```
'Example: BringFrameToFront method
' This example creates two frames and changes the order of the layering
' for the two frames.
' RUNTIME DEPENDENCIES: You must have a document open for this script to work.

.NewFrame 3285, 1200, 1575, 1830
.Frame.Layout.Background.Color.Red = 82
.Frame.Layout.Background.Color.Blue = 239
.Frame.Layout.Background.Color.Green = 145
.Frame.Layout.Background.Color.Override = $LwpColorOverrideRgb
.Frame.Anchor $LwpAnchorWhereLayout, $LwpConditionTypeOnlyspecificpage,
$LwpRelativeTypeLytParent

.Deselect

.NewFrame 5285, 2200, 1575, 1830
.Frame.Anchor $LwpAnchorWhereLayout, $LwpConditionTypeOnlyspecificpage,
$LwpRelativeTypeLytParent
.Frame.Layout.Background.Color.Red = 182
.Frame.Layout.Background.Color.Blue = 139
.Frame.Layout.Background.Color.Green = 45
.Frame.Layout.Background.Color.Override = $LwpColorOverrideRgb

MessageBox "Click OK to send frame to back. ",MB_OK,"Example Script"
.SendFrameToBack

MessageBox "Click OK to bring frame to front. ",MB_OK,"Example Script"
.BringFrameToFront
```


'Example: CalculateSmartLevels method

' This example updates the SmartLevels for the currently active division.

' RUNTIME DEPENDENCIES: You must have a document open for this script to work.

.CalculateSmartLevels

```

'Example: CanRepeatClickHere method
'This example inserts a Click Here block, then tests to see if the click here
'can be repeated. The first test is done outside the click here, while the
'second is done inside the click here.

MyClick = .InsertClickHere()
.Division.Foundry.ClickHeres(MyClick).Prompt.Clear
.Division.Foundry.ClickHeres(MyClick).Prompt.InsertText "Click here to type Text",
False, $LwpTextTypeNative
.Division.Foundry.ClickHeres(MyClick).MirrorName = ""
.Division.Foundry.ClickHeres(MyClick).Action = 1
.Division.Foundry.ClickHeres(MyClick).HelpText = "Click here to type Text"
.Division.Foundry.ClickHeres(MyClick).UsesHelp = False
.Division.Foundry.ClickHeres(MyClick).TabOrder = 1
.Division.Foundry.ClickHeres(MyClick).TabExits = True
.Division.Foundry.ClickHeres(MyClick).ReturnExits = False
.Division.Foundry.ClickHeres(MyClick).Name = MyClick
.Division.Foundry.ClickHeres(MyClick).AllowListEdit = True
.Division.Foundry.ClickHeres(MyClick).AllowListMultiValues = False
.Division.Foundry.ClickHeres(MyClick).RemoveNamedProperty
.Division.Foundry.ClickHeres(MyClick).SetNamedProperty "Collect", "Off"
.Division.Foundry.ClickHeres(MyClick).SetNamedProperty "Required", "Off"
.Division.Foundry.ClickHeres(MyClick).SetNamedProperty "Notes/FX", "Off"
.UpdateUI

'Check to see if we can repeat the click here we just created
If .Text.CanRepeatClickHere(MyClick) Then
    MessageBox "OK to repeat the Click Here Block at this point", MB_OK, "Example
Script"
Else
    MessageBox "Problems if you repeat the Click Here Block at this point", MB_OK,
"Example Script"
End If

'Now, move inside the click here we want to repeat, and try again.
.GoTtoClickHere (MyClick)
If .Text.CanRepeatClickHere(MyClick) Then
    MessageBox "OK to repeat the Click Here Block at this point", MB_OK, "Example
Script"
Else
    MessageBox "Problems if you repeat the Click Here Block at this point", MB_OK,
"Example Script"
End If

```

```
'Example: Caption property
Dim FileMenu As MenuItem
Set FileMenu = .ApplicationWindow.LWPMenuBar.Items("&File")
' Set a counter
Count% = 1

' Cycle through all options on the File menu
Forall Item In FileMenu.Items

' Display some info on each menu option
    Print Count% , "Name -", Item.Caption
    Print , "Action -", Item.Action
    Count% = Count% + 1
End Forall
```

```
'Example: CascadeWindow method
' This example creates two new documents based on the 'DEFAULT.MWP'
' SmartMaster.
' The script then prompts you to cascade the new windows.

.NewDocument , , "DEFAULT.MWP", ,
.NewDocument , , "DEFAULT.MWP", ,
MessageBox "Click OK to cascade the new windows.", MB_OK, "Example Script"

.CascadeWindow
```

'Example: Cascade method

' This example cascades any document windows.

' RUNTIME DEPENDENCIES: You must have a document open for this script to work.

' Restore will restore the main Word Pro window

' Cascade the open document windows within Word Pro

.ApplicationWindow.Cascade

```
'Example: CellLayout method
' This example creates a table with 5 columns and 5 rows then inserts text and
' changes the background color for row 1, column 0.
' RUNTIME DEPENDENCIES: You must have a document open for this script to work.

Dim ExampleTable As Table
Dim ExampleCell As CellLayout

.CreateTable False, "Default Table", 5,5
Set ExampleTable = .Table
Set ExampleCell = ExampleTable.CellLayout(1,0)
ExampleCell.GotoLayout

ExampleCell.Content.InsertText "Hello"
With ExampleCell.BackGround
    .Pattern = $LtsFillSolid
    .Color.Red = 255
    .Color.Blue = 194
    .Color.Green = 255
    .Color.Override = $LwpColorOverrideRgb
    .BackColor.Red = 65
    .BackColor.Blue = 176
    .BackColor.Green = 0
End With
```

```
'Example: CellRevert method
' This example creates a table with 5 columns and 5 rows then inserts text and
' changes the background color for row 1, column 0. The background color is
' then reverted after the message box is closed.
' RUNTIME DEPENDENCIES: You must have a document open for this script to work.
```

```
Dim ExampleTable As Table
Dim ExampleCell As CellLayout
```

```
.CreateTable False, "Default Table", 5,5
Set ExampleTable = .Table
ExampleTable.CellLayout(1,0).GotoLayout
```

```
Set ExampleCell = ExampleTable.CellLayout(1,0)
```

```
ExampleCell.Content.InsertText "Hello"
With ExampleCell.BackGround
    .Pattern = $LtsFillSolid
    .Color.Red = 255
    .Color.Blue = 194
    .Color.Green = 255
    .Color.Override = $LwpColorOverrideRgb
    .BackColor.Red = 65
    .BackColor.Blue = 176
    .BackColor.Green = 0
End With
```

```
Messagebox "Click OK to revert cell color change." ,MB_OK,"Example Script"
```

```
.CellRevert
```

```
'Example: Cell property
' This example creates a table, then prints the width of the active cell
' to the LotusScript output panel.

.CreateTable
TwipCellWid = .Cell.Width
Print "The cell width is " & TwipCellWid/1440 & " inches."
```



```
'Example: Changed property
'This example tells the user if the current document has been edited since
'the last save.

If .ActiveDocument.Changed = True then
    MessageBox "This document has been edited since it was last saved.", MB_OK,
    "Example Script"
Else
    MessageBox "This document has not changed since it was last saved.", MB_OK,
    "Example Script"
Endif
```

'Example: ChangeSmartMaster method

' This example changes the current division's SmartMaster to BUSPLAN.MWP.

' RUNTIME DEPENDENCIES: You must have a document open for this script to work.

SmartMasterPath = .ApplicationWindow.UserInterfacePrefs.StylePath

SmartMaster = SmartMasterPath & "\" & "BUSPLAN.MWP"

.ChangeSmartMaster SmartMaster, "Lotus Word Pro", ""

```
'Example: Checked property
' This example toggles whether or not 'Example Menu' item is checked
' or unchecked. This script run from the 'NewItem' example and is not
' intended to be run stand-alone.
' RUNTIME DEPENDENCIES: You must have a document open for this script to work.
```

```
Dim NewMenu As MenuItem
Dim MenuName as String
```

```
MenuName = "&Example Menu"
Set NewMenu =.ApplicationWindow.LwpMenuBar.Items.Item("&File")
```

```
If NewMenu.Items(MenuName).Checked Then
    NewMenu.Items(MenuName).Checked = False
Else
    NewMenu.Items(MenuName).Checked = True
End If
```

'Example: ClassName property

'This example assigns the category 'Proposal' to the current document.

' and the category 'Title Page' to the current division.

.ActiveDocument.DivisionInfo.**ClassName** = "Proposal"

.Division.DivisionInfo.**ClassName** = "Title Page"

'Example: CleanScreenOnStartUp property

'This example asks the user whether to start Word Pro in Clean Screen view,

'and then sets the appropriate option.

```
stat = MessageBox ("Do you want to start in Clean Screen view?", 36, "Example Script")
```

```
If stat = 6 Then ' user said yes
```

```
    .ApplicationWindow.UserInterfacePrefs.CleanScreenOnStartUp = True
```

```
Else
```

```
    .ApplicationWindow.UserInterfacePrefs.CleanScreenOnStartUp = False
```

```
End If
```

'Example: ClearAll method

' This example inserts 10 right aligned tabs with leader dots which are spaced
' one half inch apart. After the message box is closed all of the tabs are
' removed.

' RUNTIME DEPENDENCIES: You must have a document open for this script to work.

Dim TabNumber As Integer

For TabNumber = 1 To 10

 stat = .Text.TabRack.InsertOne(720 * TabNumber, \$LwpTabTypeLeft, \$LwpTabLeaderDot,
 \$LwpTabRelativeLeft, 32)

 .Type TabNumber & "[Tab]"

Next

.Text.TabRack.ClearAll

```
'Example: ClearDivisionList method
' This example prints the current division to the default printer.
' RUNTIME DEPENDENCIES: You must have a document open for this script to work.
```

```
Dim CurrentDivName As String
```

```
CurrentDivName = .Division.Name
```

```
.ActiveDocument.GetPageRange $LwpPresentationTypeLayout, 100
```

```
' make sure there are no divisions currently selected to print
```

```
.ActiveDocument.PrintSettings.ClearDivisionList
```

```
.ActiveDocument.PrintSettings.AddDivisionToPrint CurrentDivName
```

```
.ActiveDocument.PrintSettings.SelectedPages = "1-9999"
```

```
.ActiveDocument.PrintSettings.PrintRange = $LwpPrintRangeSelectedDivisions
```

```
.ActiveDocument.PrintSettings.Copies = 1
```

```
.ActiveDocument.PrintSettings.PrintPagesFrom = 1
```

```
.ActiveDocument.PrintSettings.PrintPagesTo = 1
```

```
.ActiveDocument.PrintSettings.PrintPageType = $LwpPrintPageEvenAndOddPages
```

```
.ActiveDocument.PrintSettings.Collate = False
```

```
.ActiveDocument.PrintSettings.OutputToFile = False
```

'Example: ClearParaRevisionTags method

' This example removes paragraph revision tags from the active document.

' RUNTIME DEPENDENCIES: You must have a document open for this script to work.

.ClearParaRevisionTags


```
'Example: ClearSplits method
' This example sets a split view. It displays the same document in two
' windows. In the top window it displays the document at page width.
' In the bottom window, it displays the document in a multi-page view showing
' the first seven pages.
' RUNTIME DEPENDENCIES: You must have a document open for this script to work.

' clear any current splits
.ActiveDocWindow.WinViewPrefs.ClearSplits

' the first window will display seven pages left to right
.ActiveDocWindow.WinViewPrefs.NumCols = 7
.ActiveDocWindow.WinViewPrefs.IsInDraft = False
.ActiveDocWindow.WinViewPrefs.ViewType = &H40

' now set a new window that will take 66% of the doc window
.ApplicationWindow.UserInterfacePrefs.VerticalSplitWindow = True
.ApplicationWindow.UserInterfacePrefs.SplitPercentage = 66

' open the new window
.NewWindow

' change the new window to display page width
.ApplicationWindow.UserInterfacePrefs.VerticalSplitWindow = False
.ActiveDocWindow.WinViewPrefs.IsInDraft = False
.ActiveDocWindow.WinViewPrefs.ViewType = &H100
```

```
'Example: Clear method
' This example inserts some text into the current document. After the message
' box is closed all text is cleared.
' RUNTIME DEPENDENCIES: You must have a document open for this script to work.

Dim x as Integer
For x = 1 To 5
    .Text.InsertText "Some Text " & x
Next
MessageBox "Click OK to remove all text.",MB_OK,"Example Script"
.Text.Clear $LwpClearWhatDefault
```

```
'Example: ClickHerePrompts property
'This example asks the user whether to display unfilled Click Here Block prompts
'in the current document, and then sets the appropriate option.

stat = MessageBox ("Do you want to show Click Here Block prompts?", 36, "Example
Script")
If stat = 6 Then ' user said yes
    .ActiveDocument.DocOptions.ClickHerePrompts = 1
Else
    .ActiveDocument.DocOptions.ClickHerePrompts = 0
End If
```

```
'Example: ClickHeres property
' This example inserts a ClickHere block in the current document and assigns
' the return value (the ClickHere ID) to a variable (NewClickHereId).
' The script then stores the new ClickHere in a variable (NewClickHere) and
' inserts some text in it.
' The script then prints the names and values for all ClickHeres in the
' current division to the Script Editor Output panel.
' RUNTIME DEPENDENCIES: You must have a document open for this script to work.

Dim NewClickHereId as String
Dim NewClickHere as ClickHere

NewClickHereId = .InsertClickHere()
Set NewClickHere = .Division.Foundry.ClickHeres(NewClickHereId)
NewClickHere.Prompt.InsertText "Click Here to type text "
NewClickHere.InsertText "Some text for the click here."

Forall Clicks in .Division.Foundry.ClickHeres
    Print Clicks.Name &" -- " & Clicks.GetText($LWPGetObjectParagraph, False)
End Forall
```

'Example: CloseAll method

' This example closes all open files including untitled files.

' RUNTIME DEPENDENCIES: You must have a document open for this script to work.

.CloseAll \$LwpCloseFileIfLastdocOpenUntitled

'Example: CloseDocWindow method

' This example creates a new document window based upon the current document
' and displays a message box prompting you to close the window.
' After you click OK, the script closes the new window.
' RUNTIME DEPENDENCIES: You must have a document open for this script to work.

.NewWindow

MessageBox "Click OK to close the current window.",MB_OK,"Example Script"

.CloseDocWindow

```
'Example: CloseMergeDataFile method
' This example merges data for the current merge document.
' RUNTIME DEPENDENCIES: You must have a document open which has been assigned
' to a merge data file and which has inserted merge fields for this script to work.

Dim stat As Integer

' Set up to merge and view
.ApplicationWindow.ActiveDocument.MergeOptions.Options = &H2
.ApplicationWindow.ActiveDocument.MergeOptions.MergeStepNumber = $LwpMergeStepNumber3
.MergeStart

Do
    stat = .Merge($LwpMergeActionNextRecord)
    .Merge $LwpMergeActionMergeOne
Loop Until stat = False

.Merge $LwpMergeActionClose

.CloseMergeDataFile
```

'Example: CloseObject method

'This example inserts a comment note in the document, then closes the note.

.InsertNote

.Text.InsertText "Some text for the note."

MessageBox "Click OK to close the note.", MB_OK, "Example Script"

.Type "[Esc]" ' get back to the main text stream

.Text.**CloseObject**

'Example: Close method

' This example completely shuts down Word Pro while asking to save any changes.

.ApplicationWindow.Close

'Example: Collapsible property
'This example inserts two markers into the current document. The contents
'of the markers is inserted into the document at the end.
'For the first marker, the additional typed text is included in the marker
'because the Collapsible property is not set until after the text is inserted.
'The text is not included in the second marker because the Collapsible
'property is set immediately after creating the marker.

```
.Text.InsertText "This is marker 1 "  
.Text.Backward $LwpNavigateObjectTypeWord, 3  
.Text.Shade $LwpLocationTypeWord, $LwpNavigateDirectionRight, 2  
Mark1 = .Text.Mark ($LwpMarkerTypeDefault)  
.Deselect  
.Text.InsertText " This is additional text. ", True  
.Division.Foundry.Markers(Mark1).Collapsible = True  
.Text.InsertText "This is marker 2."  
.Text.Backward $LwpNavigateObjectTypeWord, 3  
.Text.Shade $LwpLocationTypeWord, $LwpNavigateDirectionRight, 1  
Mark2 = .Text.Mark ($LwpMarkerTypeDefault)  
.Text.Forward $LwpNavigateObjectTypeWord, 1  
.Division.Foundry.Markers(Mark2).Collapsible = True  
.Deselect  
.Text.InsertText " This is additional text.", True  
Mark1Text = .Division.Foundry.Markers(Mark1).GetMarkedText  
.Text.InsertText "The marked text for Marker 1 is: " & Mark1Text, True  
Mark2Text = .Division.Foundry.Markers(Mark2).GetMarkedText  
.Text.InsertText "The marked text for Marker 2 is: " & Mark2Text, True
```

```
'Example: Collate property
'This example sets the current document to print two copies, then
'asks the user whether to collate the output. It then sets the appropriate
'option, and prints the document.

.ActiveDocument.PrintSettings.Copies = 2
stat = MessageBox ("Do you want to collate printed output?", 36, "Example Script")
If stat = 6 Then ' user said yes
    .ActiveDocument.PrintSettings.Collate = True
Else
    .ActiveDocument.PrintSettings.Collate = False
End If
.Print
```

'Example: ColorOverride property

'This example creates a table and inserts a negative number in the first
'cell. The user is then given the option of changing the color for
'negative numbers.

```
.CreateTable False, "Default Table", 2, 2
```

```
.Text.InsertText "-1", False
```

```
.Table.CurrentCell.NumericFormat.Negative.Prefix = ""
```

```
.Table.CurrentCell.NumericFormat.Negative.Suffix = ""
```

```
stat = MessageBox ("Would you like negative numbers to appear in red?", 36, "Example  
Script")
```

```
If stat = 6 Then 'user said yes
```

```
    .Table.CurrentCell.NumericFormat.Negative.ColorOverride = True
```

```
    .Table.CurrentCell.NumericFormat.Negative.Color.Red = 255
```

```
    .Table.CurrentCell.NumericFormat.Negative.Color.Blue = 0
```

```
    .Table.CurrentCell.NumericFormat.Negative.Color.Green = 0
```

```
    .Table.CurrentCell.NumericFormat.Negative.Color.Override = $LwpColorOverrideRgb
```

```
Else
```

```
    .Table.CurrentCell.NumericFormat.Negative.ColorOverride = False
```

```
End If
```

Word Pro: BetweenLines class

This language element has not yet been defined. You can check for a newer Help file on the Lotus web site.

Base Classes

[BorderLines](#)

Derived Classes

None.

Contained by

[ParagraphBorder](#) in the [BetweenLines](#) Property

Usage

Word Pro: FrameCaptionOptions class

This language element has not yet been defined. You can check for a newer Help file on the Lotus web site.

Base Classes

[BaseObject](#)

Derived Classes

None.

Contained by

[TextDocument](#) in the [FrameCaptionOptions](#) Property

Usage

Word Pro: HTMLOptions class

This language element has not yet been defined. You can check for a newer Help file on the Lotus web site.

Base Classes

[BaseObject](#)

Derived Classes

None.

Contained by

[UserInterfacePrefs](#) in the [HTMLOptions](#) Property

Usage

Word Pro: OleControl class

This class has a single property used for storing the complete hierarchical name of an OLE control. See Usage for details.

Note You cannot access embedded OLE controls from this class.

Base Classes

This class is the only stand-alone class in Word Pro. It is not derived from any other class and therefore does not inherit any class members.

Derived Classes

None.

Contained by

Objects created from the OleControl class are not accessible through any property. More information about accessing the hierarchical name of an OLE control is provided below under Usage.

Usage

The only use for this class is to provide access to the complete hierarchical name of an embedded OLE control. This name is stored in the WordProObjectName property. When you embed an OLE control in a Word Pro document, Word Pro attaches the WordProObjectName property to that OLE control. The only way to access the property is through the object returned in the Source argument of an OLE control's event script.

For example, the event script below is for the Click event on a LotusCommandButton control.

```
Sub Click(Source as Lotuscommandbutton)
    MsgBox Source.WordProObjectName
End Sub
```

When you click the LotusCommandButton which contains this event script, Word Pro passes the OLE control object to this script through the Source argument. In addition to the usual properties and methods provided by the developer of the OLE control, Word Pro provides the WordProObjectName property.

Note This name is for informational purposes only. You do not need an OLE control's hierarchical name to access the control.

To access and script an OLE control without using the event script, you must use the Object property on the OleObject class.

Word Pro: OutlineStyleSequence class

The arrangement of styles in an outline.

Base Classes

BaseObject

Derived Classes

None.

Contained by

Usage

Word Pro: OutSeqItem class

An item in an outline sequence.

Base Classes

BaseObject

Derived Classes

None.

Contained by**Usage**

Word Pro: PageContainer class

The container object for pages. This object only exists for one page at a time and only when there is a page with in the focus. When a PageContainer object is present, it is stored in the Page property on the WPApplication object.

Base Classes

BaseObject\BaseContainer

Derived Classes

[SubPageContainer](#)

[SuperPageContainer](#)

Contained by

[WPApplication](#) in the [Page](#) Property

Usage

The primary use for a PageContainer object is to provide quick and easy access to the PageLayout object for the currently active page. A PageContainer object always represents the page that currently has the focus. Therefore, if you assign a PageContainer object to a variable, you can use that variable to access the currently active page. However, you must remember that the page referenced by the variable changes as the focus moves from one page to another. This is because the variable references the PageContainer object, and the PageContainer object always represents the page that has the focus. If there is no page with in the focus, there is no PageContainer object. Therefore, a variable that stores a PageContainer object has a null value whenever the focus does not contain a page. There is never more than one PageContainer object at any given time.

For more information about container objects, see [BaseContainer](#).

Word Pro: PageLayoutCollection class

A collection of page layout objects in the foundry of a specific division.

Base Classes

BaseObject\BaseCollection

Derived Classes

None.

Contained by

Foundry in the Pages Property

Foundry in the PageStyles Property

Usage

Use this collection to access any of the page layout objects in the foundry of a specific division.

Word Pro: PageLayout class

A page layout for a PageContainer object. This class inherits most of its members from the Layout class.

Base Classes

BaseObject\Layout

Derived Classes

None

Contained by

Usage

The PageLayout class provides you with a way to access and modify the format and appearance of PageLayout objects within your document.

Since the PageLayout class is derived from the Layout class, PageLayout objects can be stored within properties of the Layout type. For example, the Layout property within the PageContainer class is of the Layout type. However, this property often stores objects of the PageLayout type. The Layout property is implemented in this way, so that objects of other derived layout class types can be stored there as well.

PageLayout objects within a division are stored together in a collection. You can use the collection to access all PageLayout objects in the collection, or you can reference a particular PageLayout object in the collection. For example, by using the PageLayouts collection, you can modify each PageLayout object in the collection to have a specific top margin value. For more information on how to work with collections, see [Overview: Word Pro LotusScript Collection Classes](#).

At many locations within your document, multiple layouts are available. For example, your cursor may be in a frame on a page. In this case, the frame and the page both have associated layout objects. These layout objects may be combined with other objects into related groups, known as containers. For more information on containers and their associated layouts, see the Help topic titled [Word Pro: BaseContainer class](#).

Word Pro: ParagraphBackground class

This language element has not yet been defined. You can check for a newer Help file on the Lotus web site.

Base Classes

[BaseObject](#)

Derived Classes

None.

Contained by

[ClickHere](#) in the [ParagraphBackground](#) Property

[ParagraphStyle](#) in the [ParagraphBackground](#) Property

[Text](#) in the [ParagraphBackground](#) Property

[TextMarker](#) in the [ParagraphBackground](#) Property

Usage

Word Pro: ParagraphBorder class

The border around a paragraph object.

Base Classes

BaseObject

Derived Classes

None.

Contained by

[ClickHere](#) in the [ParagraphBorder](#) Property

[ParagraphStyle](#) in the [ParagraphBorder](#) Property

[Text](#) in the [ParagraphBorder](#) Property

[TextMarker](#) in the [ParagraphBorder](#) Property

Usage

Word Pro: ParagraphStyleCollection class

A collection of paragraph style objects in the foundry of a specific division.

Base Classes

BaseObject\BaseCollection

Derived Classes

None.

Contained by

Foundry in the ParagraphStyles Property

Usage

Use this collection to access any of the paragraph style objects in the foundry of a specific division.

Word Pro: ParagraphStyle class

The style assigned to a paragraph.

Base Classes

BaseObject

Derived Classes

None.

Contained by

[ClickHere](#) in the [ParagraphStyle](#) Property

[Text](#) in the [ParagraphStyle](#) Property

[TextMarker](#) in the [ParagraphStyle](#) Property

Usage

Word Pro: ParallelColsCollection class

A collection of parallel column objects in the foundry of a specific division.

Base Classes

BaseObject\BaseCollection

Derived Classes

None.

Contained by

Foundry in the ParallelColumns Property

Usage

Use this collection to access any of the parallel column objects in the foundry of a specific division.

Word Pro: ParallelColsContainer class

The container object for parallel columns. This object only exists for one parallel columns object at a time, and only when there are parallel columns with in the focus. When parallel columns are present, the ParallelColsContainer is stored in the TableContainer property on the WPApplication object.

Base Classes

BaseObject\BaseContainer\TableContainer

Derived Classes

None.

Contained by

Not contained in any property.

Usage

The primary use for a ParallelColsContainer object is to provide quick and easy access to the CellLayout object for the currently active parallel columns. A ParallelColsContainer object always represents the parallel columns that currently have the focus. Therefore, if you assign a ParallelColsContainer object to a variable, you can use that variable to access the currently active parallel columns. However, you must remember that the parallel columns referenced by the variable change as the focus moves from one parallel columns object to another. This is because the variable references the ParallelColsContainer object, and the ParallelColsContainer object always represents the parallel columns that have the focus. If there are no parallel columns with in the focus, there is no ParallelColsContainer object. Therefore, a variable that stores a ParallelColsContainer object has a null value whenever the focus does not contain parallel columns. There is never more than one ParallelColsContainer object at any time.

For more information about container objects, see [BaseContainer](#).

Word Pro: ParallelColumns class

A parallel (newspaper) column object in a document.

Base Classes

BaseObject\Content\BaseTable

Derived Classes

Glossary

Contained by

WPApplication in the ParallelColumns Property

Usage

Word Pro: PowerFieldCollection class

A collection of power field objects in the foundry of a specific division.

Base Classes

BaseObject\BaseCollection

Derived Classes

None.

Contained by

Foundry in the PowerFields Property

Usage

Use this collection to access any of the power field objects in the foundry of a specific division.

Word Pro: PowerField class

A power field object in a document.

Base Classes

BaseObjectMarker

Derived Classes

None.

Contained by

Usage

Word Pro: Preferences class

Allows you to customize Word Pro by setting various options for file saving, file location, default files, and general and personal options.

Base Classes

BaseObject

Derived Classes

None.

Contained by

WPApplication in the Preferences Property

Usage

Word Pro: Presentation class

Base Classes

BaseObject

Derived Classes

None.

Contained by

BaseContainer in the Presentation Property

Usage

Word Pro: PrintManager class

A tool to manage document printing.

Base Classes

BaseObject

Derived Classes

None.

Contained by

TextDocument in the PrintManager Property

Usage

Word Pro: PrintSettings class

The print settings displayed in the Print and Print Options dialog boxes.

Base Classes

BaseObject

Derived Classes

None.

Contained by

Document in the PrintSettings Property

Usage

Use this class to access the print settings for a specific document.

Word Pro: RelativeIndent class

Base Classes

BaseObject\Indent

Derived Classes

None.

Contained by

[ClickHere](#) in the [RelativeIndent](#) Property

[ParagraphStyle](#) in the [RelativeIndent](#) Property

[Text](#) in the [RelativeIndent](#) Property

[TextMarker](#) in the [RelativeIndent](#) Property

Usage

Word Pro: ReviewVersions class

Allows you to review versions of a Word Pro document.

Base Classes

BaseObject

Derived Classes

None.

Contained by

ApplicationWindow in the ReviewVersions Property

Usage

Word Pro: RevisionDisplay class

Allows you to display the Revision tool in a document.

Base Classes

BaseObject

Derived Classes

None.

Contained by

Preferences in the RevisionDisplay Property

Usage

Word Pro: Revision class

The Revision tool in the Word Pro application.

Base Classes

BaseObject

Derived Classes

None.

Contained by

[ClickHere](#) in the [RevisionMark](#) Property

[Text](#) in the [RevisionMark](#) Property

[TextMarker](#) in the [RevisionMark](#) Property

Usage

Word Pro: RowContainer class

The container object for a table row. This object only exists for one row at a time and only when there is a table with in the focus.

Base Classes

BaseObject\BaseContainer

Derived Classes

None.

Contained by

Not contained in a property of any object.

Usage

The primary use for a RowContainer object is to provide quick and easy access to the RowLayout object for the currently active row. A RowContainer object always represents the row that currently has the focus. Therefore, if you assign a RowContainer object to a variable, you can use that variable to access the currently active row. However, you must remember that the row referenced by the variable changes as the focus moves from one row to another. This is because the variable references the RowContainer object, and the RowContainer object always represents the row that has the focus. If there is no row with in the focus, there is no RowContainer object. Therefore, a variable that stores a RowContainer object has a null value whenever the focus does not contain a row. There is never more than one RowContainer object at any time.

For more information about container objects, see [BaseContainer](#).

Word Pro: RowGroupLayout class

This language element is not yet defined.

Base Classes

unkown

Derived Classes

None.

Contained by

None.

Usage

Word Pro: RowLayoutCollection class

A collection of row layout objects in the foundry of a specific division.

Base Classes

BaseObject\BaseCollection

Derived Classes

None.

Contained by

Foundry in the Rows Property

Usage

Use this collection to access any of the row layout objects in the foundry of a specific division.

Word Pro: RowLayout class

The layout for a row in a table object.

Base Classes

BaseObjectLayout

Derived Classes

None.

Contained by

BaseTable in the CurrentRow Property

WPApplication in the CurrentRow Property

Usage

Word Pro: RubyContainer class

Not implemented in the US English version of Word Pro.

Base Classes

BaseObject\BaseContainer\FrameContainer

Derived Classes

None.

Contained by

Usage

Word Pro: RubyLayoutCollection class

A collection of ruby layout objects in the foundry of a specific division.

Base Classes

BaseObject\BaseCollection

Derived Classes

None.

Contained by

Foundry in the RubyLayouts Property

Usage

Use this collection to access any of the ruby layout objects in the foundry of a specific division.

Word Pro: RubyLayout class

The ruby layout for a ruby object in a division.

Base Classes

BaseObjectLayout

Derived Classes

None.

Contained by

Usage

Word Pro: RubyMarker class

A marker for a ruby object in a division.

Base Classes

BaseObjectMarker

Derived Classes

None.

Contained by

Usage

Word Pro: Ruler class

A horizontal or vertical ruler object that indicates tab settings, indents, margins, and columns.

Base Classes

BaseObject

Derived Classes

None.

Contained by

ApplicationWindow in the HorzRuler Property

ApplicationWindow in the VertRuler Property

Usage

Word Pro: ScriptDataSet class

An abstract class which defines properties and methods that are common to all data set objects in Word Pro.

Base Classes

BaseObject

Derived Classes

WPDataSet

Contained by

Usage

Data set objects provide you with a way to create a set of variables that can be stored with a document. Each of the items within the data set has a name and a value associated with it.

Word Pro: SectionCollection class

A collection of section objects in the foundry of a specific division.

Base Classes

BaseObject\BaseCollection

Derived Classes

None.

Contained by

Foundry in the Sections Property

Usage

Use this collection to access any of the section objects in the foundry of a specific division.

Word Pro: SectionTabs class

The divider tabs used to indicate a section or a division.

Base Classes

BaseObject

Derived Classes

None.

Contained by

ApplicationWindow in the SectionTabs Property

Usage

Use the methods and properties of this class to create or modify a section or division object. You cannot use this class to delete a section or division object.

Word Pro: Section class

Sections are areas of text in a document that reside within divisions.

Base Classes

BaseObject

Derived Classes

[IndexSection](#)

Contained by

Usage

Word Pro: SetTabsDialog class

The Word Pro dialog used to set tabs in a document.

Base Classes

BaseObject

Derived Classes

None.

Contained by

ApplicationWindow in the SetTabsDialog Property

Usage

Word Pro: Shadow class

The setting of shadow depth for text, frames, tables, headers, footers, and pages in a document.

Base Classes

BaseObject

Derived Classes

None.

Contained by

Layout in the Shadow Property

ParagraphBorder in the Shadow Property

Usage

Use the properties in this class to set the color or the depth of a shadow for text, frames, tables, headers, footers, and pages in a document.

Word Pro: SilverBulletCollection class

A collection of silver bullet objects in the foundry of a specific division.

Base Classes

BaseObject\BaseCollection

Derived Classes

None.

Contained by

Foundry in the SilverBullets Property

Usage

Use this collection to access any of the silver bullet objects in the foundry of a specific division.

Word Pro: SilverBullet class

A silver bullet object in a document.

Base Classes

BaseObject

Derived Classes

None.

Contained by

Bullet in the SilverBullet Property

Usage

Word Pro: SmartCorrectCollection class

A collection of SmartCorrect objects in a WPAApplication object.

Base Classes

BaseObject/BaseCollection

Derived Classes

None.

Contained by

WPAApplication in the SmartCorrects Property

Usage

Word Pro: SmartCorrect class

Provides access to Word Pro's SmartCorrect options and entries.

Base Classes

BaseObject

Derived Classes

None.

Contained by

WPAApplication in the SmartCorrect Property

Usage

SmartCorrect is a feature of Spell Check that automatically corrects certain errors as you type.

Word Pro: SmartFillCollection class

A collection of SmartFill objects in a WPApplication object. This collection class contains SmartFill objects. There is one SmartFill object for each supported language.

Base Classes

BaseObject/BaseCollection

Derived Classes

None.

Contained by

WPApplication in the SmartCorrects Property

Usage

You can determine which SmartFill object you want to use by specifying the language number. For example, SmartFill(1033) represents the SmartFill object for the American language. A list of language numbers is located in the KeyboardLanguage property description in LotusScript Help.

Word Pro: SmartFill class

This language element is not yet defined.

Base Classes

BaseObject

Derived Classes

None.

Contained by

None.

Usage

Word Pro: SortKey class

This class allows you to set the sort options for each of the sort levels.

Base Classes

BaseObject

Derived Classes

None.

Contained by

SortOptions in the SortLevel1 Property

SortOptions in the SortLevel3 Property

SortOptions in the SortLevel2 Property

Usage

Use the properties in this class to specify the field or column that you want to sort, and in what order you want the data to sort.

Word Pro: SortOptions class

The sort options displayed in the Sort dialog box.

Base Classes

BaseObject

Derived Classes

None.

Contained by

Division in the SortOptions Property

TextDocument in the SortOptions Property

Usage

Word Pro: Spacing class

The spacing of objects in Word Pro, such as text, tabs, columns, and so on.

Base Classes

BaseObject

Derived Classes

None.

Contained by

[ClickHere](#) in the [Spacing](#) Property

[ParagraphStyle](#) in the [Spacing](#) Property

[Text](#) in the [Spacing](#) Property

[TextMarker](#) in the [Spacing](#) Property

Usage

Word Pro: StatusBarButtonCollection class

A collection of status bar button objects in the StatusBar class.

Base Classes

BaseObject\BaseCollection

Derived Classes

None.

Contained by

StatusBar in the StatusBarButtons Property

Usage

Use this collection to access any of the status bar buttons in a StatusBar object.

Word Pro: StatusBarButton class

A button on the status bar object in the Word Pro application.

Base Classes

BaseObject

Derived Classes

None.

Contained by

ApplicationWindow in the StatusBar Property

Usage

Use this class to set options for individual buttons located on the status bar.

Word Pro: StatusBar class

The status bar object in the Word Pro application, visible at the bottom of the workspace.

Base Classes

BaseObjectWindow

Derived Classes

None.

Contained by

ApplicationWindow in the StatusBar Property

Usage

When Word Pro is opened, the status bar also opens at the bottom of the workspace. You can view the status bar whenever Word Pro opens, but you cannot use it until you open a document.

Word Pro: StringCollection class

A collection of strings.

Base Classes

BaseObject\BaseCollection

Derived Classes

None.

Contained by

[BaseTable](#) in the [RowLayouts](#) Property

[BaseTable](#) in the [ColumnLayouts](#) Property

[BaseTable](#) in the [CellLayouts](#) Property

[Division](#) in the [DivisionNames](#) Property

[Filter](#) in the [TableExports](#) Property

[Filter](#) in the [TextandTableExports](#) Property

[Filter](#) in the [GraphicImports](#) Property

[Filter](#) in the [TextandTableImports](#) Property

[Filter](#) in the [GraphicExports](#) Property

[Filter](#) in the [TableImports](#) Property

[IconBarManager](#) in the [IconBarSets](#) Property

[Preferences](#) in the [SnapShotSaveOptions](#) Property

[PrintManager](#) in the [FaceNames](#) Property

[PrintManager](#) in the [PaperNames](#) Property

[PrintManager](#) in the [BinNames](#) Property

[PrintSettings](#) in the [Divisions](#) Property

[RevisionDisplay](#) in the [UndoLevels](#) Property

[ScriptDataSet](#) in the [DataNames](#) Property

[TextDocument](#) in the [DivisionNames](#) Property

[UserInterfacePrefs](#) in the [StylePaths](#) Property

[UserInterfacePrefs](#) in the [DocumentPaths](#) Property

[UserInterfacePrefs](#) in the [UserDictionaryPaths](#) Property

[UserInterfacePrefs](#) in the [BackupPaths](#) Property

[UserInterfacePrefs](#) in the [UserDictionaryFiles](#) Property

[UserInterfacePrefs](#) in the [Units](#) Property

[UserInterfacePrefs](#) in the [IconPaths](#) Property

[UserInterfacePrefs](#) in the [MacroPaths](#) Property

[UserInterfacePrefs](#) in the [GlossaryDataPaths](#) Property

[UserInterfacePrefs](#) in the [GlossaryDataFiles](#) Property

[Version](#) in the [Editors](#) Property

Usage

Use this collection to access any of the string objects in the String class.

Word Pro: SubPageContainer class

The container object for sub pages. This object only exists for one sub page at a time and only when there is a page with in the focus.

Base Classes

BaseObject\BaseContainer\PageContainer

Derived Classes

None.

Contained by

Not contained in the property of any other object.

Usage

The primary use for a SubPageContainer object is to provide quick and easy access to the PageLayout object for the currently active sub page. A SubPageContainer object always represents the sub page that currently has the focus. Therefore, if you assign a SubPageContainer object to a variable, you can use that variable to access the currently active sub page. However, you must remember that the sub page referenced by the variable changes as the focus moves from one sub page to another. This is because the variable references the SubPageContainer object, and the SubPageContainer object always represents the sub page that has the focus. If there is no sub page with in the focus, there is no SubPageContainer object. Therefore, a variable that stores a SubPageContainer object has a null value whenever the focus does not contain a sub page. There is never more than one SubPageContainer object at any time.

For more information about container objects, see [BaseContainer](#).

Word Pro: SuperPageContainer class

The container object for super pages. This object only exists for one super page at a time and only when there is a page with in the focus.

Base Classes

BaseObject\BaseContainer\PageContainer

Derived Classes

None.

Contained by

Not contained in the property of any other object.

Usage

The primary use for a SuperPageContainer object is to provide quick and easy access to the PageLayout object for the currently active super page. A SuperPageContainer object always represents the super page that currently has the focus. Therefore, if you assign a SuperPageContainer object to a variable, you can use that variable to access the currently active super page. However, you must remember that the super page referenced by the variable changes as the focus moves from one super page to another. This is because the variable references the SuperPageContainer object, and the SuperPageContainer object always represents the super page that has the focus. If there is no super page with in the focus, there is no SuperPageContainer object. Therefore, a variable that stores a SuperPageContainer object has a null value whenever the focus does not contain a super page. There is never more than one SuperPageContainer object at any time.

For more information about container objects, see [BaseContainer](#).

Word Pro: SuperTableCollection class

A collection of super table objects in the foundry of a specific division.

Base Classes

BaseObject\BaseCollection

Derived Classes

None.

Contained by

Foundry in the SuperTables Property

Usage

Use this collection to access any of the super table objects in the foundry of a specific division.

Word Pro: SuperTableContainer class

The container object for super tables. This object only exists for one super table at a time and only when there is a table with in the focus. When a SuperTableContainer object is present, it is stored in the SuperTableContainer property on the WPAApplication object.

To access the container object for a table, use the [TableOnlyContainer](#) property on WPAApplication.

To access the container object for parallel columns, use the [TableContainer](#) property on WPAApplication.

Base Classes

BaseObject\BaseContainer

Derived Classes

None.

Contained by

[WPAApplication](#) in the [SuperTableContainer](#) Property

Usage

The primary use for a SuperTableContainer object is to provide quick and easy access to the SuperTableLayout object for the currently active super table. A SuperTableContainer object always represents the super table that currently has the focus. Therefore, if you assign a SuperTableContainer object to a variable, you can use that variable to access the currently active super table. However, you must remember that the super table referenced by the variable changes as the focus moves from one super table to another. This is because the variable references the SuperTableContainer object, and the SuperTableContainer object always represents the super table that has the focus. If there is no super table with in the focus, there is no SuperTableContainer object. Therefore, a variable that stores a SuperTableContainer object has a null value whenever the focus does not contain a super table. There is never more than one SuperTableContainer object at any time.

A table in Word Pro consists of a SuperTableLayout object, which is in turn comprised of one or more TableLayout objects. TableLayout objects contain RowLayout and ColumnLayout objects. Most of these layouts can be accessed through a corresponding container object.

In order to modify attributes of an entire Word Pro table, you must access the SuperTableLayout object of the table. This can be done by accessing the Layout property of the SuperTableContainer object. For example, if you want to set the background color of an entire table, you can use the following statements:

```
.SuperTableContainer.Layout.Background.Pattern = $ItsFillSolid  
.SuperTableContainer.Layout.Background.Color.Red = 255
```

For more information about container objects, see [BaseContainer](#).

Word Pro: SuperTableGroupLayout class

A super table layout for a group of super tables.

Base Classes

BaseObjectLayout

Derived Classes

None.

Contained by

Usage

Word Pro: SuperTableLayoutCollection class

A collection of super table layout objects in the foundry of a specific division.

Base Classes

BaseObject\BaseCollection

Derived Classes

None.

Contained by

Foundry in the SuperTableLayouts Property

Usage

Use this collection to access any of the super table layout objects in the foundry of a specific division.

Word Pro: SuperTableLayout class

The layout for a super table object.

Base Classes

BaseObjectLayout

Derived Classes

None.

Contained by

Usage

A table in Word Pro consists of a SuperTableLayout object, which is in turn comprised of one or more TableLayout objects. TableLayout objects contain RowLayout and ColumnLayout objects.

In order to modify attributes of an entire Word Pro table, you must access the SuperTableLayout object of the table. For example, if you want to set the background color of an entire table, you can use the following statements:

```
.SuperTableContainer.Layout.Background.Pattern = $LtsFillSolid  
.SuperTableContainer.Layout.Background.Color.Red = 255
```

In order to modify attributes of selected cells, you must access the Table object. For example, in order to set the background pattern of cells which are currently selected, you can use the following statements:

```
.Table.TableFill.Background.Pattern = $LtsFillBubbles  
.Table.TableFill.FillStyle = $LwpTableFillStyleAll
```

Notice that in the previous example, the background color and pattern color of the selected cells will not be modified. Word Pro uses your current settings for these colors, which may include a white pattern on a white background or a black pattern on a black background.

Layout attributes of row and column objects can also be accessed through a Table object. For example, if you want to modify the width of all columns which are currently selected, you can use the following statement:

```
.Table.CurrentColumn.Width = 720
```

Word Pro: SuperTable class

A super table object in a document.

Base Classes

BaseObject\Content

Derived Classes

None.

Contained by

Usage

A table in Word Pro consists of a SuperTableLayout object, which is in turn comprised of one or more TableLayout objects. TableLayout objects contain RowLayout and ColumnLayout objects.

In order to modify attributes of an entire Word Pro table, you must access the SuperTableLayout object of the table. For example, if you want to set the background color of an entire table, you can use the following statements:

```
.SuperTableContainer.Layout.Background.Pattern = $LtsFillSolid  
.SuperTableContainer.Layout.Background.Color.Red = 255
```

In order to modify attributes of selected cells, you must access the Table object. For example, in order to set the background pattern of cells which are currently selected, you can use the following statements:

```
.Table.TableFill.Background.Pattern = $LtsFillBubbles  
.Table.TableFill.FillStyle = $LwpTableFillStyleAll
```

Notice that in the previous example, the background color and pattern color of the selected cells will not be modified. Word Pro uses your current settings for these colors, which may include a white pattern on a white background, or a black pattern on a black background.

Layout attributes of row and column objects can also be accessed through a Table object. For example, if you want to modify the width of all columns which are currently selected, you can use the following statement:

```
.Table.CurrentColumn.Width = 720
```

Word Pro: TableCollection class

A collection of base table objects in the foundry of a specific division.

Base Classes

BaseObject\BaseCollection

Derived Classes

None

Contained by**Usage**

Use this collection to access any of the base table objects in the foundry of a specific division.

Word Pro: TableContainer class

A container class that provides the basis for both the ParallelColsContainer and the TableOnlyCont classes. This class is used as the data type for the TableContainer property on WPAApplication. This allows the TableContainer property to contain either TableOnlyCont or ParallelColsContainer objects.

For more information on container classes, see [BaseContainer](#).

To access the container object for a table, use the [TableOnlyContainer](#) property on WPAApplication.

To access the container object for parallel columns, use the [TableContainer](#) property on WPAApplication.

Base Classes

BaseObject\BaseContainer

Derived Classes

[ParallelColsContainer](#)

[TableOnlyCont](#)

Contained by

[WPAApplication](#) in the [TableContainer](#) Property

Usage

Do not use this class for instantiating objects. You may use this class as the data type for a variable in which you wish to store objects created from either of this class' derived classes (TableOnlyCont or ParallelColsContainer).

Word Pro: TableFill class

This class allows you to modify appearance properties of selected cell objects within a table.

Base Classes

BaseObject

Derived Classes

None.

Contained by

BaseTable in the TableFill Property

Usage

The TableFill class gives you access to background and fill style information for table cells. It allows you to fill selected cells, or certain ranges of selected cells.

Word Pro: TableHeadingCollection class

A collection of table heading objects in the foundry of a specific division.

Base Classes

BaseObject\BaseCollection

Derived Classes

None.

Contained by

Foundry in the TableHeadings Property

Usage

Use this collection to access any of the table heading objects in the foundry of a specific division.

Word Pro: TableHeadingLayoutCollection class

A collection of table heading layout objects in the foundry of a specific division.

Base Classes

BaseObject\BaseCollection

Derived Classes

None.

Contained by

Foundry in the TableHeadingLayouts Property

Usage

Use this collection to access any of the table heading layout objects in the foundry of a specific division.

Word Pro: TableHeadingLayout class

The layout for a table heading in a division.

Base Classes

BaseObject\Layout\TableLayout

Derived Classes

None.

Contained by

Usage

Word Pro: TableHeading class

A heading for a table object.

Base Classes

BaseObject\Content\BaseTable

Derived Classes

None.

Contained by**Usage**

Word Pro: TableLayoutCollection class

A collection of table layout objects in the foundry of a specific division.

Base Classes

BaseObject\BaseCollection

Derived Classes

None.

Contained by

Foundry in the TableLayouts Property

Foundry in the TableStyles Property

Usage

Use this collection to access any of the table layout objects with in the foundry of a specific division.

Word Pro: TableLayout class

The layout for a table object.

Base Classes

BaseObjectLayout

Derived Classes

[EndnoteLayout](#)

[FootnoteLayout](#)

[TableHeadingLayout](#)

Contained by

Usage

A table in Word Pro consists of a SuperTableLayout object, which, in turn, is comprised of one or more TableLayout objects. TableLayout objects contain RowLayout and ColumnLayout objects.

In order to modify attributes of an entire Word Pro table, you must access the SuperTableLayout object of the table. For example, if you want to set the background color of an entire table, you can use the following statements:

```
.SuperTableContainer.Layout.Background.Pattern = $LtsFillSolid  
.SuperTableContainer.Layout.Background.Color.Red = 255
```

In order to modify attributes of selected cells, you must access the Table object. For example, in order to set the background pattern of cells which are currently selected, you can use the following statements:

```
.Table.TableFill.Background.Pattern = $LtsFillBubbles  
.Table.TableFill.FillStyle = $LwpTableFillStyleAll
```

Notice that in the previous example, the background color and pattern color of the selected cells will not be modified. Word Pro uses your current settings for these colors, which may include a white pattern on a white background or a black pattern on a black background.

Layout attributes of row and column objects can also be accessed through a Table object. For example, if you want to modify the width of all columns which are currently selected, you can use the following statement:

```
.Table.CurrentColumn.Width = 720
```

Word Pro: TableLine class

A borderline, outline, or diagonal line in a table object.

Base Classes

BaseObject

Derived Classes

None.

Contained by

BaseTable in the TableLine Property

Usage

Use this class to set or modify borderline, outline, or diagonal line styles in a table object.

Word Pro: TableMarkerCollection class

A collection of table marker objects in the foundry of a specific division.

Base Classes

BaseObject\BaseCollection

Derived Classes

None.

Contained by

Foundry in the TableMarkers Property

Usage

Use this collection to access any of the table marker objects in the foundry of a specific division.

Word Pro: TableMarker class

A marker for a table object in a division.

Base Classes

BaseObjectMarker

Derived Classes

None.

Contained by**Usage**

Word Pro: TableOnlyCollection class

A collection of table only objects in the foundry of a specific division.

Base Classes

BaseObject\BaseCollection

Derived Classes

None.

Contained by

Foundry in the Tables Property

Usage

Use this collection to access any of the table only objects in the foundry of a specific division.

Word Pro: TableOnlyCont class

The container object for tables. This object only exists for one table at a time and only when there is a table with in the focus. When a TableOnlyCont object is present, it is stored in the TableOnlyContainer property on the WPAApplication object.

Base Classes

BaseObject\BaseContainer\TableContainer

Derived Classes

None.

Contained by

WPAApplication in the TableOnlyContainer Property

Usage

The primary use for a TableOnlyCont object is to provide quick and easy access to the TableLayout object for the currently active table. A TableOnlyCont object always represents the table that currently has the focus. Therefore, if you assign a TableOnlyCont object to a variable, you can use that variable to access the currently active table. However, you must remember that the table referenced by the variable changes as the focus moves from one table to another. This is because the variable references the TableOnlyCont object, and the TableOnlyCont object always represents the table that has the focus. If there is no table with in the focus, there is no TableOnlyCont object. Therefore, a variable that stores a TableOnlyCont object has a null value whenever the focus does not contain a table. There is never more than one TableOnlyCont object at any time.

The container classes for tables and parallel columns are derived from the same abstract class: TableContainer. However, only the table container object has its own property on WPAApplication. The container object for parallel columns, when it is present, is stored in the TableContainer property on WPAApplication.

For more information about container objects, see BaseContainer.

Word Pro: Table class

A table object in a document.

Base Classes

BaseObject\Content\BaseTable

Derived Classes

None.

Contained by

WPApplication in the Table Property

Usage

A table in Word Pro consists of a SuperTableLayout object, which is in turn comprised of one or more TableLayout objects. TableLayout objects contain RowLayout and ColumnLayout objects.

In order to modify attributes of an entire Word Pro table, you must access the SuperTableLayout object of the table. For example, if you want to set the background color of an entire table, you can use the following statements:

```
.SuperTableContainer.Layout.Background.Pattern = $LtsFillSolid  
.SuperTableContainer.Layout.Background.Color.Red = 255
```

In order to modify attributes of selected cells, you must access the Table object. For example, in order to set the background pattern of cells which are currently selected, you can use the following statements:

```
.Table.TableFill.Background.Pattern = $LtsFillBubbles  
.Table.TableFill.FillStyle = $LwpTableFillStyleAll
```

Notice that in the previous example, the background color and pattern color of the selected cells will not be modified. Word Pro uses your current settings for these colors, which may include a white pattern on a white background or a black pattern on a black background.

Layout attributes of row and column objects can also be accessed through a Table object. For example, if you want to modify the width of all columns that are currently selected, you can use the following statement:

```
.Table.CurrentColumn.Width = 720
```

Word Pro: TabRack class

Acts as the tab parent class for all tab objects in a document.

Base Classes

BaseObject

Derived Classes

None.

Contained by

[ClickHere](#) in the [TabRack](#) Property

[Layout](#) in the [TabRack](#) Property

[ParagraphStyle](#) in the [TabRack](#) Property

[Text](#) in the [TabRack](#) Property

[TextMarker](#) in the [TabRack](#) Property

Usage

Word Pro: TextCollection class

A collection of text objects in the foundry of a specific division.

Base Classes

BaseObject\BaseCollection

Derived Classes

None.

Contained by

Foundry in the Texts Property

Usage

Use this collection to access any of the text objects in the foundry of a specific division.

Word Pro: TextDocument class

A Word Pro document, including all its divisions, sections, versions, pages, text objects, frames, tables, styles, print settings, scripts, and so on.

Base Classes

BaseObject\Document

Derived Classes

None.

Contained by

Application in the ActiveDocument Property

ApplicationWindow in the ActiveDocument Property

Division in the Master Property

DocWindow in the Document Property

TextDocument in the Master Property

Usage

For each Lotus application that uses document-like objects, there is a corresponding subclass which is derived from the Document class. In Word Pro, the subclass is TextDocument. In Lotus 1-2-3, the subclass is Sheet. In Lotus Notes, the subclass is Note. The OpenDocument method on the WPAApplication class enables you to open an existing Word Pro document. WPAApplication also provides CreateDocument for creating new objects from the TextDocument class. It is also possible to get a TextDocument object associated with an already open document by using the LotusScript Bind operator. To get an object created by another Lotus application, use the GetObject operation.

Note In Word Pro for OS/2 Warp 4, it is not possible to use the GetObject function to get a Word Pro TextDocument object. To get a Word Pro TextDocument object, use the following:

```
Set WPObj = GetObject (."WordPro.Application")
Set TextDocObj = WPObj.OpenDocument("c:\temp\sample.lwp")
```

The Name property (inherited from BaseObject) returns the file name (without the path) of the Word Pro document.

Properties

Changed As Bool - Has the document changed since the last save?

Embedded As Bool - Is the document OLE embedded?

FullName As String - Name and path of this document.

IsOpen As Bool - Checks to see if a document is open, but since you cannot read a closed document, this value is always True.

Location As String - Path only of this document. In the future, this may be a variant.

Path As String - Path only of this document.

PrintSettings As PrintSettings - Allows you to set different print settings for each document.

ReadOnly As Bool - Allows you to set a document to ReadOnly or ReadWrite.

Saved As Bool - Has the document ever been saved?

Methods

Activate - Makes this document active.

Close - Closes your document. All parameters are ALOI, except the CloseFile parameter which = If closing this and it is the last one, do you want a new untitled document to appear? CloseFile enum.

CopySelection - Copy to Clipboard.

CutSelection - Cut to Clipboard.

Paste - Pastes Clipboard. The MakeVisible parameter allows you to bring what is pasted into view when Paste occurs.

Print - LOI parameters; NoDialog added which inhibits the progress box.

PrintOut - Same as Print but exists for VB users.

Save - Saves the document.

SaveAs - Saves as another file type. Most parameters are LOI. AddToLastFileOpen list (recent files) added.
SaveCopyAs allows you to save a copy of an OLE launched object.

Events

PreClose - Raised prior to closing.

Save - Pre-Save event.

SaveAs - Pre-SaveAs.

Saved - After Save.

SavedAs - After SaveAs.

Opened - After opening.

Created - After creating.

PrePrint - Before printing.

Keystroke WMCommand - Raised when a menu item is picked, a pre-defined icon is clicked, or certain push buttons cause another dialog box to display from a dialog box. This event passes the following information to the script event handler:

- menu ID
- icon ID
- ID of the menu command, WPBITMSK.LSS, containing all menu IDs

Word Pro: TextMarkerCollection class

A collection of text marker objects in the foundry of a specific division.

Base Classes

BaseObject\BaseCollection

Derived Classes

None.

Contained by

Foundry in the TextMarkers Property

Usage

Use this collection to access any of the text marker objects in the foundry of a specific division.

Word Pro: TextMarker class

Marks an insertion point or a selection of text. Derives from Marker. A hidden object used to attach some data or functionality to text in the document.

Base Classes

BaseObjectMarker

Derived Classes

None.

Contained by**Usage**

Word Pro: TextStyleCollection class

A collection of paragraph and character style objects in the foundry of a specific division.

Base Classes

BaseObject\BaseCollection

Derived Classes

None.

Contained by

Foundry in the TextStyles Property

Usage

Use this collection to access any of the text style objects in the foundry of a specific division.

Word Pro: Text class

A text object in a document.

Base Classes

BaseObject\Content

Derived Classes

None.

Contained by

[Bullet](#) in the [Text](#) Property

[ClickHere](#) in the [Prompt](#) Property

[DivisionInfo](#) in the [FillerPageText](#) Property

[Graphic](#) in the [Text](#) Property

[NoteLayout](#) in the [Text](#) Property

[SilverBullet](#) in the [Text](#) Property

[WPApplication](#) in the [Text](#) Property

Usage

Word Pro: TOCSuperTableLayout class

A layout for a table of contents super table object.

Base Classes

BaseObjectLayout

Derived Classes

None.

Contained by

Usage

Word Pro: UserInterfacePrefs class

The user interface preferences in Word Pro.

Base Classes

BaseObject

Derived Classes

None.

Contained by

ApplicationWindow in the UserInterfacePrefs Property

Usage

Use this class to set options in the Word Pro Preferences dialog boxes, including default paths, default directories, load options, display options, Spell Check options, and so on.

Word Pro: UseWhen class

Base Classes

BaseObject

Derived Classes

None.

Contained by

Layout in the UseWhen Property

Usage

Word Pro: VersionCollection class

A collection of version objects in the VersionManager class.

Base Classes

BaseObject\BaseCollection

Derived Classes

None.

Contained by

VersionManager in the Versions Property

Usage

Use this collection to access any of the version objects in the VersionManager class.

Word Pro: VersionManager class

A tool to manage the Version tool in a division.

Base Classes

BaseObject

Derived Classes

None.

Contained by

Division in the VersionManager Property

TextDocument in the VersionManager Property

Usage

Word Pro: Version class

The Version tool in the Word Pro application.

Base Classes

BaseObject

Derived Classes

None.

Contained by

VersionManager in the CurrentVersion Property

Usage

Word Pro: Window class

A window in the Word Pro application.

Base Classes

BaseObject

Derived Classes

[ApplicationWindow](#)

[DocWindow](#)

[StatusBar](#)

Contained by

Usage

Word Pro: WinViewPrefs class

The window view preferences in Word Pro.

Base Classes

BaseObject

Derived Classes

None.

Contained by

[DocWindow](#) in the [WinViewPrefs](#) Property

[UserInterfacePrefs](#) in the [WinViewPrefs](#) Property

Usage

Represents the view preference options available in Word Pro. For example, you can use properties of this class to set the custom view level, or determine which marks are displayed with in the document.

For information about how WinViewPrefs objects are used within different classes, see the [WinViewPrefs property](#) topic.

Word Pro: WPAApplication class

The whole of the Word Pro application, including the application engine, workspace, and any documents created by the application. You can access any part of Word Pro, or a Word Pro document through the members of the WPAApplication object.

Base Classes

BaseObject\Application

Derived Classes

None.

Contained by

BaseObject in the Application Property

Usage

Each time you launch Word Pro, a single object is instantiated from the WPAApplication class. That object represents the Word Pro application. Under normal circumstances, there will be only one WPAApplication object active at any time. WPAApplication gives you access to all the Word Pro-specific objects, features, events, and information which exist as part of Word Pro. You can use the WPAApplication object to access and control nearly every aspect of Word Pro and its components.

Static and Current Context Properties

WPAApplication is unique in that it defines two types of properties: static and current context. The static properties, such as Name and AppViewPrefs, represent things which apply to the Word Pro application as a whole. Their contents are not dependent on which document or other object is active. The contents of static properties remain the same, regardless of which document may be active or where the focus of the application is directed.

The content of current context properties changes as the focus of Word Pro moves from one document or division to another. These properties are called current context properties because their contents depend on the current context of Word Pro. For example, in a document with two divisions named ChapterOne and ChapterTwo, the focus of Word Pro can move from one division to the other. While the focus is in the ChapterOne division, the current context of Word Pro is the ChapterOne Division object, and the Division property on WPAApplication contains the ChapterOne Division object, as long as the ChapterOne division is active. However, when you move the focus of Word Pro into the division named ChapterTwo, the current context changes, and the contents of the Division property changes to the ChapterTwo Division object. It is important for you to keep current context in mind whenever you access the current context properties on the WPAApplication object.

The current context properties include:

- ActiveDocument
- ActiveDocWindow
- ApplicationWindow
- BaseTable
- Cell
- Container
- Content
- CurrentCell
- CurrentColumn
- CurrentRow
- Division
- Divisions
- Documents
- Foundry
- Frame
- Graphic
- GraphicOleObject
- Layout
- OleObject
- Page
- ParallelColumns
- SuperTableContainer
- Table
- TableContainer

TableOnlyContainer
Text

The Abstract Application Class

Each Lotus application is represented in LotusScript by its own unique application object. Each application object is instantiated from a unique application class. But all application classes share a common abstract application class, called Application. This abstract class provides a common fundamental definition for all application objects. However, each Lotus application uses the abstract Application class as a starting point, and then defines additional properties, methods, and events as needed to create its own unique application class. In this way, each Lotus application class shares a common set of class members from the abstract class, while still providing the additional class members needed to represent the unique components of each application. The table below shows the name of each product's application class (each derived from the same abstract Application class), the name of the object instantiated from that application class, and the name of the global variable used for accessing that object.

Product Name	Application Class	Object Name	Global Variable
Word Pro	WPApplication	WordPro	CurrentApplication
Lotus 1-2-3	Application	1-2-3	CurrentApplication
Approach	Application	Approach	CurrentApplication
Freelance	Application	Freelance	

When referring to the WPApplication object from within a Word Pro script, you should use the global variable, CurrentApplication.

When referring to the WPApplication object from outside of Word Pro, you should use the GetObject function with the name, "WordPro."

Note In Word Pro for OS/2 Warp 4, it is not possible to use the GetObject function to get a Word Pro TextDocument object.

LotusScript and Multiple Sessions Of Word Pro

A single object (named WordPro) is instantiated from the WPApplication class each time you launch the Word Pro application. If you run multiple sessions of Word Pro simultaneously, each session of Word Pro is represented by its own WPApplication object. You can run multiple sessions of Word Pro and use LotusScript to manipulate the different instances of WPApplication. However, the memory requirements for running multiple sessions are quite high and great care must be taken to identify correctly the different WPApplication objects. A single error in managing multiple WPApplication objects could result in the loss of valuable data or unexpected changes in the default settings within Word Pro.

Note The documentation for the Word Pro LotusScript object model assumes that you never have more than one Word Pro session running at any time. While it is possible to run multiple Word Pro sessions and use LotusScript to manipulate them, it is not recommended.

If you do write scripts to control multiple sessions of Word Pro, you should keep in mind the following:

- The WPApplication object which is active when your script starts running is the "native" WPApplication object. All other WPApplication objects must be accessed through OLE automation, using the LotusScript GetObject or CreateObject functions.
- Word Pro reads default settings from .INI files as those default settings are used within a Word Pro session. For example, when the user opens the Set Tabs dialog box, Word Pro checks an .INI file for the current tab settings and displays those settings in the dialog box.
- Word Pro writes any changes to a default setting into an .INI file when the Word Pro session ends. For example, if the user changes the type of leader dot in the Set Tabs dialog box, that change will be written into an .INI file when the user ends the Word Pro session.
- Changes made to an .INI file by one Word Pro session may affect other simultaneous Word Pro sessions, if the first session ends and the other sessions make use of the settings changed in the first session. This is due to the fact that the other sessions read from the same .INI file as the first session and reflect any changes made to the .INI file when the first session ended.

For example, while running two sessions of Word Pro simultaneously, you can change the value of the LeaderDotType property on the AppViewPrefs object in one session of Word Pro. You can then end that session, at which time the appropriate .INI file will be updated with the new LeaderDotType value. You can then open the Set Tabs dialog box from within the second session and see the new LeaderDotType value reflected in the dialog box settings. This can result in some unpredictable behavior if you run multiple sessions of Word Pro.

Word Pro: WPDataSetCollection class

A collection of Word Pro data sets in the Word Pro application object, a document object, or some object within a document.

Base Classes

BaseObject\BaseCollection\ScriptDataSetCollection

Derived Classes

None.

Contained by

[CharacterStyle](#) in the [WPDataSets](#) Property

[ClickHere](#) in the [WPDataSets](#) Property

[Division](#) in the [WPDataSets](#) Property

[Layout](#) in the [WPDataSets](#) Property

[Marker](#) in the [WPDataSets](#) Property

[ParagraphStyle](#) in the [WPDataSets](#) Property

[TextDocument](#) in the [WPDataSets](#) Property

[Text](#) in the [WPDataSets](#) Property

[TextMarker](#) in the [WPDataSets](#) Property

[WPApplication](#) in the [WPDataSets](#) Property

Usage

Item ScriptDataSet is not implemented.

Use the WPDataSetCollection class to create and manage sets of variables attached to Word Pro objects.

Word Pro: WPDataSet class

The Word Pro data set that holds a collection of variables. Each variable has a name and a string value.

Base Classes

BaseObject\ScriptDataSet

Derived Classes

None.

Contained by**Usage**

Each WPDataSet can be attached to the following Word Pro objects: TextDocument, Layout, Marker, CharacterStyle, ParagraphStyle, WPAApplication, Division, and Text classes. If you close a document that has a data set attached to it, Word Pro saves the data set(s) with the document. When you reopen the document, Word Pro reads the data sets back in from the document.

A special data construct which corresponds to a specific type of item in an application and defines what aspects of that item you can access through LotusScript, and what you can do with that item. For example, the TextDocument class in Word Pro corresponds to a Word Pro document. The class definition includes variables for storing the name of the document and other information. The class definition also includes methods for performing specific tasks with a document. In short, the TextDocument class defines exactly how much of a Word Pro document you can access and what you can do with it.

When you open a document in Word Pro, Word Pro creates an instance of the TextDocument class. That instance of the TextDocument class is an object. You use that object to control the document you opened.

By Category

Classes

Properties

Methods

Events

Overview: Division names in LotusScript

In LotusScript, each division has two names: an external name and an internal name.

External division names

The external name is the name you see displayed in the divider tab on your screen. This external name is read-write and does not have to be unique. While the external name is usually for display purposes only, you can use the external name to retrieve the internal name as described below. The external name is stored in the Name property on the DivisionInfo object.

This statement returns the external name for the current division:

```
.Division.DivisionInfo.Name
```

Internal division names

The internal name is read-only and is unique for each division. It allows you to uniquely identify a division when you use methods to access and control that division. The internal name is stored in the Name property on each Division object.

This statement returns the internal name for the current division:

```
.Division.Name
```

Getting the internal name of a division

Every method which allows you to control a division requires that you identify a division by its unique internal name. Getting the internal name for the currently active division is done in the following statement:

```
.Division.Name
```

However, to get the internal name of a division when you are not in that division, you must first gain access to that division. You can gain access to any division by assigning that division to a variable, using the Bind statement.

The Bind statement allows you to assign an object to a variable using that object's name. In the case of a division, the Bind statement allows you to use the division's external name, as shown in the following statements:

```
Dim mydivision As Division
Dim divisionname As String
Set mydivision = Bind("!Body")
divisionname = mydivision.Name
```

This example declares two variables (mydivision and divisionname). It then binds the division named "Body" to the mydivision variable. Then it assigns the value of the Name property on mydivision to the other variable (divisionname). You can use the divisionname variable as the value for any parameter which refers to the Body division. For example, if you use the CombineDivisions method, you can use the divisionname variable as the value for either the StartName or EndName parameters.

An action, defined as part of a class, that returns specific information about an object. An event can be a keystroke, entering a layout, or any other action that Word Pro can detect. When an event occurs in Word Pro, we say the event was raised and the object involved in that event is said to have raised the event.

For example, the EnterLayout event is raised each time you move the insertion point to a different layout. You can write a script that will run each time this event is raised. If more than one object has a layout, you must attach the script to the EnterLayout event for each object.

Overview: Internal and external names in LotusScript

In LotusScript, certain types of objects carry two different names: an external name and an internal name. External names are visible to and editable by the Word Pro user. Internal names are only visible within LotusScript and are always read-only.

External names

You use external names to differentiate between similar items in a Word Pro document. For example, you might create two bookmarks in your document and give them different names so you can tell them apart. You could place those two bookmarks in two separate divisions, and Word Pro would allow you to give them the same external name. However, when objects share the same external name, LotusScript needs a way to tell them apart. For that reason, Word Pro provides unique internal names for all objects.

Internal names

Internal names are entirely unique. No two objects of any type will ever have the same internal name. Internal names are usually hexadecimal values of type String. Bookmarks, power fields, ClickHere blocks, DDE markers, and divisions are just some of the objects which have both internal and external names.

The table below shows the location/property for the external and internal names for several objects.

Object	External Name in Word Pro	External Name in LotusScript	Internal Name in LotusScript
ClickHere	Click Here Block Options dialog box. Choose Create - Click Here Block and click Options.	<ClickHereObject>.Name For example: ForAll blocks IN _ .Division.Foundry.ClickHeres Print blocks.Name End ForAll	
Bookmark	Bookmarks dialog box. Choose Create - Bookmark.	<BookmarkObject>.Name For example: ForAll bmarks IN _ .Division.Foundry._ BookmarkManager.Bookmarks Print bmarks.Name End ForAll	<BookmarkObject>.MarkerName For example: ForAll bmarks IN _ .Division.Foundry._ BookmarkManager.Bookmarks Print bmarks.MarkerName End ForAll
PowerField	Document Fields dialog box. Choose Text - Insert Other - Power/Doc. Field.	Not accessible in LotusScript.	<PowerFieldObject>.Name For example: ForAll pfields IN _ .Division.Foundry.Powerfields Print pfields.Name End ForAll
DDE	none	none	
Division	on Division Tab or in InfoBox for Division	<DivisionObject>.DivisionInfo.Name For example: ForAll DIVs IN _ CurrentDocument.Divisions Print DIVs.DivisionInfo.Name End ForAll	<DivisionObject>.Name For example: ForAll DIVs IN _ CurrentDocument.Divisions Print DIVs.Name End ForAll

A function, defined as part of a class, that performs a specific task with any object created from that class. Once an object is instantiated from a class, you can use that class' methods to access and manipulate the object.

For example, you can use the AddDivision method in a TextDocument object to add a division to a Word Pro document.

Acts as a link between your script and a specific part of an application or a document. All objects are created from class descriptions. The class acts as a blueprint or template for the object itself. You can create more than one object from the same class. If you create more than one object from the same class, you can access and control them independently of one another. Each object created from a class is said to be an instance of that class. When an object is created, we say the object has been "instantiated."

A kind of variable which is defined as part of a class. Once an object is instantiated from a class, these variables store information about the object, such as the object's name or location. Properties can also contain other objects instantiated from other classes.

For example, the `TextDocument` class defines a `FullName` property and a `LineNumberOptions` property. When you open a Word Pro document, Word Pro creates an object based on the `TextDocument` class. In addition to filling some properties with strings and numbers, Word Pro must create all the objects which are parts of the `TextDocument` object, such as `LineNumberOptions`, `PrintSettings`, and `SortOptions`. Each object is stored in one of the `TextDocument` object's properties, just as if it were a string or a number.

Overview: The Word Pro LotusScript Object Model

Word Pro provides its own set of LotusScript language tools called classes. These classes define which parts of Word Pro you can access with LotusScript, and what you can do with them. Each Lotus application that uses LotusScript has its own set of these "product-defined classes."

Each Word Pro class corresponds to a type of object in Word Pro, such as windows, frames, and pieces of text. Each type of object has its own Word Pro class. The class itself is like a blueprint for one type of object. The class only defines the parts of an object (called the properties), and the tools for manipulating or interacting with that type of object (called methods and events). When you run Word Pro, Word Pro automatically keeps track of all its active objects. Many objects, such as frames, are created from the same class. But every object has unique qualities, such as name and location, which allow you to control it independently of other objects created from the same class.

When you look at all the LotusScript classes defined in Word Pro, you see a model of the Word Pro application and all its parts. Everything from the application window, to the individual documents, to the divisions and pages of those documents and the information they contain, is represented in LotusScript by an object. Every object has a corresponding class which defines what the object is and how you can interact with it. We call these classes the Word Pro object model because, when seen as a whole, the classes form a model of the objects which comprise Word Pro, its features and documents. When you write a script, you can access and control any part of Word Pro that is represented by a class in the Word Pro object model.

{button ,AL('H_WP_THE_LOTUSSCRIPT_LANGUAGE_OVER;H_USING_LOTUSSCRIPT_IN_WORD_PRO_OVER'
,0)} [See related topics](#)

A Twip is a standard unit of measurement used to measure space on a computer screen. This table shows a few standard measurements in twips:

Inches	Twips
12"	17,280
1"	1,440
3/4"	1,080
1/2"	720
1/4"	360
1/8"	180
1/16"	90

Overview: Using LotusScript in Word Pro

The [LotusScript language](#) works the same way in all Lotus applications. However, Word Pro gives you its own access to LotusScript, dialog boxes, and a script recorder. The following topics explain some of the basic concepts of how to use LotusScript effectively in Word Pro:

[Overview: Word Pro scripts](#)

[Overview: Recording a script](#)

[Playing a script](#)

[Playing an Ami Pro 3.x macro](#)

[Displaying the Script Editor](#)

[Attaching a script to an icon](#)

While using the Word Pro Reference Help, you may encounter terminology with which you are not familiar. For your convenience, many of these terms are defined and explained in the following topics:

[Overview: Word Pro LotusScript Object Model](#)

[Overview: Word Pro LotusScript Object References](#)

[Overview: Word Pro LotusScript Units Of Measurement](#)

[Overview: Word Pro LotusScript Collection Classes](#)

[Overview: Word Pro LotusScript Enumerated Values](#)

[Overview: Word Pro LotusScript Abstract Classes](#)

[Overview: Word Pro LotusScript Object Containment](#)

[Overview: Word Pro LotusScript Class Hierarchy and Inheritance](#)

Overview: Word Pro LotusScript Abstract Classes

Word Pro uses a number of classes, such as BaseObject, Application, and Layout, as a starting point for a group of similar classes, or as a means of passing class members onto an entire group of classes. For example, the BaseObject class serves only one purpose in the Word Pro object model. It provides a set of six basic properties which are then inherited by every single class in the Word Pro object model. The Application class is an abstract class which provides a basic set of properties, methods, and events, which are shared by all Lotus applications. The Layout class provides a basic set of class members which is shared by all its derived classes, including PageLayout, FooterLayout, HeaderLayout, TableLayout, and FrameLayout.

The concept of using one class as the basis for a group of other classes is known as inheritance. When a class gets a set of class members from another class, it is said to inherit those class members. You can tell the origin of class members by looking at the Base Classes and Derived Classes headings in a class definition.

Overview: Word Pro LotusScript Class Hierarchy and Inheritance

Classes based on other classes are said to be derived from the original class.

Overview: Word Pro LotusScript Collection Classes

Word Pro helps you keep track of objects by grouping those objects together in collection objects. Each collection object in Word Pro corresponds to one of the Word Pro object classes. Each object in a collection is said to be an item of that collection. For example, the TextDocumentCollection object contains all objects instantiated from the TextDocument class. Each TextDocument object is said to be an item of the TextDocumentCollection object. Grouping objects together in this way makes it easier to locate and access one or more objects of a particular class.

The Scope of Collections

Each collection object has a fixed scope which determines where the collection object derives its items. Most collections are limited to a particular division. For example, the FrameLayoutCollection object contains all the FrameLayout objects in a particular division in a document. Therefore, if the document has three divisions, Word Pro maintains one FrameLayoutCollection for each division that contains a FrameLayout object.

To access an object through its collection, you must know which collection object contains the object(s) for which you are looking.

Accessing objects in a collection

You can access objects in a collection in one of two ways:

- Iteration - Using the ForAll statement to access every object, in turn, by stepping through the entire collection.

- Indexing - Using the Item method or the indexing syntax to access one specific object in a collection.

In both iteration and indexing, you access the object(s) through the corresponding collection object.

Overview: Word Pro LotusScript Enumerated Values

Many properties and methods in Word Pro make use of enumerated lists of values. These lists of values serve two purposes:

- They define the legal values for a property or method parameter.
- They provide textual and numeric constants for those legal values.

For example, the AlignmentType property has only nine legal values. They are represented in LotusScript by the following constants:

```
$LtsAlignmentSmart  
$LtsAlignmentLeft  
$LtsAlignmentRight  
$LtsAlignmentHorizCenter  
$LtsAlignmentJustify  
$LwpAlignmentTypeJustifyall  
$LwpAlignmentTypeNumericleft  
$LwpAlignmentTypeNumericright  
$LwpAlignmentTypeAlignRevert
```

Each constant corresponds to a different alignment setting for the AlignmentType property. However, these constants can only be used within the confines of LotusScript and they require more keystrokes than their numeric equivalents (which are seen in parentheses below):

```
$LtsAlignmentSmart (1056964612)  
$LtsAlignmentLeft (1056964609)  
$LtsAlignmentRight (1056964610)  
$LtsAlignmentHorizCenter (1056964611)  
$LtsAlignmentJustify (1056964613)  
$LwpAlignmentTypeJustifyall (5)  
$LwpAlignmentTypeNumericleft (6)  
$LwpAlignmentTypeNumericright (7)  
$LwpAlignmentTypeAlignRevert (8)
```

Depending on your needs and preference, you can use either the textual constant or its numeric equivalent. If you call the property or method from a non-Lotus application, you must use the numeric constant.

The data type for a property or parameter which has an enumerated list of values is always Variant. This allows you to use either the textual constant or its numeric equivalent.

Note Many properties, method parameters, and method return values in the Word Pro LotusScript object model are limited to -1 or 0, and will accept the LotusScript constants, True and False, in place of the integers, -1 and 0. These are not enumerated values. They are Boolean expressions that have a data type of Integer. The constants, True and False, can be used anywhere to take the place of -1 and 0. In contrast, enumerated constants can only be used as values in those properties and parameters which list them as legal values.

Bitmasks

Word Pro also employs a second kind of enumerated constant, called a "bitmask." Like the enumerated value described above, a bitmask limits the number of legal values for a property or a parameter. A bitmask also provides both a textual and a numeric constant that you can use interchangeably. However, in a bitmask, you can often combine the constants as a means of achieving a combined result in the property or parameter.

For example, a property might use a bitmask which defined four constants as seen below:

```
LwpStartOnNewPage (&H0)  
LwpStartOnCurrentPage (&H1)  
LwpIncludePrevHeader (&H20)  
LwpIncludePrevFooter (&H10)
```

You can use one of these constants to achieve one of the effects, or you could combine constants to get a combined result. For example, the following combination would start something on a new page and would include the previous header and footer:

LwpStartOnNewPage OR LwpIncludePrevHeader OR LwpIncludePrevFooter

We use the OR operator to combine bitmask constants. This combination of the hexadecimal equivalents can do the same thing:

&H0 OR &H20 OR &H10

Some bitmasks have constants which are mutually exclusive. This means they cannot be combined. In the example above, the first two constants would be mutually exclusive because the first constant starts something on a new page, while the second constant starts the same thing on the current page.

You cannot use the textual constants unless you include the contents of the file, "WPBITMSK.LSS," in the Declarations section of your script. WPBITMSK.LSS provides the link between the hexadecimal bitmask constant and its textual equivalent. Even if you include this file in the Declarations section, you cannot use the textual constant from another application through OLE automation. OLE automation always requires the use of the hexadecimal constant. Word Pro always returns the hexadecimal constant from properties that employ bitmasks.

Note OLE is not supported under OS/2.

To include the WPBITMSK.LSS file, place the following statement in the Declarations script of the !Globals object.

```
%include "wpbitmsk.lss"
```

Overview: Word Pro LotusScript Object Containment

In Word Pro, you will encounter many objects that are contained in the properties of other objects. For example, when you create a frame in Word Pro, that frame is represented in LotusScript by a FrameLayout object. That FrameLayout object has a number of properties. Some of those properties contain strings, or integers. Thus, when you want to set the name of the frame, you can use the following statement:

```
.Frame.Layout.Name = "BlueFrame"
```

But other properties on that FrameLayout object contain smaller objects which are part of the frame you created. For example, the Background property on your FrameLayout object contains a Background object. When you want to manipulate the background of your frame, you use that Background object. Thus, you can set the fill pattern for the background of the frame to a horizontal bar pattern, as follows:

```
.Frame.Layout.Background.Pattern = "$LtsFillHorizBar"
```

But the chain of containment doesn't end there. The Background object also has the Color and BackColor properties which contain Color objects. The Color object in the Color property lets you control the color of the fill pattern in the background of the frame. The Color object in the BackColor property lets you control the color of the null space behind the fill pattern. Thus, if you wanted to set the color of the horizontal bars to red and the space behind them to white, you can use the following code:

```
.Frame.Layout.Background.Color.SetRGB(255,0,0)
.Frame.Layout.Background.BackColor.SetRGB(0,0,0)
```

The Word Pro object model is structured so that many objects are comprised of smaller objects that are contained in the properties of the larger object. This containment of objects gives you much finer control over Word Pro and all its parts.

The focus in contained objects

When an object has the focus, the object (or objects) which contain that object share the focus with that object. For example, when the insertion point is in a table cell, the cell has the focus. But the cell shares the focus with the table that contains the cell, and the WordPro object shares the focus with the Table object. This can affect which properties and methods will be available to you at any given time. Thus, you must bear in mind both the focus and the containment of an object when using leading dot notation to reference an object.

For example, if you use the leading dot to call the Parent property while the CellLayout object has the focus, you get the CellContainer object for that CellLayout. However, if you changed the focus by selecting the table and used the same statement, the leading dot would return the WPApplication object because the Parent property for the Table contains a reference to WPApplication.

Overview: Word Pro LotusScript Object References

Each time Word Pro creates a LotusScript object, it uses that object's class to define exactly how much access you have to an object. How much of an object you can see and what you can do with that object is all determined by the class definition. The class defines properties to give you access to an object's attributes, methods to give you special tools for manipulating the object, and events which allow you to run certain scripts when an event occurs in a specific object. When you use LotusScript to access and manipulate the object, your actions are passed on to the part of the application represented by that object.

The syntax descriptions for properties and methods include "[objectreference.]" at the beginning of the syntax. When you use a property or method in a script statement, you must replace "[objectreference.]" with a reference to the object that contains or defines the property or method you are using.

For example, the syntax for the AlignmentType property is described as follows:

```
[objectreference].AlignmentType
```

The AlignmentType property is defined as part of the Alignment class. Therefore, the AlignmentType property is always part of an Alignment object. When you use the AlignmentType property in a statement, you must include a reference to its Alignment object, as seen in the following statement.

```
.Text.Alignment.AlignmentType
```

The Alignment object is always contained by another object (such as Text). Therefore, you must include the containing object as part of your object reference. The amount of information required in an object reference depends on two factors:

- the object or objects which have focus
- the number of objects within the focus which match the object reference

How the focus affects the object reference

The focus is usually defined as the place in an application that is currently active. For example, when your cursor is in a Word Pro document and that document is active, we say the document has the focus. When you save the document and the Save dialog box opens, we say the dialog box has the focus. There are even more subtle differences in focus. For example, when you move your cursor from the page of your document into a frame, you again change the focus. When you move from the frame to a table, you change the focus again. In each case, you may notice that the menus, SmartIcons, status bar, and even the cursor itself change, depending on what part of Word Pro has the focus. The same holds true when you are running a script.

If you try to use a Text object method while a picture has the focus, you may get an error. If your focus is on a table and you try to check a property on a frame, you will get an error. That is why you must always provide enough of an object reference to specify exactly what object you are trying to access. While it is difficult to define exactly how much of an object reference is required in every situation, there are a few simple rules to keep in mind when referencing an object in a script.

- If the object you are referencing will have the focus when you run the script, you can simply precede the property or method name with a dot (.). Word Pro always interprets a leading dot as the object with the current focus. For example, when a Text object has the focus, you can select the word which has the focus by using the Select method, without explicitly referencing the text object itself:

```
.Select(1749)
```

You can use a property using the same reference:

```
.SelectionHidden = True
```

- If you want to access an object that is contained as a part of another object which has the focus (such as a Font object on a Text object), you must provide the name of the property which contains the object before calling the property or method. For example, while a Text object has the focus, you can reference the Font object in the Font property in this way:

```
.Font.FontName = "Helvetica"
```

Once again, Word Pro sees the leading dot and uses the focus to determine the object reference. Note that the focus in an application can only be placed in certain areas. You cannot place the focus on a Font object; you must provide an object reference to the Font object as seen in the example above.

Note When an object is stored in the property of another object, we say the first object is contained by the second object. For more information about object containment, see [Overview: Word Pro LotusScript Object Containment](#).

- Another means of accessing objects which don't have the focus is through the WordPro object (created from WPAApplication class). The WordPro object provides direct or indirect access to nearly every part of the Word Pro application. Even when your focus is on a Text object, you can use the Application property on that Text

object to access the WordPro object, and therefore the rest of the objects in Word Pro. For example, you can get to the background color of a frame even while the focus remains in a Text object, using the following syntax.

```
.Application.Divisions(itemreference).Foundry.Frames(itemreference).Background.BackColor.SetRGB (255,255,255)
```

Note the use of the properties, Divisions and Frames, in the example above. These properties contain collection objects, each of which serves as a storage area for a particular class of objects. Divisions has a data type of DivisionCollection and contains all the Division class objects in the active document. Frames has a data type of FrameLayoutCollection and contains all the FrameLayout class objects per division. See [Overview: Word Pro LotusScript Collection Classes](#) for more information on collections.

Overview: Word Pro LotusScript Units Of Measurement

Word Pro uses a number of different units of measurement in its LotusScript object model. Some properties and methods accept inches, while others may accept points or twips or units.

The following list of equivalent values may be useful in determining the appropriate value to use.

Inch equivalents

1 inch = 72 points

1 inch = 1440 twips

1 inch = 4718592 units

Point equivalents

1 point = 1/72"

1 point = 20 twips

1 point = 65536 units

Twip equivalents

1 twip = 1/1440"

1 twip = 1/20 point

1 twip = 3276.8 units

Unit equivalents

1 unit = 1/4718592"

1 unit = 1/65536 point

1 unit = 1/3276.8 twips

Overview: Word Pro Menu Command IDs

Below is the list of constants for the Word Pro menu command IDs. You can use either the text constant or the hexadecimal constant to identify a specific Word Pro menu command.

LwpMenuFilemenu	&H64	LwpMenuMIFieldauto	&HE0
LwpMenuMfNew	&H65	LwpMenuMIFieldshowrt	&HE1
LwpMenuMfOpen	&H66	LwpMenuMIFielddoauto	&HE2
LwpMenuMfSave	&H67	LwpMenuMIFieldremove	&HE3
LwpMenuMfSaveas	&H68	LwpMenuMIExechand	&H23E
LwpMenuMfRevert	&H69	LwpMenuMIIndexopts	&HE4
LwpMenuMfAppendtext	&H6A	LwpMenuHelpmenu	&H226
LwpMenuMfImport	&H6B	LwpMenuMhIdxhelp	&H227
LwpMenuMfFilemanage	&H6C	LwpMenuMhAbout	&H228
LwpMenuMfDocdesc	&H6D	LwpMenuMhMacrohelp	&H229
LwpMenuMfCreatedatafile	&H6E	LwpMenuMhCompatible	&H22B
LwpMenuMfPrint	&H6F	LwpMenuMhUsingHelp	&H33E
LwpMenuMfChgprinter	&H70	LwpMenuMhKeyboard	&H33F
LwpMenuMfExit	&H71	LwpMenuMhHowDoI	&H340
LwpMenuMfPrintreset	&H72	LwpMenuMhDoHelp	&H341
LwpMenuMfDde	&H73	LwpMenuMhForUpgraders	&H342
LwpMenuMfMergeaction	&H74	LwpMenuMhTutorial	&H345
LwpMenuMfJustprint	&H75	LwpMenuMhSearch	&H36F
LwpMenuMfStandalonep	&H147	LwpMenuMsControl	&H259
LwpMenuMfAllfldsrename	&H76	LwpMenuMsFileman	&H25A
LwpMenuMfPrintwrongpaper	&H77	LwpMenuMtTblayout	&H272
LwpMenuMfPrintcancel	&H78	LwpMenuMtInscolorw	&H273
LwpMenuMfDummymrgsel	&H79	LwpMenuMtDelcolorw	&H274
LwpMenuMfDummymrgext	&H7A	LwpMenuMtEditfrmla	&H275
LwpMenuMfAsciiopts	&H7B	LwpMenuMtTbladd	&H276
LwpMenuMfCanprint	&H7C	LwpMenuMtHeading	&H277
LwpMenuMfFldrename	&H7D	LwpMenuMtTableinfox	&H279
LwpMenuMfPrintopt	&H7E	LwpMenuMtSizecolorw	&H27A
LwpMenuMfSaveasnw	&H7F	LwpMenuMtConnect	&H27B
LwpMenuMfImportnw	&H80	LwpMenuMtSavefrmla	&H27D
LwpMenuMfDocdescnw	&H81	LwpMenuMtQuickaddrow	&H283
LwpMenuMfMasternw	&H82	LwpMenuMtQuickaddcol	&H284
LwpMenuMfNwimport	&H83	LwpMenuMtDelcolumn	&H285
LwpMenuMfMergeviewprint	&H84	LwpMenuMtDelrow	&H286
LwpMenuMfLastopen1	&H85	LwpMenuMtInsertrow	&H287
LwpMenuMfLastopen2	&H86	LwpMenuMtInsertcol	&H288
LwpMenuMfLastopen3	&H87	LwpMenuMtInsertaccel	&H289
LwpMenuMfLastopen4	&H88	LwpMenuMtDeleteaccel	&H28A
LwpMenuMfLastopen5	&H89	LwpMenuMtDeltab	&H28B
LwpMenuMfOpennostyle	&H8A	LwpMenuMtSelectcolumn	&H28C
LwpMenuMfMergelabels	&H8B	LwpMenuMtSelectrow	&H28D
LwpMenuMfOtherflds	&H8C	LwpMenuMtSelecttable	&H28E
LwpMenuMfImportpicture	&H8D	LwpMenuMtChgiconsize	&H28F

LwpMenuMfExport	&H8E	LwpMenuMtDisconnect	&H290
LwpMenuMfAttribnw	&H8F	LwpMenuMtSplitcells	&H291
LwpMenuMfCanmergeprint	&H92	LwpMenuMtSplittable	&H292
LwpMenuMfImporttext	&H98	LwpMenuMtPcolinscolrow	&H293
LwpMenuMfMergeprintopts	&H235	LwpMenuMtPcolinsertrow	&H294
LwpMenuMfClose	&H90	LwpMenuMtPcolinsertcol	&H295
LwpMenuMfCloseall	&H2DB	LwpMenuMtPcoldeltable	&H296
LwpMenuMfCloseallreplacelast	&H2DC	LwpMenuMtPcoldelcolumn	&H297
LwpMenuMfPassword	&H91	LwpMenuMtPcoldelrow	&H298
LwpMenuMfPassword2	&H122	LwpMenuMtPcolconnect	&H299
LwpMenuMfMasterdocument	&H2E0	LwpMenuMtPcoldisconnect	&H29A
LwpMenuMfClosefile	&H93	LwpMenuMtPcolsplitcells	&H29B
LwpMenuMfLock	&H94	LwpMenuMtPcolselectcolumn	&H29C
LwpMenuMfNextmdiwindow	&H2D7	LwpMenuMtPcolselectrow	&H29D
LwpMenuMfMail	&HE5	LwpMenuMtPcolselecttable	&H29E
LwpMenuMfMailNew	&H2E1	LwpMenuMtPcoldelcolrow	&H29F
LwpMenuMfMailread	&HE6	LwpMenuMtPcolheading	&H2A0
LwpMenuMfQuickopen	&HE7	LwpMenuMtCellinfoebox	&H2A1
LwpMenuMfFilesep	&HEB	LwpMenuMtShowtablelineinfoebox	&H2A2
LwpMenuMfSavecopyas	&H124	LwpMenuMtShowtablesizeinfoebox	&H2A3
LwpMenuMfUpdateobject	&H125	LwpMenuMtCreatetablewithgrid	&H2A4
LwpMenuMfEditorgreet	&H2DD	LwpMenuMtShowpcollineinfoebox	&H2A5
LwpMenuMfNewdivision	&H2DE	LwpMenuMtShowpcolsizeinfoebox	&H2A6
LwpMenuMfOpendivision	&H2DF	LwpMenuMtSplitpcol	&H2A7
LwpMenuMfSaContinueRoute	&H2E2	LwpMenuMtPcolselectentiretable	&H2A8
LwpMenuMfSaRoute	&H2E3	LwpMenuMtSelectentiretable	&H2A9
LwpMenuMfSeldatafile	&H30C	LwpMenuMtSelectentirecellrange	&H2AA
LwpMenuMfMrgdelimit	&H30C	LwpMenuMtSelectentirepcolcellrange	&H2AB
LwpMenuMfMrgletter	&H30D	LwpMenuMtTabletopalign	&H2AC
LwpMenuMfMrgenvelope	&H312	LwpMenuMtTablecenteralign	&H2AD
LwpMenuMfMrggotorecord	&H30E	LwpMenuMtTablebottomalign	&H2AE
LwpMenuMfCreatedesc	&H30F	LwpMenuMtConnectrow	&H2AF
LwpMenuMfMrgsort	&H310	LwpMenuMtPcolconnectrow	&H2B0
LwpMenuMfCreatemerge	&H311	LwpMenuMtGotonextblock	&H2B1
LwpMenuMfEditdatafile	&H313	LwpMenuMtCellinfoeboxmouse	&H2B2
LwpMenuMfUsecurasmrgdoc	&H314	LwpMenuMtTableinfoeboxmouse	&H2B3
LwpMenuMfOpenandeditdatafile	&H315	LwpMenuMxRunrexx	&H2BB
LwpMenuMfOpenpreview	&H316	LwpMenuMxRunontime	&H2BC
LwpMenuMfNewpreview	&H317	LwpMenuMxHourglass	&H2D0
LwpMenuMxMergemacro	&H318	LwpMenuMxMacpause	&H2D1
LwpMenuMfMergeprintall	&H319	LwpMenuMxMacresume	&H2D2
LwpMenuMfMergeopen	&H31A	LwpMenuMxTimedelay	&H2D3
LwpMenuMIsRunscript	&H31B	LwpMenuMxTestforcancel	&H2D4
LwpMenuMIsCreatescript	&H35F	LwpMenuMxMacmessages	&H2D5
LwpMenuMIsCreatedialog	&H360	LwpMenuMxPreanswermmsgbox	&H2D6

LwpMenuMIsCompile	&H365	LwpMenuMxMaccancel	&H2D8
LwpMenuMIsAutoapp	&H369	LwpMenuMxMaccontinue	&H2D9
LwpMenuMIsIdeStepIn	&H3C0	LwpMenuMxMacrovariables	&H2DA
LwpMenuMIsIdeStepOver	&H3C1	LwpMenuMxRunautonew	&H31E
LwpMenuMIsIdeStepExit	&H3C2	LwpMenuMxRunautoopen	&H31F
LwpMenuMIsIdeContinue	&H3C3	LwpMenuOutlinemenu	&H2EE
LwpMenuMIsIdeStop	&H3C4	LwpMenuMolPromote	&H2EF
LwpMenuMIsIdeSetBp	&H3C5	LwpMenuMolDemote	&H2F0
LwpMenuMIsIdeDisBp	&H3C6	LwpMenuMolMoveup	&H2F1
LwpMenuMIsIdeClearAll	&H3C7	LwpMenuMolMovedown	&H2F2
LwpMenuMIsIdeDisAll	&H3C8	LwpMenuMolExpand	&H2F3
LwpMenuMIsIdeBreakpoints	&H3C9	LwpMenuMolContract	&H2F4
LwpMenuMIsIdeBrowser	&H3CA	LwpMenuMolShowlevels	&H2F5
LwpMenuMIsIdeOutput	&H3CB	LwpMenuMolExpandsingle	&H2F6
LwpMenuMIsIdeVariables	&H3CC	LwpMenuMolContractsingle	&H2F7
LwpMenuMIsIdeCheckScript	&H3CD	LwpMenuMolUseoutlinestyle	&H2F8
LwpMenuMIsIdeNewSub	&H3CE	LwpMenuMolSetshowlevel1	&H2F9
LwpMenuMIsIdeNewFun	&H3CF	LwpMenuMolSetshowlevel2	&H2FA
LwpMenuMIsIdeDebug	&H3D0	LwpMenuMolSetshowlevel3	&H2FB
LwpMenuMIsIdeRedo	&H3D1	LwpMenuMolSetshowlevel4	&H2FC
LwpMenuMIsShowide	&H3D2	LwpMenuMolSetshowlevel5	&H2FD
LwpMenuMIsIdeImport	&H3D3	LwpMenuMolSetshowlevel6	&H2FE
LwpMenuMIsIdeExport	&H3D4	LwpMenuMolSetshowlevel7	&H2FF
LwpMenuMIsIdeScriptPref	&H3D5	LwpMenuMolSetshowlevel8	&H300
LwpMenuMIsIdeCloseWin	&H3D6	LwpMenuMolSetshowlevel9	&H301
LwpMenuMIsIdeToggle	&H3D7	LwpMenuMolSetshowlevel10	&H302
LwpMenuMIsIdePrevSub	&H3D8	LwpMenuMolOutstylesequence	&H2ED
LwpMenuMIsIdeNextSub	&H3D9	LwpMenuMtTabkey	&H320
LwpMenuMIsIdeTogglebp	&H3DA	LwpMenuMeDefpathsnw	&H324
LwpMenuMIsIdeCodeWindow	&H3DB	LwpMenuMvHidepicts	&H325
LwpMenuMIsIdeCheckAll	&H3DC	LwpMenuMvHidemarks	&H326
LwpMenuMIsIdeSaveLso	&H3DD	LwpMenuMvHidetabrack	&H327
LwpMenuMIsIdeShowicon	&H3E0	LwpMenuMvColguide	&H328
LwpMenuMIsIdeRuncursub	&H3E1	LwpMenuMvShowcleanscreen	&H348
LwpMenuMIsIdeHelpLs	&H3E2	LwpMenuMvFixediconbar	&H34A
LwpMenuMIsIdeHelpWp	&H3E3	LwpMenuMvViewlevel	&H34B
LwpMenuMIsIdeHelpSe	&H3E4	LwpMenuMtbChange	&H32A
LwpMenuMIsIdeHelpAbout	&H3E5	LwpMenuMtbTabbox	&H32B
LwpMenuMIsIdeInsertLso	&H3E6	LwpMenuMvVertruler	&H32E
LwpMenuMIRenumnotes	&H95	LwpMenuMvShownotes	&H330
LwpMenuEditmenu	&H96	LwpMenuMvShowtabsandreturns	&H332
LwpMenuMeUndo	&H97	LwpMenuMvOut2lay	&H334
LwpMenuMeCut	&H99	LwpMenuMfImportdraw	&H335
LwpMenuMeCopy	&H9A	LwpMenuMIMacroendrecord	&H336
LwpMenuMePaste	&H9B	LwpMenuMvLay2out	&H337

LwpMenuMeUndoredo	&H9C	LwpMenuMvSetpreferences	&H338
LwpMenuMePastelinks	&H9D	LwpMenuMrCreatelastframe	&H339
LwpMenuMeInsvariable	&H9E	LwpMenuMIQuickendrecord	&H33A
LwpMenuMeFootnotes	&HA0	LwpMenuMrUngroupframes	&H33B
LwpMenuMeNotes	&HA1	LwpMenuMIRevmarkingoff	&H33C
LwpMenuMeNotes20	&H11A	LwpMenuMIFieldhide	&H33D
LwpMenuMeDefaults	&HA2	LwpMenuMcExtracton	&H346
LwpMenuMeTypeover	&HA3	LwpMenuMdivDivisionproperties	&H34C
LwpMenuMeNbDefaults	&HA4	LwpMenuMdivNewdivision	&H34D
LwpMenuMeInsertdate	&HA5	LwpMenuMdivCombinedivisions	&H34E
LwpMenuMeAddmergevar	&HA6	LwpMenuMdivDeletedivision	&H34F
LwpMenuMeAdddocdescvar	&HA7	LwpMenuMdivExpanddivision	&H350
LwpMenuMxMdiswitch	&HA8	LwpMenuMdivGroupdivision	&H356
LwpMenuMeSeclevaction	&HA9	LwpMenuMsecSectionproperties	&H351
LwpMenuMeDefpaths	&HAA	LwpMenuMsecNewsection	&H352
LwpMenuMeFootopts	&HAB	LwpMenuMsecCombinesections	&H353
LwpMenuMeRedo	&HAC	LwpMenuMsecSelectsection	&H354
LwpMenuMeDocoptions	&HAD	LwpMenuMsecExpandsection	&H355
LwpMenuMeDefpath	&HAE	LwpMenuMtabSettabs	&H357
LwpMenuMeDefsty	&HAF	LwpMenuMtabCleartabs	&H358
LwpMenuMeDefback	&HB0	LwpMenuMtabLefttab	&H359
LwpMenuMeFootnoteinsert	&HB1	LwpMenuMtabRighttab	&H35A
LwpMenuMeInsertnote	&HB2	LwpMenuMtabCentertab	&H35B
LwpMenuMeDelete	&HB3	LwpMenuMtabNumerictab	&H35E
LwpMenuMeIndexentry	&HB4	LwpMenuMtabDefaulttabs	&H361
LwpMenuMeAddnote	&HB5	LwpMenuMtabRemoveruler	&H362
LwpMenuMeMergevar	&HB6	LwpMenuMtabHideruler	&H363
LwpMenuMeDescvar	&HB7	LwpMenuMtabQuickright	&H3DE
LwpMenuMeFormatcheckbar	&HB8	LwpMenuMtabQuickcenter	&H3DF
LwpMenuMePageup	&HB9	LwpMenuMrSetmarginsinfolbox	&H366
LwpMenuMePagedown	&HBA	LwpMenuMfToggleinfolbox	&H367
LwpMenuMeRevinsertion	&HBB	LwpMenuMfCreateversion	&H368
LwpMenuMeProtected	&HBC	LwpMenuMfCreateversionDivision	&H389
LwpMenuMeNohyphen	&HBD	LwpMenuMvViewversion	&H370
LwpMenuMeGetglosname	&HC6	LwpMenuMafidHelp	&H384
LwpMenuMeMacpath	&H11E	LwpMenuMafidNotfound	&H385
LwpMenuMeTocentry	&H343	LwpMenuMfMailNextstop	&H386
LwpMenuMeBullet	&H344	LwpMenuMfMailEditroute	&H387
LwpMenuMeMergedatafile	&H232	LwpMenuMergeScriptNotify	&H38E
LwpMenuMeMergeextdesc	&H234	LwpMenuFirstpopupid	&H39D
LwpMenuMeMergevarsdatafile	&H236	LwpMenuRenderformat	&H35C
LwpMenuMePastespecial	&H24C	LwpMenuRenderallformats	&H35D
LwpMenuMeInsertobject	&H24D	LwpMenuMcCommenttoolsOn	&H379
LwpMenuMeChangelink	&H24E	LwpMenuMnDeletenote	&H37A
LwpMenuMeUseworkpath	&H24F	LwpMenuMnClosenote	&H37B

LwpMenuMeSelectword	&H253	LwpMenuMnNextnote	&H37C
LwpMenuMeSelectsentence	&H254	LwpMenuMnPrevnote	&H37D
LwpMenuMeSelectparagraph	&H255	LwpMenuMcCommenttools	&H37E
LwpMenuMeSelectfile	&H256	LwpMenuMcHighnote	&H37F
LwpMenuMeRevisionmode	&H257	LwpMenuMnCloseall	&H380
LwpMenuMIDoccontrol	&H258	LwpMenuMnOpenall	&H381
LwpMenuMfSharedoccontrol	&H25A	LwpMenuMnDeleteall	&H382
LwpMenuMIDoccontrolEditorpanel	&H25B	LwpMenuMnShowinitials	&H383
LwpMenuMIDoccontrolExternal	&H25C	LwpMenuMnNumbernotes	&H384
LwpMenuMvViewversionExternal	&H25D	LwpMenuMfQuickprint	&H3B2
LwpMenuMfDocdescExternal	&H25E	LwpMenuMdDebugger	&H3E8
LwpMenuMfMrglabel	&H25F	LwpMenuMdDebugDoc	&H3E9
LwpMenuMfMrgletterssetup	&H260	LwpMenuMdDebugPres	&H3EA
LwpMenuMfMrgenvelopesetup	&H261	LwpMenuMdDebugDivinfo	&H3EB
LwpMenuMfMrgviewprint	&H264	LwpMenuMdDebugCont	&H3EC
LwpMenuNwobjects	&HBE	LwpMenuMdDebugCol	&H3ED
LwpMenuMoOpennw	&HBF	LwpMenuMdDebugLine	&H3EE
LwpMenuMoCreatenw	&HC0	LwpMenuMdDebugVobj	&H3EF
LwpMenuMoListnw	&HC1	LwpMenuMdDebugHeap	&H3F0
LwpMenuMoLinksnw	&HC2	LwpMenuMdDebugMain	&H3F1
LwpMenuMoAttribnw	&HC3	LwpMenuMdDebugProf	&H3F2
LwpMenuViewmenu	&HC8	LwpMenuMdDebugVcnt	&H3F3
LwpMenuMvFit	&HC9	LwpMenuMdDebugDivision	&H3F4
LwpMenuMvWorking	&HCA	LwpMenuMdDebugLay	&H3F5
LwpMenuMv75	&H31C	LwpMenuMdDebugVermgr	&H3F6
LwpMenuMv150	&H31D	LwpMenuMdDebugVerctrl	&H3F7
LwpMenuMv100	&HCB	LwpMenuMdDebugDoclay	&H3F8
LwpMenuMv200	&HCC	LwpMenuMdDebugPara	&H3FA
LwpMenuMvSwitchiconstate	&HCD	LwpMenuMdDebugInfo	&H3FB
LwpMenuMvPagesort	&H303	LwpMenuMdDebugAtom	&H3FC
LwpMenuMvZoomoptions	&H304	LwpMenuMdDebugWrdarg	&H3FD
LwpMenuMvComment	&H306	LwpMenuMdDebugDwrdarg	&H3FE
LwpMenuMvShowheadfoot	&H307	LwpMenuMdDebugStrarg	&H3FF
LwpMenuMvCustom	&H305	LwpMenuMdDebugSline	&H400
LwpMenuMvFacingpages	&H308	LwpMenuMdDebugPhint	&H401
LwpMenuMvMultiplepages	&H309	LwpMenuMdDebugChint	&H402
LwpMenuMvPagemargin	&H30A	LwpMenuMdDebugMsgtrace	&H403
LwpMenuMvPagewidth	&H30B	LwpMenuMdDebugFoundry	&H404
LwpMenuMvSpecialview	&H323	LwpMenuMdDebugCounts	&H405
LwpMenuMvClearspecialview	&H371	LwpMenuMdDebugPresinfo	&H406
LwpMenuMvSynchronizeviews	&H364	LwpMenuMdDebugMeter	&H407
LwpMenuMvClearhighlight	&H36A	LwpMenuMdDebugStack	&H408
LwpMenuMvPreferences	&HCE	LwpMenuMdDebugPtrholder	&H409
LwpMenuMvTabrack	&HCF	LwpMenuMdDebugMemory	&H40A
LwpMenuMvFunctionbar	&HD0	LwpMenuMdDebugMemorydump	&H40B

LwpMenuMvChgsidebar	&HD1	LwpMenuMdDebugContent	&H40C
LwpMenuMvLayout	&HD2	LwpMenuMdDebugSilver	&H40D
LwpMenuMvDraft	&HD3	LwpMenuMdDebugVernode	&H40E
LwpMenuMvFacing	&HD4	LwpMenuMdDebugCheckpoint	&H40F
LwpMenuMvFacecancel	&HD6	LwpMenuMdDebugSocket	&H410
LwpMenuMvShowpicts	&HD7	LwpMenuMdDebugSfb	&H411
LwpMenuMvShowmarks	&HD8	LwpMenuMdDebugDemandload	&H412
LwpMenuMvPreferdefaults	&HD9	LwpMenuMdDebugNumberhint	&H413
LwpMenuMvOutline	&HDA	LwpMenuMdDebugDialog	&H414
LwpMenuMvFloatingicon	&HDB	LwpMenuMdDebugBack	&H415
LwpMenuMvLefticon	&HDC	LwpMenuMdDebugHeadtailholder	&H416
LwpMenuMvRighticon	&HDD	LwpMenuMdDebugNotifypersistent	&H417
LwpMenuMvTopicon	&HDE	LwpMenuMdDebugObjectholder	&H418
LwpMenuMvBottomicon	&HDF	LwpMenuMdDebugEnumobjhead	&H419
LwpMenuMvSmarticons	&HE8	LwpMenuMdDebugNotify	&H41A
LwpMenuMvFullscreen	&HE9	LwpMenuMdDebugNamedobjectholde r	&H41B
LwpMenuMvFullscreenoptions	&HEA	LwpMenuMdDebugSection	&H41C
LwpMenuMvEditicon	&HEC	LwpMenuMdDebugFontmgr	&H41D
LwpMenuMfPrintenvelope	&HED	LwpMenuMdDebugSanity	&H41E
LwpMenuMvNexticonset	&HEF	LwpMenuMdDebugAtomWatch	&H41F
LwpMenuMvSaveicon	&HF0	LwpMenuMdDebugAtomCount	&H420
LwpMenuMvCreateicon	&HF1	LwpMenuMdDebugPieceCount	&H421
LwpMenuMvFullscreenstartup	&HF2	LwpMenuMdDebugObjectSizes	&H422
LwpMenuMvSectiontabs	&HF4	LwpMenuMdDebugTextstyle	&H423
LwpMenuCharmenu	&HFA	LwpMenuMdDebugEnd	&H44B
LwpMenuMcTextinfobox	&H11B	LwpMenuMkPageup	&H44C
LwpMenuMcTextinfoboxmouse	&H262	LwpMenuMkPagedown	&H44D
LwpMenuMcDrawinfoboxmouse	&H263	LwpMenuMcDooledragdrop	&H44E
LwpMenuMcFont	&HFB	LwpMenuMsMplMargins	&H514
LwpMenuMcNormal	&HFD	LwpMenuMsMplPage	&H515
LwpMenuMcBold	&HFE	LwpMenuMsMplFooter	&H516
LwpMenuMcItalic	&HFF	LwpMenuMsMplHeader	&H517
LwpMenuMcUnderline	&H100	LwpMenuMsMplLines	&H518
LwpMenuMcWordunder	&H101	LwpMenuMrMflType	&H578
LwpMenuMcLeft	&H103	LwpMenuMrMflSizepos	&H579
LwpMenuMcCenter	&H104	LwpMenuMrMflLines	&H57A
LwpMenuMcRight	&H105	LwpMenuMrMflColumnstabs	&H57B
LwpMenuMcJustify	&H106	LwpMenuMISetmacrofile	&H5DC
LwpMenuMcAlignment	&H107	LwpMenuMICompilemacro	&H5DD
LwpMenuMcCaps	&H108	LwpMenuMIMacrorun	&H5DE
LwpMenuMcIndentall	&HF7	LwpMenuMIMacroappend	&H5DF
LwpMenuMcIndentfirst	&HF8	LwpMenuMIPrototype	&H5E0
LwpMenuMcIndentrest	&HF9	LwpMenuMeSmartcorrect	&H5E1
LwpMenuMcExtract	&HF6	LwpMenuMrChart	&H5E2
LwpMenuMcMacrofastformat	&HEE	LwpMenuMrCharttodraw	&H5E3

LwpMenuMcFontvert	&HF3	LwpMenuMmCascades	&H640
LwpMenuMcEnlargetext	&HFC	LwpMenuSubmenuInsert	&H640
LwpMenuMcReducetext	&HF5	LwpMenuMmFields	&H641
LwpMenuMcIndent	&H102	LwpMenuSubmenuMarktextas	&H642
LwpMenuMcOutdent	&H11C	LwpMenuSubmenuAlignment	&H643
LwpMenuMcBulletpanel	&H11D	LwpMenuMmCaps	&H644
LwpMenuMcNextfacename	&H3B6	LwpMenuMmMacros	&H647
LwpMenuMcNextalignment	&H3B7	LwpMenuSubmenuInsertsmartsum	&H649
LwpMenuMcNextattribute	&H3B8	LwpMenuMMailmemo	&H64A
LwpMenuMcNextbullet	&H3B9	LwpMenuMPrinttofile	&H64B
LwpMenuMcNextnumbering	&H3BA	LwpMenuSubmenuUsersetup	&H64C
LwpMenuMcNextstyle	&H3BB	LwpMenuSubmenuScripts	&H64D
LwpMenuMcDoubleunder	&H10E	LwpMenuMmShowoptions	&H64E
LwpMenuMcSuperscript	&H10F	LwpMenuSubmenuQuickdemo	&H64F
LwpMenuMcSubscript	&H110	LwpMenuSubmenuOtherhelpstuff	&H650
LwpMenuMcAllcaps	&H111	LwpMenuMmOtherframe	&H651
LwpMenuMfEnvelopeprint	&H112	LwpMenuMmOthertable	&H652
LwpMenuMcAllower	&H11F	LwpMenuSubmenuOthertextstuff	&H653
LwpMenuMcInitcaps	&H120	LwpMenuSubmenuPriority	&H654
LwpMenuMcSmallcaps	&H121	LwpMenuMmOtherstuff	&H655
LwpMenuMcJustfont	&H116	LwpMenuSubmenuSelect	&H656
LwpMenuMcJustpointsize	&H117	LwpMenuSubmenuOtherdocpart	&H657
LwpMenuMcStrikethru	&H118	LwpMenuSubmenuAttributes	&H658
LwpMenuMcFastformat	&H123	LwpMenuSubmenuOutline	&H659
LwpMenuMcHighlight	&H126	LwpMenuSubmenuCollapse	&H65A
LwpMenuStylemenu	&H12C	LwpMenuSubmenuEdit	&H65B
LwpMenuMsCreate	&H12D	LwpMenuSubmenuDebug	&H65C
LwpMenuMsStylemanage	&H12F	LwpMenuSubmenuDocprops	&H65D
LwpMenuMsChgstyle	&H130	LwpMenuSubmenuZoomto	&H65E
LwpMenuMsSavestyle	&H131	LwpMenuSubmenuShowhide	&H65F
LwpMenuMsSelectstyle	&H132	LwpMenuSubmenuOtherviewstuff	&H660
LwpMenuMsOutlinestyle	&H140	LwpMenuSubmenuNamedstyles	&H661
LwpMenuMsStylemanageinit	&H141	LwpMenuSubmenuDelete	&H662
LwpMenuMsStylemanageaction	&H142	LwpMenuSubmenuFile	&H663
LwpMenuMsChangestylename	&H143	LwpMenuSubmenuView	&H664
LwpMenuMsDefinestyle	&H144	LwpMenuSubmenuTeammail	&H665
LwpMenuMsModcreatestyle	&H145	LwpMenuSubmenuEditmodemenu	&H666
LwpMenuMsStylemanagefinish	&H146	LwpMenuSubmenuDebugmodemenu	&H667
LwpMenuMsRedefinestyle	&H150	LwpMenuSubmenuHelp	&H668
LwpMenuMsMargins	&H148	LwpMenuCreatemenu	&H672
LwpMenuMsPagesettings	&H149	LwpMenuParallelstylemenu	&H67A
LwpMenuMsApply	&H14A	LwpMenuEnvAddress	&H67D
LwpMenuMsStylemru1	&H14B	LwpMenuEnvSave	&H681
LwpMenuMsStylemru2	&H14C	LwpMenuEnvPrint	&H682
LwpMenuMsStylemru3	&H14D	LwpMenuMfCreatetabdivision	&H686

LwpMenuMsStylemru4	&H14E	LwpMenuMfCreatedivision	&H683
LwpMenuMrManuatable	&H14F	LwpMenuMeTocassist	&H684
LwpMenuMrFrameanchorinfo	&H187	LwpMenuMpNewsection	&H685
LwpMenuMrFramelineinfo	&H188	LwpMenuMtInsertclickhere	&H687
LwpMenuMrFramebackinfo	&H189	LwpMenuMcTextlinepanel	&H688
LwpMenuMrFramegfxinfo	&H18A	LwpMenuMvOtherviewstuff	&H689
LwpMenuMrFrametopalign	&H18B	LwpMenuMoApplystyle	&H68A
LwpMenuMrFramecenteralign	&H18C	LwpMenuMsFramecreate	&H68B
LwpMenuMrFramebottomalign	&H18D	LwpMenuMsFrameredefinestyle	&H68C
LwpMenuMrDeleteframe	&H18F	LwpMenuMsTablecreate	&H68D
LwpMenuPagemenu	&H15E	LwpMenuMsTableredefinestyle	&H68E
LwpMenuMpCreatelayout	&H15F	LwpMenuMsTablecellcreate	&H68F
LwpMenuMpModifylayout	&H160	LwpMenuMsTablecellredefinestyle	&H690
LwpMenuMpTabs	&H163	LwpMenuMvShowmisspelled	&H691
LwpMenuMpPagenum	&H164	LwpMenuMtDelrowacc	&H692
LwpMenuMpPagebreak	&H166	LwpMenuMtCtrlretacc	&H693
LwpMenuMpLinenum	&H167	LwpMenuMsQuickredefinestyle	&H694
LwpMenuMpHeadfoot	&H168	LwpMenuMtInsertrowacc	&H695
LwpMenuMpModlayoutinit	&H169	LwpMenuMrOleconvert	&H696
LwpMenuMpModlaypagesettings	&H16A	LwpMenuMrOleconvertandactivate	&H697
LwpMenuMpModlayrightfacepage	&H16B	LwpMenuMrFrameleft	&H698
LwpMenuMpModlayrightheadfacepage	&H16C	LwpMenuMrFramecenter	&H699
LwpMenuMpModlayrightfooterfacepage	&H16D	LwpMenuMrFrameright	&H69A
LwpMenuMpModlayrightlines	&H16E	LwpMenuMrFramehorzspan	&H69B
LwpMenuMpModlayleftfacepage	&H16F	LwpMenuMrFramevertspan	&H69C
LwpMenuMpModlayleftheaderfacepage	&H170	LwpMenuMfNewanytab	&H69D
LwpMenuMpModlayleftfooterfacepage	&H171	LwpMenuMfOdmaopen	&H69E
LwpMenuMpModlayleftlines	&H172	LwpMenuMfOdmaininsert	&H69F
LwpMenuMpDellayout	&H175	LwpMenuMfOdmaconvert	&H6A0
LwpMenuMpRevertlayout	&H176	LwpMenuMfOdmasaveas	&H6A1
LwpMenuMpInslayout	&H177	LwpMenuMfOdmaexternalize	&H6A2
LwpMenuMpModlayoutfinish	&H178	LwpMenuMfOdmaexport	&H6A3
LwpMenuMpHeaderfootergoto	&H179	LwpMenuMrOleverb	&H6A4
LwpMenuMpMacinslayout	&H17A	LwpMenuMrOleverbmax	&H707
LwpMenuMpParallel	&H17B	LwpMenuMrOledivtabverb	&H708
LwpMenuMpPcolinfo	&H17C	LwpMenuMrOledivtabverbmax	&H76B
LwpMenuMpPageproperties	&H17D	LwpMenuMtSelectcell	&H76C
LwpMenuMpCreate	&H17E	LwpMenuMtPcolselectcell	&H76D
LwpMenuMpNewpagestyle	&H17F	LwpMenuMpColumnbreak	&H76E
LwpMenuMpHeaderproperties	&H180	LwpMenuMeViewtablegrid	&H76F
LwpMenuMpFooterproperties	&H181	LwpMenuMeViewtableheading	&H770
LwpMenuMpPagepropertiesmouse	&H182	LwpMenuMeViewpcolguides	&H771
LwpMenuMpFooterpropertiesmouse	&H183	LwpMenuMeViewmarginguides	&H772
LwpMenuMpHeaderpropertiesmouse	&H184	LwpMenuMeViewpagegauge	&H773
LwpMenuMpPcolinfo	&H185	LwpMenuMeViewshowallmarks	&H774

LwpMenuFramemenu	&H190	LwpMenuMeViewhideallmarks	&H775
LwpMenuMrFrametext	&H192	LwpMenuMtTableleft	&H776
LwpMenuMrScale	&H193	LwpMenuMtTablecenter	&H777
LwpMenuMrGroupframes	&H194	LwpMenuMtTableright	&H778
LwpMenuMrFrameinfobox	&H195	LwpMenuMtTablehorzspan	&H779
LwpMenuMrBringtofront	&H197	LwpMenuMtGotoinfobox	&H77A
LwpMenuMrSendtoback	&H198	LwpMenuMhTour	&H77B
LwpMenuMrAdd	&H199	LwpMenuMfCreatedivision	&H77D
LwpMenuMrSendbackone	&H19A	LwpMenuMfToggleclickhereprompts	&H77E
LwpMenuMrDrawingicon	&H19B	LwpMenuMcToggleskipbullet	&H77F
LwpMenuMrChartingicon	&H19C	LwpMenuMlSkipmisspelledword	&H780
LwpMenuMrCreateframe	&H19D	LwpMenuMlSkipallmisspelledword	&H781
LwpMenuMrAddframewithdlg	&H19E	LwpMenuMlAddtouserdictionary	&H782
LwpMenuMrFramedefaults	&H19F	LwpMenuMlTogglebubblehelp	&H783
LwpMenuMrFramelines	&H1A0	LwpMenuMfPlainnew	&H784
LwpMenuMrFramestype	&H1A1	LwpMenuMvToggleshowanchors	&H785
LwpMenuMrFrameborders	&H1A2	LwpMenuMvNextnewestversion	&H786
LwpMenuMrFramecolumns	&H1A3	LwpMenuMvMostrecentversion	&H787
LwpMenuMrModframereflow	&H1A4	LwpMenuMvOldestversion	&H788
LwpMenuMrModframeinit	&H1A5	LwpMenuMcJustifyall	&H789
LwpMenuMrBringfrontone	&H186	LwpMenuMcClosegraphicole	&H78A
LwpMenuMrSelectframe	&H18E	LwpMenuMfFramerevert	&H78B
LwpMenuMrEquationsicon	&H1A7	LwpMenuMtCellrevert	&H78C
LwpMenuMrImageprocessing	&H1A8	LwpMenuMfOpentabdivision	&H78D
LwpMenuMrFrameinfoboxmouse	&H26C	LwpMenuMfNewtabdivision	&H78E
LwpMenuMrFramelineinfoboxmouse	&H26D	LwpMenuMtInsertclickherelink	&H78F
LwpMenuMrFramebackinfoboxmouse	&H26E	LwpMenuMtEditclickherelink	&H790
LwpMenuMrFramegfxinfoboxmouse	&H26F	LwpMenuMsecDeletesection	&H79D
LwpMenuMrFrameanchorinfoboxmouse	&H270	LwpMenuMfExternalfileexit	&H79E
LwpMenuMrExpert	&H38F	LwpMenuMfExternalfilesavedivision	&H79F
LwpMenuMrExpertmax	&H39C	LwpMenuMfFtpopen	&H7A0
LwpMenuMrWppack	&H15D	LwpMenuMfFtpinsert	&H7A1
LwpMenuMrAfid	&H1A9	LwpMenuMfFtpconvert	&H7A2
LwpMenuMrAfidmax	&H1C1	LwpMenuMfFtpsaveas	&H7A3
LwpMenuToolmenu	&H1C2	LwpMenuMfFtpexternalize	&H7A4
LwpMenuMlSearch	&H1C3	LwpMenuSubmenuInternet	&H7A5
LwpMenuMlGoto	&H1C4	LwpMenuMfFtpoptions	&H7A6
LwpMenuMlSpellopts	&H1C5	LwpMenuMfOdmashowattributes	&H7A7
LwpMenuMlUserdict	&H1C6	LwpMenuMrRuby	&H7A8
LwpMenuMlThesaurus	&H1C7	LwpMenuMfFtpopendivision	&H7A9
LwpMenuMlSort	&H1C8	LwpMenuMfFtpopenurl	&H7AA
LwpMenuMlTables	&H1C9	LwpMenuMlShowfrommenu	&H7AB
LwpMenuMlGotolast	&H1CA	LwpMenuMlHideinfobox	&H7AC
LwpMenuMlSpellopts2	&H1CB	LwpMenuMfWelcome	&H7AD
LwpMenuMlRevmarkoptions	&H1CD	LwpMenuWp5Expert	&H7AE

LwpMenuMIToc	&H1CE	LwpMenuWp6Expert	&H7AF
LwpMenuMIAssignmacro	&H1CF	LwpMenuWpWelcome	&H7B0
LwpMenuMIMacros	&H1D0	LwpMenuMsQuickredefineframestyle	&H7B1
LwpMenuMIMacroaccel	&H1A6	LwpMenuMsQuickredefinetablestyle	&H7B2
LwpMenuMIGeneratetoc	&H1D1	LwpMenuMsQuickredefinecellstyle	&H7B3
LwpMenuMIIndexnextfile	&H119	LwpMenuMsQuickredefinestyle	&H7B4
LwpMenuMIBookmark	&H1D2	LwpMenuMeInsertsymbol	&H7B5
LwpMenuMIGlossary	&H1D3	LwpMenuMeWordcount	&H7B6
LwpMenuMIGlosset	&H1D4	LwpMenuMeDropcaps	&H7B7
LwpMenuMIMacpause	&H1D5	LwpMenuMfOdmaintportpicture	&H7B8
LwpMenuMIGenerateindex	&H1D6	LwpMenuMrOleobjectinabox	&H7B9
LwpMenuMISpelladdict	&H1D7	LwpMenuMcInternettoolson	&H7BA
LwpMenuMISpellskipall	&H1D8	LwpMenuMcInternettools	&H7BB
LwpMenuMISpellreplace	&H1DA	LwpMenuMhLotushomepage	&H7BC
LwpMenuMISpell	&H1DB	LwpMenuMhLotussupport	&H7BD
LwpMenuMISpellinit	&H1DC	LwpMenuMhLotusftpsite	&H7BE
LwpMenuMISpellcheck	&H1DD	LwpMenuMrLinkframes	&H7BF
LwpMenuMISpellcancel	&H1DE	LwpMenuMrUnlinkframes	&H7C0
LwpMenuMISpellskip	&H1DF	LwpMenuMfMrgwelcome	&H7C1
LwpMenuMIReplace	&H1E0	LwpMenuMfHtmloptions	&H7C2
LwpMenuMIChglang	&H1E1	LwpMenuMcUpdateindex	&H7C3
LwpMenuMIRenumseq	&H1E2	LwpMenuMcUpdatetoc	&H7C4
LwpMenuMISrattr	&H1E5	LwpMenuMvDesignmode	&H7C5
LwpMenuMIRepopts	&H1E6	LwpMenuMfNosopen	&H7C6
LwpMenuMISropts	&H1E7	LwpMenuMfNossaveas	&H7C7
LwpMenuMISearchonly	&H1E8	LwpMenuMfMailSendmessage	&H7C8
LwpMenuMIMacplay	&H1E9	LwpMenuMcInsertdefnumber	&H7C9
LwpMenuMIAssign	&H1EA	LwpMenuMcInsertdefbullet	&H7CA
LwpMenuMIACdde	&H1EB	LwpMenuMeTogglesmartselect	&H7CB
LwpMenuMIACtype	&H1EC	LwpMenuMcCreateobjectviewer	&H7CC
LwpMenuMICall	&H1ED	LwpMenuMcCreateobjectcomment	&H7CD
LwpMenuMISammy2	&H1EE	LwpMenuMcCreateobjectdraw	&H7CE
LwpMenuMITocopts	&H1EF	LwpMenuMcCreateobjectschedule	&H7CF
LwpMenuMISrfindmatch	&H1F0	LwpMenuMcCreateobjectsheetspreadsheet	&H7D0
LwpMenuMISrreplacethenfind	&H1F1	LwpMenuMcCreateobjectchart	&H7D1
LwpMenuMISrcancel	&H1F2	LwpMenuMhInternetsearch	&H7D2
LwpMenuMISrinit	&H1F3	LwpMenuMhLotusSMARTsuite	&H7D3
LwpMenuMIDoccompare	&H1F4	LwpMenuMcClickherekeyword	&H7D4
LwpMenuMIACedit	&H1F5	LwpMenuMfPublishweb	&H7D5
LwpMenuMIACplay	&H1F6	LwpMenuMICrossreference	&H7D6
LwpMenuMIACrecord	&H1F7	LwpMenuMtabHidevertruler	&H7D7
LwpMenuMIACoptions	&H1F8	LwpMenuMpUpdateheaders	&H7D8
LwpMenuMIACquickrec	&H1F9	LwpMenuMpUpdatefooters	&H7D9
LwpMenuMIACquickplay	&H1FA	LwpMenuMrRubyAbove	&HBB8
LwpMenuMIACgrammar	&H1FB	LwpMenuMrRubyBelow	&HBB9

LwpMenuMIGrammaropts	&H1FC	LwpMenuMrRubyDelete	&HBBA
LwpMenuMIGrammarstart	&H1FD	LwpMenuMxMinmacromenu	&H2328
LwpMenuMIGrammarinit	&H1FE	LwpMenuMxMaxmacromenu	&H238B
LwpMenuMIModcreatorule	&H1FF	LwpMenuMrAfidmenu	&H2710
LwpMenuMIGrammarnext	&H200	LwpMenuMrAfidmenumax	&H3E80
LwpMenuMIGrammarcancel	&H201	LwpMenuMaNotesflow	&H3E81
LwpMenuMIGrammarsugtext	&H202	LwpMenuMaNotesflowmax	&H3F48
LwpMenuMIGrammaroptionstext	&H203	LwpMenuFcsTextmenu	&H0
LwpMenuMIGrammarchange	&H204	LwpMenuFcsFramemenu	&H1
LwpMenuMIGrammarstats	&H205	LwpMenuFcsFrametextmenu	&H2
LwpMenuMIIconpath	&H206	LwpMenuFcsTabletextmenu	&H3
LwpMenuMICyclekeysetup	&H207	LwpMenuFcsFramegraphicmenu	&H4
LwpMenuMISmartfill	&H208	LwpMenuFcsPowerfieldmenu	&H5
LwpMenuMINewspell	&H209	LwpMenuFcsParallelcoltextmenu	&H6
LwpMenuMIRevisionbar	&H20A	LwpMenuFcsDivisionmenu	&H7
LwpMenuMIToa	&H20B	LwpMenuFcsSectionmenu	&H8
LwpMenuWindowmenu	&H4	LwpMenuFcsTablemenu	&H9
LwpMenuMwNewwindow	&H20D	LwpMenuFcsParallelcolmenu	&HA
LwpMenuMwTilewindow	&H20E	LwpMenuFcsTablegraphicmenu	&HB
LwpMenuMwCascadewindow	&H20F	LwpMenuFcsParallelcolgraphicmenu	&HC
LwpMenuMwStartoffiles	&H210	LwpMenuFcsHeadermenu	&HD
LwpMenuMwTilewindowhorz	&H246	LwpMenuFcsFootermenu	&HE
LwpMenuMwSplitvertwindow	&H251	LwpMenuFcsRulermenu	&HF
LwpMenuMwSplithorzwindow	&H252	LwpMenuFcsNotemenu	&H10
LwpMenuMwEndoffiles	&H224	LwpMenuFcsOutlinemenu	&H11
LwpMenuMwFilesseparator	&H225	LwpMenuFcsOledivisionmenu	&H12
LwpMenuMIIMacresume	&H22A	LwpMenuFcsFramechartmenu	&H13
LwpMenuMIReviewrevs	&H22D	LwpMenuFcsTablechartmenu	&H14
LwpMenuMIRevacceptall	&H36B	LwpMenuFcsParallelcolchartmenu	&H15
LwpMenuMIRevcancelall	&H36C	LwpMenuFcsTablecornermenu	&H16
LwpMenuMIRevaccept	&H36D	LwpMenuFcsTablerowmenu	&H17
LwpMenuMIRevcancel	&H36E	LwpMenuFcsTablecolumnmenu	&H18
LwpMenuMIField	&H238	LwpMenuFcsVertrulermenu	&H19
LwpMenuMIFieldnext	&H239	LwpMenuFcsRubymenu	&H1A
LwpMenuMIFieldprev	&H23A	LwpMenuFcsFootnotetextmenu	&H28
LwpMenuMIFieldeval	&H23B	LwpMenuFcsIndextextmenu	&H29
LwpMenuMIFieldadd	&H23C	LwpMenuFcsToctextmenu	&H2A
LwpMenuMIFieldupdate	&H23D	LwpMenuFcsTablecornermenu	&H2B
LwpMenuMIFieldtog	&H10A	LwpMenuFcsTabletextmenu	&H2C
LwpMenuMIFieldupall	&H10B	LwpMenuFcsFrametextmenudropcap	&H2D
LwpMenuMIFieldlock	&H10C	LwpMenuFcsFramemenudropcap	&H2E
LwpMenuMIFieldsave	&H10D		

Overview: Word Pro Text Subobjects

In Word Pro, text can appear in many places and in many forms, including the text which flows from page to page, the text within a table cell, or the text within a frame. LotusScript sees each of these text streams as a separate Text object and each Text object is based on the Text class. But each Text object can include as many as 27 distinct types of subobjects. Eleven of these text subobjects have their own LotusScript classes. The remaining 16 subobjects can only be detected or used by special methods found on each Text object.

Text subobjects

Text subobjects come in many forms. Some are simple, such as non-breaking spaces, soft hyphens, or tabs. Others are much more complex, such as text, comment notes, and tables. For example, the following sentence is comprised of four distinct subobjects:

```
My summer vacation:[TAB]What a trip![End Of Paragraph]
```

While the entire sentence might be a single Text object or part of a larger Text object, the sentence itself is comprised of two Text subobjects, one Tab subobject, and one EOP subobject, as shown in the table below.

<u>Subobject</u>	<u>Subobject Type</u>
My summer vacation:	Text
[TAB marker]	Tab
What a trip!	Text
[End Of Paragraph]	EOP

All together, there are 27 different types of subobjects. Most users will never need to distinguish one subobject from another. But for those who do, there are three methods in the Text class which are designed specifically for locating and navigating these subobjects. The methods are:

GetText

Returns the type for the subobject located at the cursor.

Next

Selects the next subobject of the type you specify.

Previous

Selects the previous subobject of the type you specify.

These methods are defined in the Text class and available on every Text object. The different types of subobjects are listed in the table below.

<u>Subobject Name</u>	<u>Description</u>
AnchoredFrame*	A frame which is placed "In text," "In text - vertical," or "With paragraph above."
Bookmark*	The marker for either the start or end of a bookmark range.
ClickHereBlock*	The marker for either the start or end of a Click Here block.
ColumnBreak	A hard column break within a multi-column page layout; created by choosing Text - Insert Other - Column Break.
DDE*	The marker for either the start or end of a DDE link.
DocVariable	A DocInfo field
EOP	The marker at the end of a paragraph created by pressing ENTER.
FootnoteMark*	The marker for either the start or end of a footnote.
HardSpace	A non-breaking space created by pressing CTRL+Spacebar.
HKatakana	A special Asian language character; seen only in Asian language versions of Word Pro.
Kanji	A special Asian language character; seen only in Asian language versions of Word Pro.
LineBreak	A soft line break created by pressing SHIFT+ENTER.

Note*	A Comment Note created by choosing Create - Comment Note.
PageBreak	A hard page break created by pressing CTRL+ENTER.
PageNumber	A page number created by choosing Text - Insert Page Number.
ParaNumber	Any number, bullet, or other text which is assigned to a paragraph as part of a numbering/bullet style. Note that any outline number sequence which uses decimal notation (for example, 1.1.1, 1.1.2, 1.1.3, and so on) contains one ParaNumber subobject for each digit in the sequence, and one Text subobject for each decimal in the sequence. These subobjects are separate from the subobject(s) which comprise the text of the paragraph itself.
PowerField*	The marker for either the start or end of a PowerField.
RubyFrame*	Seen only in Asian language versions of Word Pro.
RubyMarker*	Seen only in Asian language versions of Word Pro.
Section*	A section marker which defines the separation of two sections within a Word Pro document.
SoftHyphen	A hyphen which does not appear unless the hyphenated word flows to the end of a line where it would break naturally.
SpecialTab	SpecialTabs contain their own tab stop and alignment information. Normal tab markers derive their tab stop and alignment information from the ruler. SpecialTab markers are created in two ways: the InsertSpecialTab method or by importing a document from a file format which uses SpecialTabs (such as WordPerfect).
Tab	A normal tab marker created by pressing TAB.
Table*	A table which is placed "In text," "In text - vertical," or "With paragraph above."
Text	Text characters as seen in the user's code page. Users running the US English version of Word Pro under Windows 95 usually uses the ANSI 1252 code page.
Tombstone	A revision marker which displays the initials of the author of a particular revision.
Unicode	A special foreign-language type of character used in multi-language documents.

Note Some of the subobjects in the table above are marked with an asterisk (*). Each of these subobjects is complex or useful enough to merit its own class in the Word Pro LotusScript object model. See section titled "Subobjects with their own LotusScript classes" below.

Subobject Modifiers

Modifiers are Word Pro's way of keeping track of any changes you make to subobjects. And while most modifiers have no affect on certain types of subobjects, they can be applied to anything. For example, you can select a frame and some text and change the font from Helvetica to Arial. Word Pro would then mark both the Text subobject and the Frame subobject with a Font modifier. But only the Text subobject would show any change. If you selected just the frame, the status bar would show Arial as the applied font, but that Font modifier would have no effect on the appearance of the frame itself.

There are six type of modifiers in Word Pro. In most cases, only the subobjects we usually think of as being part of our text would be affected by these modifiers.

<u>Modifier</u>	<u>Action which applies the modifier</u>
Font	Changing any of the following font characteristics on a subobject: font face, font size, font color, bold, italic,

	underline, word underline, double underline, superscript, subscript, strikethrough, small caps, no hyphen, hidden, protected, upper case, or lower case.
Revision	Making a revision to a subobject while Markup Edits is turned on (choose Edit - Markup Edits).
Character Style	Changing the character style assigned to a subobject.
Attributes	Applied to a subobject if: <ul style="list-style-type: none"> • you tell the spell-checker to skip it. • the spell-checker marks it as misspelled. • the spell-checker marks it as a double word. • the grammar checker marks it as a grammar error. • you apply one or more of the following attributes to the subobject: no hyphen, hidden, protected.
Language	Changing the language for a subobject. This can be done from the Misc. panel of the Text InfoBox.
Code Page	Changing the code page of a subobject. In Word Pro, you can accomplish this by choosing File - User Setup - Word Pro Preferences. Then specify the "layout is set by text's language" option on the General panel of the Word Pro Preferences dialog box. This will cause Word Pro to change the keyboard layout, and hence the code page, to match that of the language assigned to the text.
Editor	Automatically applied to ALL subobjects. Initially, all subobjects are marked as belonging to the editor who created the document. Each time someone new makes a change to a document, Word Pro changes this modifier to indicate which subobjects were changed or added by each editor.

Modifiers and the Text subobject

Word Pro uses the Text subobject to refer to any group of adjacent ASCII text characters which share the same modifiers. That means that all the plain characters that are next to each other form a single Text subobject. Whenever the characters take on other modifiers such as a font or attributes modifier, Word Pro sees another distinct Text subobject.

For example, let us examine the following sentence:

My summer vacation: What a trip!

<u>Subobject</u>	<u>Subobject Type</u>	<u>Modifier Type</u>
My summer vacation: What a trip!	Text	none

This sentence is seen by Word Pro and LotusScript as a single Text subobject because there are no other types of subobjects within the text stream and there are no modifiers assigned to the characters. Now look at this sentence:

My summer vacation: *What a trip!*

Bold and Italic are font modifiers and they cause Word Pro to see five distinct Text subobjects as described in the table below.

<u>Subobject</u>	<u>Subobject Type</u>	<u>Modifier Type</u>
My	Text	Font (Bold)
summer vacation:	Text	none
<i>What</i>	Text	Font (Italic)
a trip!	Text	none
<i>trip!</i>	Text	Font (Bold & Italic)

Each group of characters which shares the same modifiers is seen as a single Text subobject.

Caution Word Pro makes every effort keep track of the creation and deletion of all subobjects. However, in the case of Text subobjects where modifiers can mean the difference between one subobject and multiple subobjects, Word Pro may not reconnect Text subobjects when modifiers are removed or made to match. For example, if you apply the

Bold modifier to a single word and then apply the same modifier to the rest of the sentence, Word Pro will attempt to reconnect the word and the sentence as a single Text subobject with a bold modifier. However, if the attempt fails, Word Pro will continue to see the word and the rest of the sentence as separate subobjects even though they share the same modifier.

Subobjects with their own LotusScript classes

While most subobjects are too small or simple to warrant their own LotusScript classes, there are eleven subobjects which are also represented by their own LotusScript classes. The table below lists these eleven subobjects and their corresponding LotusScript class names.

<u>Subobject Name</u>	<u>Class Name of Corresponding LotusScript Object</u>
Table	TableLayout, SuperTableLayout
AnchoredFrame	FrameLayout
FootnoteMark	Footnote
Section	Section
Note	NoteLayout
Bookmark	BookMark
DDE	DdeLink
PowerField	PowerField
ClickHereBlock	ClickHere
RubyMarker	RubyMarker (Asian language versions only)
RubyFrame	RubyLayout (Asian language versions only)

If a subobject is represented by its own LotusScript class, you may be able to locate that subobject without going through the Text object. However, subobjects which do not have their own LotusScript class can only be found by the GetText, Next, and Previous methods found on the Text Object.

For example, in a sentence which contains a soft hyphen and a Bookmark, you could use the Bookmark class to find the Bookmark. However, there is no class for the soft hyphen so you would have to find the soft hyphen by going through the Text object which contains that soft hyphen.

Overview: Word Pro Scalar Data Types

LotusScript recognizes the following scalar (numeric and string) data types:

Data type	Suffi	Value range	Size
Integer	%	-32,768 to 32,767Initial value: 0	2 bytes
Long	&	-2,147,483,648 to 2,147,483,647Initial value: 0	4 bytes
Single	!	-3.402823E+38 to 3.402823E+38Initial value: 0	4 bytes
Double	#	-1.7976931348623158+308 to 1.7976931348623158+308Initial value: 0	8 bytes
Currency	@	-922,337,203,685,477.5807 to 922,337,203,685,477.5807Initial value: 0	8 bytes
String	\$	(String length ranges from 0 to 32K characters)Initial value: "" (empty string)	(2 bytes/character)

Besides these scalar data types, LotusScript supports the following additional data types and data structures:

Data type or structure	Description	Size
Array	An aggregate set of elements having the same data type. An array can comprise up to 8 dimensions whose subscript bounds can range from -32768 to 32767. Initial value: Each element in a fixed array has an initial value appropriate to its data type.	Up to 64K bytes
List	A one-dimensional aggregate set whose elements have the same data type and are referred to by name rather than by subscript.	Up to 64K bytes
Variant	A special data type that can contain any scalar value, array, list, or object reference. Initial value: EMPTY	16 bytes
User-defined data type	An aggregate set of elements of possibly disparate data types. Comparable to a record in Pascal or a structure in C. Initial value: Member variables have initial values appropriate to their data type.	Up to 64K bytes
Class	An aggregate set of elements of possibly disparate data types together with procedures that operate on them. Initial value: When you create an instance of a class, LotusScript initializes its member variables to values appropriate to their data types, and generates an object reference to it.	

Object reference	A pointer to an OLE Automation object or an instance of a product class or user-defined class. Initial value: NOTHING.	4 bytes
------------------	--	---------

In each of the preceding tables, the specified storage size is platform-independent.

Note OLE is not supported under OS/2.

StrField Function

Extracts a string from within another string. Use this function to extract a string from a source string that contains two or more strings separated by a common character, such as a comma or a space.

Syntax

StrField(SourceString, FieldNum, Separator)

Elements

SourceString

A String containing two or more strings separated by the same character. The string "Seattle, WA;Atlanta, GA;Cambridge, MA;USA, All cities" contains four or more strings, depending on which separator character you use.

FieldNum

An Integer indicating the position of the string to be extracted. String positions are determined by the separator character specified in the Separator element. In the string "Seattle, WA;Atlanta, GA;Cambridge, MA;USA, All cities," a semicolon separator puts "Atlanta, GA" at position 2. Alternatively, a comma separator puts " WA;Atlanta" at position 2.

Separator

The character which separates the strings in the SourceString. In the string "Seattle, WA;Atlanta, GA;Cambridge, MA;USA, All cities," a semi-colon separator creates four strings and a comma separator creates five strings.

Return value

StrField returns a string.

Example: StrField Function

'This example script has not yet been created.'

Overview: The LotusScript Language

LotusScript is an object-oriented programming language which is shared by most Lotus applications, including Lotus Word Pro. You can use LotusScript as you would a macro language, to automate tasks, gather information, and change the appearance and functionality of Word Pro. What makes LotusScript a better tool than a macro language is the fact that you can use it with most LotusSuite applications, including the latest releases of Word Pro, Lotus 1-2-3, Notes, Approach, and Freelance Graphics.

The LotusScript language is comprised of two kinds of language elements. To access and control a Lotus application and its documents or files, you must use both of these language elements:

- Common elements which are shared by all Lotus applications:

These elements form the core of the LotusScript language. They provide the basic tools for constructing scripts, such as variables, syntax, statements, keywords, and standard data types. For more information on the core LotusScript language elements, see the LotusScript Language Reference or the [complete on-line listing](#) of core LotusScript language elements.

- Product-defined [classes](#) which are specific to one Lotus application:

These classes are defined as part of a Lotus application. For more information on Lotus Word Pro's product-defined classes, see [Overview: The Word Pro LotusScript Object Model](#).

{button ,AL(^H_THE_WORD_PRO_LOTUSSCRIPT_OBJECT_MODEL_OVER;H_USING_LOTUSSCRIPT_IN_WOR
D_PRO_OVER',0)} [See related topics](#)

'Example: Color property

'This example sets the color of the current layout to light blue.

.Layout.Background.**Color**.Red = 224

.Layout.Background.**Color**.Blue = 255

.Layout.Background.**Color**.Green = 255

.Layout.Background.**Color**.Override = \$LwpColorOverrideRgb

```
'Example: ColumnBalance property
'This example creates newspaper columns, and inserts some text.
'The columns are then balanced.
.Page.Layout.RightPage.NumCols = 2
For i = 1 To 10
    .Type "This is some text for columns. "
Next
MessageBox "Click OK to balance the columns.", MB_OK, "Example Script"
.Page.Layout.RightPage.ColumnBalance = True
```

```
'Example: ColumnGap property
'This example creates newspaper columns on the page, and sets the gap between the
columns to 1/2 inch. Some text is then inserted into the columns.
.Page.Layout.RightPage.NumCols = 2
.Page.Layout.RightPage.ColumnGap = 720 ' 720 twips = 1/2 inch.
For i = 1 To 10
    .Type "This is some text for columns. "
Next
```



```
'Example: CombineDivisions method
' This example combines two divisions names 'Body' and 'Division' into one
' division.
' RUNTIME DEPENDENCIES: You must have a document open containing these two
' divisions for this script to work.
```

```
Dim DivIdName As String
Dim DivInfoName As String
Dim DivId1 As String
Dim DivId2 As String
```

```
'Get the hexadecimal id for the 'Body' division
DivInfoName = "Body"
Gosub GetDivId
DivId1 = DivIdName
```

```
'Get the hexadecimal id for the 'Division' division
DivInfoName = "Division"
Gosub GetDivId
DivId2 = DivIdName
```

```
'combine these two divisions into one using their hexadecimal ids.
.CombineDivisions DivId1, DivId2
```

```
Exit Sub
```

```
GetDivId:
Forall Div In .ActiveDocument.Divisions
    If (Ucase$(Div.DivisionInfo.Name) = Ucase$(DivInfoName)) Then
        DivIdName = Div.Name
    End If
End Forall
Return
```

```
'Example: CombineSections method
' This example inserts two sections and then combines them into the
' current division.
' RUNTIME DEPENDENCIES: You must have a document open for this script to work.

' create the sections first
.InsertSection "Default Page", True, True, $LwpStartTypeNextpage, False, True
.InsertSection "Default Page", True, True, $LwpStartTypeNextpage, False, True
msgbox "Click OK to combine the sections.", MB_OK, "Example Script"
' now combine all the sections in the division into the division
Forall Section In .Division.Foundry.Sections
    .CombineSections Section.Name
End Forall
```

'Example: CompareFiles method

' This example compares the current file with the file named 'COMPARE.LWP'.
' RUNTIME DEPENDENCIES: You must have a document open and have a file named
' COMPARE.LWP located in the Word Pro default document directory.

FilePath = .ApplicationWindow.UserInterfacePrefs.DocPath & "\\COMPARE.LWP"

FileType = "Lotus Word Pro"

IsMultiDocs = False

IndexOfMultiDocToCompare = 0

.CompareFiles FilePath, FileType, IsMultiDocs, IndexOfMultiDocToCompare

'Example: ConditionType property

'This example changes the header so that it starts on page 2.

.Page.Layout.RightPage.Header.**ConditionType** = \$LwpConditionTypeStartatpage

.Page.Layout.RightPage.Header.PageToUseLayoutOn = 2

```
'Example: Configure method
' This example displays the SmartIcon configuration dialog
' with the "Frame" icon set selected.
' RUNTIME DEPENDENCIES: None.
```

```
.ApplicationWindow.IconBarManager.IconBars.Item("Frame").Configure
```

```
' to configure the currently active icon set, use
' .ApplicationWindow.IconBarManager.Configure
' instead.
```

'Example: ConnectCells method

' This example creates a table with 5 columns and 4 rows based on the

' Default Table style, selects the first column, and connects the cells.

' RUNTIME DEPENDENCIES: You must have a document open for this script to work.

.CreateTable False, "Default Table", 5, 4

.SelectColumn

.ConnectCells

'Example: ConnectRows method

' This example creates a table with 4 rows and 5 columns, selects the first

' row, and connects the cells in that row.

' RUNTIME DEPENDENCIES: You must have a document open for this script to work.

.CreateTable False, "Default Table", 5, 4

.SelectRow

.ConnectRows

```
'Example: ConnectSectionTabs method
' This example insert two sections into the current division and then connects
' the section tabs which creates a new division making the currently selected
' division the child of the new division.
' RUNTIME DEPENDENCIES: You must have a document open for this script to work.

.InsertSection "Default Page", True, True, $LwpStartTypeNextpage, False, True
.InsertSection "Default Page", True, True, $LwpStartTypeNextpage, False, True

.ApplicationWindow.SectionTabs.ConnectSectionTabs
```



```
'Example: Connect method
' This example creates a table with 4 rows and 5 columns, selects the entire
' table, and connects the cells.
' RUNTIME DEPENDENCIES: You must have a document open for this script to work.

.CreateTable False, "Default Table", 5, 4
.SelectTable
.Table.Connect
```

```
'Example: Container property
' This example prints the width of the current container to the LotusScript output
panel.
' Place the insertion point in different containers and run this example script.

TwipWid = .Container.Width
Print "The width is " & twipwid/1440 & " inches."
```

'Example: ContentType property

'This example uses the ContentType property to print the type of content held in the focus to the LotusScript output panel.

```
ContentsType = .Content.ContentType
```

```
Print ContentsType
```

'Example: Content property

'This example uses the ContentType property to print the type of content held in the focus to the LotusScript output panel.

```
ContentsType = .Content.ContentType
```

```
Print ContentsType
```

'Example: ContinuedFromAlignment property

'This example asks the user whether to print a message above a footnote

'which continues from the previous page, and then sets the appropriate option.

```
stat = MessageBox ("Do you want to print a continued from message?", 36, "Example  
Script")
```

```
If stat = 6 Then ' user said yes
```

```
    .Division.FootnoteOptions.IsContinuedFrom = True
```

```
    .Division.FootnoteOptions.ContinuedFromAlignment = $LtsAlignmentHorizCenter
```

```
    .Division.FootnoteOptions.ContinuedFromMessage = "Continued from previous page..."
```

```
Else
```

```
    .Division.FootnoteOptions.IsContinuedFrom = False
```

```
End If
```

'Example: ContinuedFromMessage property

'This example asks the user whether to print a message above a footnote

'which continues from the previous page, and then sets the appropriate option.

```
stat = MessageBox ("Do you want to print a continued from message?", 36, "Example  
Script")
```

```
If stat = 6 Then ' user said yes
```

```
    .Division.FootnoteOptions.IsContinuedFrom = True
```

```
    .Division.FootnoteOptions.ContinuedFromAlignment = $LtsAlignmentHorizCenter
```

```
    .Division.FootnoteOptions.ContinuedFromMessage = "Continued from previous page..."
```

```
Else
```

```
    .Division.FootnoteOptions.IsContinuedFrom = False
```

```
End If
```

'Example: ContinuedOnAlignment property

'This example asks the user whether to print a message below a footnote

'which will continue to the next page, and then sets the appropriate option.

```
stat = MessageBox ("Do you want to print a continued on message?", 36, "Example  
Script")
```

```
If stat = 6 Then ' user said yes
```

```
    .Division.FootnoteOptions.IsContinuedOn = True
```

```
    .Division.FootnoteOptions.ContinuedOnAlignment = $LtsAlignmentRight
```

```
    .Division.FootnoteOptions.ContinuedOnMessage = "Continued on next page..."
```

```
Else
```

```
    .Division.FootnoteOptions.IsContinuedOn = False
```

```
End If
```

'Example: ContinuedOnMessage property

'This example asks the user whether to print a message below a footnote

'which will continue to the next page, and then sets the appropriate option.

```
stat = MessageBox ("Do you want to print a continued on message?", 36, "Example  
Script")
```

```
If stat = 6 Then ' user said yes
```

```
    .Division.FootnoteOptions.IsContinuedOn = True
```

```
    .Division.FootnoteOptions.ContinuedOnAlignment = $LtsAlignmentRight
```

```
    .Division.FootnoteOptions.ContinuedOnMessage = "Continued on next page..."
```

```
Else
```

```
    .Division.FootnoteOptions.IsContinuedOn = False
```

```
End If
```



```
'Example: ContractOutlineLevel method
'This example inserts text in an outline structure. It then contracts and
'expands the outline.

.Text.Bullet.Name = "Default Outline"
.Text.Numbering.Position = 3
.Text.Indent.First = 360
.Text.Indent.Rest = 360
.Type "This is outline 1[Enter]"
.DemoteOutlineLevel
.Type "This is outline 2[Enter]"
.DemoteOutlineLevel
.Type "This is outline 3[Enter]"
.DemoteOutlineLevel
' move up 2 lines after typing this line
.Type "This is outline 4[Up][Up]"
MessageBox"Click OK to hide subordinate outline levels", MB_OK, "Example Script"
.ContractOutlineLevel
MessageBox"Click OK to show subordinate outline levels", MB_OK, "Example Script"
.ExpandOutlineLevel
```

```
'Example: Contract method
' This example creates two child divisions and then contracts and expands the
' divider tabs.
' RUNTIME DEPENDENCIES: You must have a document open for this script to work.

' show the tabs, and create the child divisions
.ActiveDocWindow.WinViewPrefs.IsViewSectionTabs = True
.ApplicationWindow.SectionTabs.ConnectSectionTabs
.ApplicationWindow.SectionTabs.ConnectSectionTabs
MsgBox "Click OK to contract the tabs.", MB_OK, "Example Script"

.ApplicationWindow.SectionTabs.Contract
MsgBox "Click OK to expand the tabs.", MB_OK, "Example Script"
.ApplicationWindow.SectionTabs.Expand
```

'Example: Copies property

'This example prints two copies of the current document.

```
.ActiveDocument.PrintSettings.Copies = 2
```

```
.Print
```

```
'Example: CopyDivision method
' This example copies the current division, and pastes it immediately following
' the current division. The new division is then renamed.

' get the internal name of the current division
CurrentDivision = .Division.Name
' copy it, then paste
.CopyDivision CurrentDivision
.PasteDivision CurrentDivision
'the copied division has the focus, so rename it.
.Division.DivisionInfo.Name = "Second copy"
```

```

'Example: CopyItem method
' This example adds a new menu item name 'New Edit' to the File menu just
' above the Save item. All items from the Edit menu are then copied to
' 'New Edit Menu'
' RUNTIME DEPENDENCIES: You must have not deleted the Edit or File menus
' for this script to work.

Dim MenuName as String
Dim SourceMenu As MenuItem
Dim DestinationMenu As MenuItem
Dim MenuSpacer as String
MenuSpacer = Chr$(8)
MenuName = "&New Edit"

' Set DestinationMenu to the File menu
Set DestinationMenu=.Applicationwindow.LwpMenuBar.Items.Item("&File")

' Set SourceMenu to the Edit Menu
Set SourceMenu = .ApplicationWindow.LwpMenuBar.Items.Item("&Edit")

' Create a new Edit Menu
DestinationMenu.DeleteItem "My Edit"
DestinationMenu.NewItem MenuName,,0,"&Save" & MenuSpacer & "Ctrl+S"

' Copy all the items from the Edit Menu to My new Menu
Forall Items In SourceMenu.Items
    DestinationMenu.Items(MenuName).CopyItem Items, True, ,
End Forall

```

'Example: CopyMeaning method
'This example creates a glossary entry named "Glossary". The entry is then
'copied to the temporary foundry, and pasted into the document.

```
.Type "This will be a glossary entry[shiftHome]"  
.CreateGlossaryEntry "glossary.gls", "Glossary"  
.Type "[End][Enter][Enter]"  
Set mydoc = CurrentDocument  
.Documents("glossary.gls").Activate  
Forall gloss In .Division.Foundry.Glossarys  
    gloss.CopyMeaning("Glossary")  
End Forall  
MyDoc.Activate  
.InternalPaste $LwpFoundryTypeTemporary
```

```
'Example: CopySelection method
' This example types some text, and selects it.
' it then copies the selection to the clipboard.
' RUNTIME DEPENDENCIES: You must have a document open for this script to work.
```

```
.Type "Now is the time for all good men to come to the aid of their party. "
' select the preceding paragraph
.Text.Shade $LwpLocationTypeParagraph, $LwpNavigateDirectionLeft, 1
.CopySelection
```

```
'Example: Copy method  
' This example copies the current division.  
' RUNTIME DEPENDENCIES: You must have a document open for this script to work.
```

```
Dim Div As Division  
Set Div = Bind("!Body")  
.Division.Copy Div.Name
```


'Example: CountBlankLines property

'This example sets up line numbering. Blank lines are not counted, and the
'numbers are reset on each page.

.Division.LineNumberOptions.**CountBlankLines** = False

.Division.LineNumberOptions.ResetOnEachPage = True

.Division.LineNumberOptions.NumberWhichLines = \$LwpLineNumberOptsTextLinesOnly

'Example: CreateDataFile method

' This example creates a data file for the current document. Two records are
' added and the Merge bar is opened so to insert fields for merging.

' RUNTIME DEPENDENCIES: You must have a document open for this script to work.

.CreateDataFile "~|", "Name~Address~City~State~Zip|", False, "C:\mergedat.lwp"

.MergeAddDataRecord "Jane Doe~100 Main St.~ Atlanta~ GA~30319|"

.MergeAddDataRecord "John Doe~100 Main St.~ Atlanta~ GA~30319|"

.StartFieldInsert

```
'Example: CreateDivision method
' This example creates a new division based on the "DEFAULT.MWP" SmartMaster.
' It is placed after the current division in the current document.
' RUNTIME DEPENDENCIES: You must have a document open for this script to work.
```

```
Dim SmartMaster as String
Dim NewDivId as String
```

```
SmartMaster = .ApplicationWindow.UserInterfacePrefs.StylePath & "\DEFAULT.MWP"
.CreateDivision SmartMaster, "", $LWPDivLocInsertAfterCurrentDiv, "", ""
```

'Example: CreateExternalDivision method

' This example creates an external division based on the README95.LWP file.

' It is placed after the current division.

' RUNTIME DEPENDENCIES: The README95.LWP file must be located in the WordPro
' documents path.

Dim ExternalFilename as String

ExternalFilename = .ApplicationWindow.UserInterfacePrefs.DocPath & "\README95.LWP"

.CreateExternalDivision ExternalFilename, "", \$LwpDivLocInsertAfterCurrentdiv, "", ""

'Example: CreateFrame method

' This example inserts a 1 inch by 1 inch frame into the current document.

' The "Default Frame" style is used which places the upper left corner of the

' frame 1 inch down and 1 inch to the left of the page's upper left corner.

' RUNTIME DEPENDENCIES: You must have a document open for this script to work.

.CreateFrame False, "Default Frame",1440, 1440

```
'Example: CreateGlossaryEntry method
' This example inserts a glossary entry named NewGlossaryItem for the current
' selection in the specified Glossary file.
' RUNTIME DEPENDENCIES: You must have a document open for this script to work.

.Type "Some stuff for the glossary."
.SelectSentence
.CreateGlossaryEntry "GLOSSARY.GLS", "NewGlossaryItem"
```

'Example: CreateGlossary method

' This example stores a file name in the variable GlossFileName, hides the
' open documents, opens the default Word Pro glossary file, creates and saves
' a glossary file named "GLOSTST.GLS" in the User Setup glossary directory,
' then closes the glossary files and resets the default values user interface
' preferences.

' RUNTIME DEPENDENCIES: You must have create file rights in the specified
' glossary directory for this script to work.

Dim GlossFileName As String

GlossFileName = "GLOSTST.GLS"

.ApplicationWindow.UserInterfacePrefs.OpenDocsVisible = False

.GlossaryOpen GlossFileName, "Lotus Word Pro"

.CreateGlossary

.SaveGlossary GlossFileName, "Lotus Word Pro", False

.Close

.ApplicationWindow.UserInterfacePrefs.IsReplacement = False

.ApplicationWindow.UserInterfacePrefs.OpenDocsVisible = True

.ApplicationWindow.UserInterfacePrefs.OpenReadOnly = False

'Example: CreateGraphic method

' This example creates a Word Pro Drawing frame.

' RUNTIME DEPENDENCIES: You must have a document open for this script to work.

.CreateGraphic "WordProDraw", False


```

'Example: CreateNewButton method
' This example creates a new button on the status bar and then adds text to
' the button. The STATUSBARBUTTONOVERRIDE TEXT event is bound to the
' SetTheButtonText subroutine to set the button text during needs repainting.
' RUNTIME DEPENDENCIES: You must have a document open for this script to work.

Dim ButtonName As String
Dim NewButton As StatusBarButton
With .ApplicationWindow.StatusBar
    ButtonName = .CreateNewbutton (0, 0, 100, LwpButtonTypeText )
'create the new button
    With .StatusBarButtons(ButtonName)
'
        .SetOverrideText("New Button...")
        .InvalidateButton
        On Event STATUSBARBUTTONOVERRIDE TEXT From .StatusBarButtons(ButtonName) Call
SetTheButtonText
    End With
        .InvalidateWholeBar ' Force the bar to repaint
End With
End Sub

Sub SetTheButtonText (Source As StatusBarButton, ButtonName As String)
    'Add the the button text each time the status bar needs repainting.
    Source.SetOverrideText("New Button...")
End Sub

```

```
'Example: CreateOleEmbeddedFile method
' This example creates an embedded Lotus 1-2-3 OLE object from the file named
' 'embed.123'.
' RUNTIME DEPENDENCIES: You must have a document open and a file named
' 'embed.123' located in C:\Lotus\Work\123 for this script to
' work.
```

```
Dim FilePath As String
```

```
Dim ClassID As String
```

```
Dim IconMetaFilePictHandle As Integer
```

```
FilePath = "C:\Lotus\Work\123\Embed.123"
```

```
ClassID = "{00000000-0000-0000-0000-000000000000}"
```

```
IconMetaFilePictHandle = 0
```

```
.CreateOleEmbeddedFile ClassID, FilePath, IconMetaFilePictHandle
```

Note OLE is not supported in OS/2.

'Example: CreateOleFromFile method

'This example creates an OLE object from a bitmap file located in the
'Word Pro directory. The object is embedded in the current document.

.CreateOleFromFile "E:\lotus\wordpro\helpbutn.bmp", False

Note OLE is not supported under OS/2.

```
'Example: CreateOleLinkedFile method
' This example creates a linked Lotus 1-2-3 OLE object from the file named
' 'Link.123'.
' RUNTIME DEPENDENCIES: You must have a document open and a file named
' 'Link.123' located in the C:\Lotus\Work\123 directory for this script to
' work.
```

```
Dim FilePath As String
Dim IconMetaFilePictHandle As Integer
```

```
FilePath = "C:\Lotus\Work\123\Link.123"
```

```
IconMetaFilePictHandle = 0
.CreateOleLinkedFile FilePath, IconMetaFilePictHandle
```

Note OLE is not supported under OS/2.

```
'Example: CreateOleNew method
' This example creates a new Lotus Approach OLE object in the current
' document.
' RUNTIME DEPENDENCIES: You must have a document open and have Lotus Approach
' installed for this script to work.
```

```
Dim ClassID As String
```

```
ClassID = "{00028703-0000-0000-c000-000000000046}"
```

```
.CreateOleNew(ClassID)
```

Note OLE is not supported under OS/2.

'Example: CreateParallelColumns method

' This example creates a parallel column table with 3 columns.

' RUNTIME DEPENDENCIES: You must have a document open for this script to work.

.CreateParallelColumns 3, \$LtsAlignmentHorizCenter

'Example: CreateRemark method

' This example inserts a version remark in the current version of the current
' document.

' RUNTIME DEPENDENCIES: You must have a document open for this script to work.

.Division.VersionManager.CurrentVersion.CreateRemark("Test Remark",835302017,"LOT")

'Example: CreateTable method

' This example creates a table with 4 rows and 5 columns.

' RUNTIME DEPENDENCIES: You must have a document open for this script to work.

.CreateTable False, "Default Table", 5, 4


```
'Example: CreateTextEntryField method
' This example creates two text entry fields in the current document.
' RUNTIME DEPENDENCIES: You must have a document open for this script to work.

' create a field 2 inches long, with a solid 1/2 point black line
.CreateTextEntryField 2880, $LtsBorderPatternSolid, 10, 0

' create a field 3 inches long, with a wavy double 2 point blue line
.CreateTextEntryField 3760, $LwpBorderPatternDblWavy, 40, 16711680
```

```
'Example: CreateVersion method
' This example creates a version for the current document then deletes the
' version.
' RUNTIME DEPENDENCIES: You must have a document open for this script to work.
```

```
.Division.VersionManager.CreateVersion "NewVersion"
```

```
Forall Version In .ActiveDocument.VersionManager.Versions
  If Version.name = "NewVersion" Then
    .ActiveDocument.VersionManager.DeleteVersion Version.DocVersionId
  End If
End Forall
```

```
'Example: Create method  
' This example creates a new character style.  
' RUNTIME DEPENDENCIES: You must have a document open for this script to work.
```

```
Dim StyleName as String  
StyleName = "My New Char Style"  
Style = .Division.Foundry.Create($LwpFoundryCreateTypeStyle, StyleName, 39)  
With .Division.Foundry.CharacterStyles(Style)  
    .Font.Underline = True  
    .Font.FontColor.Blue = 128  
    .Font.FontColor.Red = 128  
    .Font.FontColor.Green = 0  
End With
```

```
'Example: CreationDateString property
With .ActiveDocument
MsgTxt = "Current Word Pro Doc is " & .FullName
MsgTxt = MsgTxt & ", the author is " & .Docinfo.AuthorName
Msgbox MsgTxt,64,"Word Pro Information"
MsgTxt = "It was created on " & .DocInfo.CreationDateString & " at "
& .DocInfo.CreationTimeString
Msgbox MsgTxt,64,"Word Pro Information"
End With
```

```
'Example: CreationTimeString property
With .ActiveDocument
MsgTxt = "Current Word Pro Doc is " & .FullName
MsgTxt = MsgTxt & ", the author is " & .Docinfo.AuthorName
Msgbox MsgTxt,64,"Word Pro Information"
MsgTxt = "It was created on " & .DocInfo.CreationDateString & " at "
& .DocInfo.CreationTimeString
Msgbox MsgTxt,64,"Word Pro Information"
End With
```

'Example: CreationTimeValue property
'This example determines the time between when the current document was
'created, and when it was last modified, and displays the results in a message box.
'In order to return a reasonable result, the document must have been saved
'at least once.

```
Dim Days As Long, Hours As Integer, Minutes As Integer
CTime = .ActiveDocument.DocInfo.CreationTimeValue
MTime = .ActiveDocument.DocInfo.ModifiedTimeValue
TotalSec = MTime - CTime ' The total number of seconds since document created
Days = Int(TotalSec / 86400) ' There 86,400 seconds in a day
Hours = Int((TotalSec Mod 86400) / 3600)
Minutes = Int((TotalSec Mod 3600) / 60)
Seconds = Int(TotalSec Mod 60)
MessageBox "This document is " & Days & " days " & Hours & " hours " & Minutes & "
minutes " & Seconds & " seconds old." , MB_OK, "Example Script"
```

'Example: Crop property

'This example asks the user whether to print with crop marks, sets the
'appropriate option, and prints the document.

```
stat = MessageBox ("Do you want to print with crop marks?", 36, "Example Script")
If stat = 6 Then ' user said yes
    .ActiveDocument.PrintSettings.Crop = True
Else
    .ActiveDocument.PrintSettings.Crop = False
End If
.Print
```

```
'Example: CurrentCell property
'This example creates a table, then protects the current cell.

.CreateTable False, "Default Table", 2, 2
.Type "This cell will be protected."
.Table.CurrentCell.IsProtected = True ' enable cell protection
```



```
'Example: CurrentEditor property
'This example sets the editing rights of all editors except the current editor to "Not
Allowed".
Forall DocEditor In .ActiveDocument.EditorManager.Editors
  ' drive through all the editors for this document.
  If DocEditor.Name <> .ActiveDocument.EditorManager.CurrentEditor.Name Then
    ' if the editor isn't the current one, set rights to not allowed.
    DocEditor.Abilities =$LwpEditAbilEditingNotAllowed
  End If
End Forall
```

'Example: CurrentLanguage property
'This example inserts text in two languages, then uses the CurrentLanguage
'property to report the language of the text.

```
.Text.InsertText "This is some text. ", True  
.Text.Language.Language = $LwpLanguagesFrenchCanadian  
.Text.InsertText "En Français. ", True  
.Text.Language.Language = $LwpLanguagesSystem  
.Text.MoveUp 2  
MessageBox "The current language is " & .Text.CurrentLanguage  
.Text.MoveDown 1  
MessageBox "The current language is " & .Text.CurrentLanguage
```

```
'Example: CurrentlyPrintingInBackground property
'This example determines if the current document is printing in
'the background.

If .ActiveDocument.PrintManager.CurrentlyPrintingInBackground = True Then
    MessageBox "Word Pro is printing in the background.", MB_OK, "Example Script"
Else
    MessageBox "Word Pro is not printing in the background.", MB_OK, "Example Script"
End If
```

```
'Example: CurrentRunningScriptName property
' This example displays a message box with the name and path of this script.

SName = .ApplicationWindow.CurrentRunningScriptName
SPath = .ApplicationWindow.CurrentRunningScriptPath
If SPath = "" Then ' no pathname; we're untitled
    MessageBox "The file holding the current script is " & SName & ", which has never
    been saved.", MB_OK, "Example Script"
Else
    MessageBox "The file holding the current script is " & SName & ", which is located
    in " & SPath & ".", MB_OK, "Example Script"
End If
```

```
'Example: CurrentRunningScriptPath property
' This example displays a message box with the name and path of this script.
' If the CurrentRunningScriptPath property is empty, the document has never
' been saved, and we tell the user this

SName = .ApplicationWindow.CurrentRunningScriptName
SPath = .ApplicationWindow.CurrentRunningScriptPath
If SPath = "" Then ' no pathname; we're untitled
    Messagebox "The file holding the current script is " & SName & ", which has never
    been saved.", MB_OK, "Example Script"
Else
    Messagebox "The file holding the current script is " & SName & ", which is located
    in " & SPath & ".", MB_OK, "Example Script"
End If
```

'Example: CurrentVersion property
'This example displays a message box with information about the current
'version.

```
Dim Ver As Version
Set Ver = .ActiveDocument.VersionManager.CurrentVersion
CDate = Ver.CreationDateString
CTime = Ver.CreationTimeString
VName = Ver.VersionName
MessageBox "The current version, " & VName & ", was created on " & CDate & " at " &
CTime, MB_OK, "Example Script"
```

'Example: CustomLength property

'This example sets the length of footnote separator lines to 4 inches.

```
.Division.FootnoteOptions.FootnoteSeparator.UseSeparatorLine = True  
.Division.FootnoteOptions.FootnoteSeparator.IsFixedLength = False  
.Division.FootnoteOptions.FootnoteSeparator.CustomLength = 5760 ' 4 inches  
.Division.FootnoteOptions.FootnoteSeparator.IndentFromLeft = 0  
.Division.FootnoteOptions.FootnoteSeparator.SpaceAbove = 0  
.Division.FootnoteOptions.FootnoteSeparator.SpaceBelow = 80
```

'Example: CustomViewLevel property

'This example sets the custom view level to 91% and displays the custom view.

%INCLUDE "WPBITMSK.LSS" ' needed to set value for LwpViewsCustom.

.ActiveDocWindow.WinViewPrefs.**CustomViewLevel** = 91

.ActiveDocWindow.WinViewPrefs.ViewType = LwpViewsCustom


```
'Example: CutDivision method
' This example inserts a new division in the document. The original division
' is then cut from the document, and pasted following the newly added division.
' Finally, the pasted division is given its original divider tab name.

' Get the internal name of the division we want to cut
CurrentDivision = .Division.Name
' Get the name on the divider tab
CurrentName = .Division.DivisionInfo.Name
' Put in a new division, and store its internal name
.ApplicationWindow.SectionTabs.AddSectionTabs
NewDiv = .Division.Name
' Cut the division, then paste it after the new division
.CutDivision CurrentDivision
.PasteDivision NewDiv
' Restore the pasted division's original divider tab name
.Division.DivisionInfo.Name = CurrentName
```

```
'Example: CutSelection method
' This example inserts some text into the current document.  The text is then
' selected and cut to the clipboard.
' RUNTIME DEPENDENCIES: You must have a document open for this script to work.

.Text.InsertText "This is some sample text."
.Text.Select $LwpSelectObjectTypeParagraph
.CutSelection
```

```
'Example: DarkMode method
' This example turns on DarkMode which prevents the screen from updating.
' Some text is inserted, a table is created and then DarkMode is turned off.
' RUNTIME DEPENDENCIES: You must have a document open for this script to work.

.ApplicationWindow.Darkmode True
For t =1 To 5
    .Type ("Hello world [Enter]")
Next
.CreateTable False, "Default Table", 4, 2
.ApplicationWindow.DarkMode False
```

'Example: DateCreatedValue property
'This example determines the time between when the current version was
'created, and when it was last modified, and displays the results in a message box.
'In order to return a reasonable result, the document must have been saved
'at least once.

```
Dim Days As Long, Hours As Integer, Minutes As Integer
CTime = .ActiveDocument.VersionManager.CurrentVersion.DateCreatedValue
MTime = .ActiveDocument.VersionManager.CurrentVersion.DateRevisedValue
TotalSec = MTime - CTime ' The total number of seconds since version created
Days = Int(TotalSec / 86400) ' There 86,400 seconds in a day
Hours = Int((TotalSec Mod 86400) / 3600)
Minutes = Int((TotalSec Mod 3600) / 60)
Seconds = Int(TotalSec Mod 60)
MessageBox "This document is " & Days & " days " & Hours & " hours " & Minutes & "
minutes " & Seconds & " seconds old." , MB_OK, "Example Script"
```

'Example: DateRevisedValue property
'This example determines the time between when the current version was
'created, and when it was last modified, and displays the results in a message box.
'In order to return a reasonable result, the document must have been saved
'at least once.

```
Dim Days As Long, Hours As Integer, Minutes As Integer
CTime = .ActiveDocument.VersionManager.CurrentVersion.DateCreatedValue
MTime = .ActiveDocument.VersionManager.CurrentVersion.DateRevisedValue
TotalSec = MTime - CTime ' The total number of seconds since version created
Days = Int(TotalSec / 86400) ' There 86,400 seconds in a day
Hours = Int((TotalSec Mod 86400) / 3600)
Minutes = Int((TotalSec Mod 3600) / 60)
Seconds = Int(TotalSec Mod 60)
MessageBox "This document is " & Days & " days " & Hours & " hours " & Minutes & "
minutes " & Seconds & " seconds old." , MB_OK, "Example Script"
```

```
'Example: DbUnderline method
' This example toggles the double underline attribute of the selected text.
' RUNTIME DEPENDENCIES: You must have a document open and some text selected
' for this script to work.
```

```
.DbUnderline
MessageBox "Click OK undo double underline change.",MB_OK,"Example Script"
.DbUnderline
```

'Example: OSType property

' This example determines the operating system Word Pro was designed for,
' and displays it, along with version information, in a message box.

```
mytype = .OsType
```

```
If MyType = $LwpPlatformWin16 Then OS = "Windows 3.x"
```

```
If MyType = $LwpPlatformWin32 Then OS = "32 bit Windows"
```

```
If MyType = $LwpPlatformOS2 Then OS = "OS/2"
```

```
RelNum = .ReleaseNumber
```

```
Version = .ProductVersion
```

```
MessageBox "This is Word Pro for " & OS & " Release " & Version & "." & RelNum, MB_OK,  
"Example Script"
```

```
'Example: PasteDivision method
' This example copies the current division, and pastes it immediately following
' the current division. The new division is then renamed.

' get the internal name of the current division
CurrentDivision = .Division.Name
' copy it, then paste
.CopyDivision CurrentDivision
.PasteDivision CurrentDivision
'the copied division has the focus, so rename it.
.Division.DivisionInfo.Name = "Second copy"
```


'Example: OSType property

' This example determines the operating system Word Pro was designed for,
' and displays it, along with version information, in a message box.

```
mytype = .OsType
```

```
If MyType = $LwpPlatformWin16 Then OS = "Windows 3.x"
```

```
If MyType = $LwpPlatformWin32 Then OS = "32 bit Windows"
```

```
If MyType = $LwpPlatformOS2 Then OS = "OS/2"
```

```
RelNum = .ReleaseNumber
```

```
Version = .ProductVersion
```

```
MessageBox "This is Word Pro for " & OS & " Release " & Version & "." & RelNum, MB_OK,  
"Example Script"
```

Word Pro: ContinuedFromMessage property

{button ,AL('H_FOOTNOTEOPTIONS_CLASS',0)} [See list of classes](#)

{button ,AL('H_CONTINUEDFROMMESSAGE_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) The "continued from" message that is generated for continued footnotes.

Data Type

[String](#)

Syntax

continuedfrommessagevalue = [objectreference].ContinuedFromMessage

[objectreference].ContinuedFromMessage = continuedfrommessagevalue

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Equivalent to the "continued from" Initial text of message box on the Continued Messages panel of the Footnote and Endnote Options dialog box. You can reach this dialog box by choosing Create - Footnote/Endnote and then clicking Options.

Word Pro: ContinuedFromStory property

{button ,AL(`H_FOUNDRY_CLASS',0)} [See list of classes](#)

(Read-only) The name of the Text object that contains the "Continued from" message for a Division object.

Data Type

String

Syntax

continuedfromstoryvalue = [objectreference].ContinuedFromStory

Legal values

Usage

When a footnote is too large to fit in the footnote space provided, Word Pro automatically flows the footnote to the footnote space on the next page. Word Pro also adds a "Continued on" message to the first page and a "Continued from" message on the next page. These messages do not vary within a division, but can vary from one division to the next. Each "Continued from" message is a Text object that you can manipulate in a script. This ContinuedFromStory property stores the name of the Continued From Text object for the specified Division object.

Word Pro does not use this property in the WPAApplication.AppFoundry, WPAApplication.TempFoundry, or TextDocument.Foundry properties.

{button ,AL(`H_CONTINUEDFROMMESSAGE_PROPERTY_MEMDEF',0)} [See related topics](#)

Word Pro: ContinuedOnAlignment property

{button ,AL(^H_FOOTNOTEOPTIONS_CLASS',0)} [See list of classes](#)

{button ,AL(^H_CONTINUEDONALIGNMENT_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Returns or sets the footnote "continued on" message alignment.

Data Type

Variant (Enumerated)

AlignmentType

Syntax

continuedonalignmentvalue = [objectreference].ContinuedOnAlignment

[objectreference].ContinuedOnAlignment = continuedonalignmentvalue

Legal values

<u>Value</u>	<u>Effect</u>
\$LtsAlignmentHorizCenter (1056964611)	The footnote is centered horizontally.
\$LtsAlignmentJustify (1056964613)	Not used.
\$LtsAlignmentLeft (1056964609)	The footnote is left-aligned.
\$LtsAlignmentRight (1056964610)	The footnote is right-aligned.
\$LtsAlignmentSmart (1056964612)	Not used.
\$LwpAlignmentTypeAlignRevert (8)	Not used.
\$LwpAlignmentTypeJustifyall (5)	Not used.
\$LwpAlignmentTypeNumericleft (6)	Not used.
\$LwpAlignmentTypeNumericright (7)	Not used.

Usage

Equivalent to the "continued on" Alignment setting on the Continued Messages panel of the Footnote and Endnote Options dialog box.

Word Pro: ContinuedOnMessage property

{button ,AL('H_FOOTNOTEOPTIONS_CLASS',0)} [See list of classes](#)

{button ,AL('H_CONTINUEDONMESSAGE_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) The "continued on" message that is generated for continued footnotes.

Data Type

[String](#)

Syntax

continuedonmessagevalue = [objectreference].ContinuedOnMessage

[objectreference].ContinuedOnMessage = continuedonmessagevalue

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Equivalent to the continued on "Initial text of message" box on the Continued Messages panel of the Footnote and Endnote Options dialog box. You can reach this dialog box by choosing Create - Footnote/Endnote and then clicking Options.

{button ,AL('H_CONTINUEDONSTORY_PROPERTY_MEMDEF',0)} [See related topics](#)

Word Pro: ContinuedOnStory property

{button ,AL(`H_FOUNDRY_CLASS',0)} [See list of classes](#)

(Read-only) The name of the Text object that contains the "Continued on" message for a Division object.

Data Type

String

Syntax

continuedonstoryvalue = [objectreference].ContinuedOnStory

Legal values

Usage

When a footnote is too large to fit in the footnote space provided, Word Pro automatically flows the footnote to the footnote space on the next page. Word Pro also adds a "Continued on" message to the first page and a "Continued from" message on the next page. These messages do not vary within a division, but can vary from one division to the next. Each "Continued on" message is a Text object that you can manipulate in a script. This ContinuedOnStory property stores the name of the Continued On Text object for the specified Division object.

Word Pro does not use this property in the WPAApplication.AppFoundry, WPAApplication.TempFoundry, or TextDocument.Foundry properties.

{button ,AL(`H_CONTINUEDONMESSAGE_PROPERTY_MEMDEF',0)} [See related topics](#)

Word Pro: ConvertOnNew property

{button ,AL(^H_POWERFIELD_CLASS',0)} [See list of classes](#)

(Read-only)

Data Type

[Integer](#)

Syntax

convertonnewvalue = [objectreference].ConvertOnNew

Legal values**Usage**

Word Pro: Copies property

{button ,AL('H_PRINTSETTINGS_CLASS',0)} [See list of classes](#)

{button ,AL('H_COPIES_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Indicates the number of copies of a document that will print.

Data Type

[Integer](#)

Syntax

copiesvalue = [objectreference].Copies

[objectreference].Copies = copiesvalue

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Equivalent to choosing File - Print and selecting a number in the "Number of copies" box.

Word Pro: CountBlankLines property

{button ,AL('H_LINENUMBEROPTIONS_CLASS',0)} [See list of classes](#)

{button ,AL('H_COUNTBLANKLINES_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write)

Data Type

[Integer](#)

Syntax

countblanklinesvalue = [objectreference].CountBlankLines

[objectreference].CountBlankLines = countblanklinesvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Word Pro: Count property

```
{button ,AL('H_BASECOLLECTION_CLASS;H_BAGCOLLECTION_CLASS;H_BOOKMARKCOLLECTION_CLASS;  
H_CELLCOLLECTION_CLASS;H_CELLLAYOUTCOLLECTION_CLASS;H_CHARACTERSTYLECOLLECTION_  
CLASS;H_CLICKHERECOLLECTION_CLASS;H_CONNECTEDLAYOUTCOLLECTION_CLASS;H_CONTENTC  
OLLECTION_CLASS;H_DDELINKCOLLECTION_CLASS;H_DIVISIONCOLLECTION_CLASS;H_DOCUMENTS_  
CLASS;H_EDITORCOLLECTION_CLASS;H_ENDNOTE LAYOUTCOLLECTION_CLASS;H_DOCINFOFIELDCOL  
LECTION_CLASS;H_POWERFIELDCOLLECTION_CLASS;H_FOOTERLAYOUTCOLLECTION_CLASS;H_FOO  
TNOTECOLLECTION_CLASS;H_FOOTNOTE LAYOUTCOLLECTION_CLASS;H_FRAME LAYOUTCOLLECTION_  
_CLASS;H_GLOSSARYCOLLECTION_CLASS;H_GRAPHICCOLLECTION_CLASS;H_GRAPHICOLEBJECTC  
OLLECTION_CLASS;H_GROUPLAYOUTCOLLECTION_CLASS;H_HEADERLAYOUTCOLLECTION_CLASS;H_I  
CONBARCOLLECTION_CLASS;H_LAYOUTCOLLECTION_CLASS;H_MARKERCOLLECTION_CLASS;H_MEN  
UITEMCOLLECTION_CLASS;H_NOTELAYOUTCOLLECTION_CLASS;H_OLEOBJECTCOLLECTION_CLASS;H_  
_OUTLINESEQCOLLECTION_CLASS;H_OUTLINESEQITEMCOLLECTION_CLASS;H_PAGELAYOUTCOLLECT  
ION_CLASS;H_PARAGRAPHSTYLECOLLECTION_CLASS;H_PARALLELCOLSCOLLECTION_CLASS;H_ROW  
LAYOUTCOLLECTION_CLASS;H_RUBY LAYOUTCOLLECTION_CLASS;H_SECTIONCOLLECTION_CLASS;H_  
SILVERBULLETCOLLECTION_CLASS;H_SMARTCORRECTCOLLECTION_CLASS;H_SMARTFILLCOLLECTIO  
N_CLASS;H_STATUSBARBUTTONCOLLECTION_CLASS;H_STRINGCOLLECTION_CLASS;H_SUPERTABLEC  
OLLECTION_CLASS;H_SUPERTABLELAYOUTCOLLECTION_CLASS;H_TABLECOLLECTION_CLASS;H_TABL  
EHEADINGCOLLECTION_CLASS;H_TABLEHEADINGLAYOUTCOLLECTION_CLASS;H_TABLE LAYOUTCOLLE  
CTION_CLASS;H_TABLEMARKERCOLLECTION_CLASS;H_TABLEONLYCOLLECTION_CLASS;H_TEXTCOLL  
ECTION_CLASS;H_TEXTMARKERCOLLECTION_CLASS;H_TEXTSTYLECOLLECTION_CLASS;H_VERSIONC  
OLLECTION_CLASS;H_DOCWINDOWCOLLECTION_CLASS;H_WPDATASETCOLLECTION_CLASS;H_DROP  
CAPLAYOUTCOLLECTION_CLASS;H_CHARTDATAPOINTS_CLASS;H_CHARTLEGENDENTRIES_CLASS;H_  
CHARTMAJORGRIDLINES_CLASS;H_CHARTPIES_CLASS;H_CHARTPIESLICEGROUPS_CLASS;H_CHARTP  
IESLICES_CLASS;H_CHARTPIETITLES_CLASS;H_CHARTSERIESCOLLECTION_CLASS;H_CHARTTABLESE  
RIESCOLLECTION_CLASS;H_CHARTTEXTENTRIES_CLASS;H_CHARTTEXTLABELS_CLASS;')0}} See list of  
classes
```

(Read-only) Returns the number of items in the specified collection.

Data Type

Long

Syntax

countvalue = [objectreference].Count

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

The Count property indicates the number of items in the specified collection object. For example, the Documents collection object stores all documents that are currently open. In order to determine how many documents are being stored within the Documents collection, you can use the following statement:

```
numdocsoopen = .Documents.count
```

Word Pro: CreatePreviewBitmap property

{button ,AL(^H_DIVISION_CLASS;H_TEXTDOCUMENT_CLASS',0)} [See list of classes](#)

(Read-write)

Data Type

Data type is Integer.

Syntax

[objectreference].CreatePreviewBitmap = createpreviewbitmapvalue

createpreviewbitmapvalue = [objectreference].CreatePreviewBitmap

Legal values

Usage

Word Pro: CreationDateString property

{button ,AL('H_DOCINFO_CLASS;H_VERSION_CLASS',0)} [See list of classes](#)

{button ,AL('H_CREATIONDATESTRING_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-only) Returns the date the document was created as a String value.

Data Type

String

Syntax

creationdatestringvalue = [objectreference].CreationDateString

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

The format of the CreationDateString property is determined by the Windows Short date style.

Word Pro: CreationTimeString property

{button ,AL('H_DOCINFO_CLASS;H_VERSION_CLASS',0)} [See list of classes](#)

{button ,AL('H_CREATIONTIMESTRING_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-only) Returns the time the document was created as a String value.

Data Type

[String](#)

Syntax

creationtimestringvalue = [objectreference].CreationTimeString

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

The format of the CreationTimeString property is determined by the Windows Time style.

Word Pro: CreationTimeValue property

{button ,AL('H_DOCINFO_CLASS',0)} [See list of classes](#)

{button ,AL('H_CREATIONTIMEVALUE_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read only) Returns the time the document was created as a Long data type.

Data Type

Long

Syntax

creationtimevaluevalue = [objectreference].CreationTimeValue

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

The value that this property returns represents the number of seconds that have elapsed since midnight on January 1, 1970.

Word Pro: Crop property

{button ,AL('H_PRINTSETTINGS_CLASS',0)} [See list of classes](#)

{button ,AL('H_CROP_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Prints fine lines that indicate the corners of the page.

Data Type

Integer

Syntax

cropvalue = [objectreference].Crop

[objectreference].Crop = cropvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Enables you to crop any page you specify in a document. Crop marks print offset .5 inches from the top left corner of the physical page. Equivalent to choosing File - Print, clicking Options, and selecting "With crop marks."

Word Pro: CurrentCell property

{button ,AL(^H_WPAPPLICATION_CLASS;H_BASetable_CLASS;H_FOOTNOTETABLE_CLASS;H_PARALLELCO
LUMNS_CLASS;H_GLOSSARY_CLASS;H_TABLEHEADING_CLASS;H_TABLE_CLASS';0)} [See list of classes](#)

{button ,AL(^H_CURRENTCELL_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-only) The CellLayout object for the table cell which is uppermost in the focus. If no cell is in the focus, this property is empty.

Data Type

[CellLayout](#)

Syntax

currentcellvalue = [objectreference].CurrentCell

Legal values

Always contains an instance of the CellLayout class.

Usage

In most cases, you can use this property to access the cell in which the insertion point is located. If the insertion point is in a frame that is in a table cell, this property contains the CellLayout object for the cell containing the frame.

Word Pro: CurrentColumn property

{button ,AL(^H_WPAPPLICATION_CLASS;H_BASSETABLE_CLASS;H_FOOTNOTETABLE_CLASS;H_PARALLELCOLUMNS_CLASS;H_GLOSSARY_CLASS;H_TABLEHEADING_CLASS;H_TABLE_CLASS';0)} [See list of classes](#)

(Read-only) Returns the layout object for the column or columns uppermost in the focus.

Data Type

[Layout](#)

Syntax

currentcolumnvalue = [objectreference].CurrentColumn

Legal values

This property can contain an instance of the Layout class or any of its derived classes.

Usage

If there is no table in the focus, this property is empty. If only one cell is selected, this property contains the CellLayout object for that cell. If more than one cell in the table is selected, this property contains a ColumnGroupLayout object representing the selected cells.

Word Pro: CurrentEditor property

{button ,AL('H_EDITORMANAGER_CLASS',0)} [See list of classes](#)

{button ,AL('H_CURRENTEDITOR_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-only) The editor object for the current editor of a document.

Data Type

[Editor](#)

Syntax

currenteditorvalue = [objectreference].CurrentEditor

Legal values

Always contains an instance of the Editor class.

Usage

Gives you access to the editor object for the current editor. Therefore, you can use this property to find out the name, initials, abilities, and so on, of the current editor of a document.

Word Pro: CurrentLanguage property

{button ,AL('H_CLICKHERE_CLASS;H_TEXTMARKER_CLASS;H_TEXT_CLASS',0)} [See list of classes](#)

{button ,AL('H_CURRENTLANGUAGE_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-only) Indicates the text's language at the insertion point. If text is selected and more than one language is assigned to the selected text, Word Pro returns the language for the text nearest the insertion point.

Data Type

Variant (Enumerated)

Syntax

currentlanguagevariable = [objectreference].CurrentLanguage

Legal values

\$LwpLanguagesAfrikaans (474)

\$LwpLanguagesAlbanian (475)

\$LwpLanguagesAmerican (442)

\$LwpLanguagesArabic (1025)

\$LwpLanguagesArabicAlgeria (5121)

\$LwpLanguagesArabicBahrain (15361)

\$LwpLanguagesArabicEgypt (3073)

\$LwpLanguagesArabicIraq (2049)

\$LwpLanguagesArabicJordan (11265)

\$LwpLanguagesArabicKuwait (13313)

\$LwpLanguagesArabicLebanon (12289)

\$LwpLanguagesArabicLibya (4097)

\$LwpLanguagesArabicMorocco (6145)

\$LwpLanguagesArabicOman (8193)

\$LwpLanguagesArabicQatar (16385)

\$LwpLanguagesArabicSyria (10241)

\$LwpLanguagesArabicTunisia (7169)

\$LwpLanguagesArabicUAE (14337)

\$LwpLanguagesArabicYemen (9217)

\$LwpLanguagesAustralian (444)

\$LwpLanguagesBasque (1069)

\$LwpLanguagesBrazilian (468)

\$LwpLanguagesBritish (443)

\$LwpLanguagesBritishize (12297)

\$LwpLanguagesBritishmedize (13321)

\$LwpLanguagesBrmedical (11273)

\$LwpLanguagesBulgarian (478)

\$LwpLanguagesByelorussian (1059)

\$LwpLanguagesCatalan (436)

\$LwpLanguagesChineseHongKong (3076)

\$LwpLanguagesChinesePRChina (481)

\$LwpLanguagesChineseSingapore (4100)

\$LwpLanguagesChineseTraditional (479)

\$LwpLanguagesCroatian (1050)

\$LwpLanguagesCroatianCyrillic (2074)

\$LwpLanguagesCroatianLatin (1050)
\$LwpLanguagesCroatianSerbian (3098)
\$LwpLanguagesCzech (437)
\$LwpLanguagesDanish (438)
\$LwpLanguagesDutch (439)
\$LwpLanguagesDutchBelgian (440)
\$LwpLanguagesEnglishCanadian (445)
\$LwpLanguagesEnglishCaribbean (9225)
\$LwpLanguagesEnglishIreland (447)
\$LwpLanguagesEnglishJamaica (8201)
\$LwpLanguagesEnglishNewzealand (446)
\$LwpLanguagesEnglishSAfrica (7177)
\$LwpLanguagesEstonian (1061)
\$LwpLanguagesFaeroese (1080)
\$LwpLanguagesFarsi (1081)
\$LwpLanguagesFinnish (452)
\$LwpLanguagesFrench (453)
\$LwpLanguagesFrenchBelgian (454)
\$LwpLanguagesFrenchCanadian (455)
\$LwpLanguagesFrenchLuxembourg (5132)
\$LwpLanguagesFrenchSwiss (456)
\$LwpLanguagesGerman (457)
\$LwpLanguagesGermanAustrian (459)
\$LwpLanguagesGermanLiechtenstein (5127)
\$LwpLanguagesGermanLuxembourg (4103)
\$LwpLanguagesGermanSwiss (458)
\$LwpLanguagesGreek (460)
\$LwpLanguagesHebrew (483)
\$LwpLanguagesHungarian (461)
\$LwpLanguagesIcelandic (484)
\$LwpLanguagesIndonesian (1057)
\$LwpLanguagesItalian (462)
\$LwpLanguagesItalianSwiss (463)
\$LwpLanguagesJapanese (485)
\$LwpLanguagesKorean (486)
\$LwpLanguagesKoreanJohab (2066)
\$LwpLanguagesLatvian (1062)
\$LwpLanguagesLithuanian (1063)
\$LwpLanguagesMedical (448)
\$LwpLanguagesNorwegian (464)
\$LwpLanguagesNynorsk (465)
\$LwpLanguagesPolish (466)
\$LwpLanguagesPortuguese (467)
\$LwpLanguagesRhaetoRoman (487)
\$LwpLanguagesRomanian (488)
\$LwpLanguagesRussian (469)
\$LwpLanguagesRussianio (470)

\$LwpLanguagesSlovak (492)
\$LwpLanguagesSlovene (493)
\$LwpLanguagesSorbian (1070)
\$LwpLanguagesSpanish (471)
\$LwpLanguagesSpanishArgentina (11274)
\$LwpLanguagesSpanishBolivia (16394)
\$LwpLanguagesSpanishChile (13222)
\$LwpLanguagesSpanishColombia (9226)
\$LwpLanguagesSpanishCostaRica (5130)
\$LwpLanguagesSpanishDominican (7178)
\$LwpLanguagesSpanishEcuador (12298)
\$LwpLanguagesSpanishGuatemala (4106)
\$LwpLanguagesSpanishMexican (2058)
\$LwpLanguagesSpanishPanama (6154)
\$LwpLanguagesSpanishParaguay (15370)
\$LwpLanguagesSpanishPeru (10250)
\$LwpLanguagesSpanishUruguay (14346)
\$LwpLanguagesSpanishVenezuela (8202)
\$LwpLanguagesSwedish (473)
\$LwpLanguagesSystem (434)
\$LwpLanguagesThai (494)
\$LwpLanguagesTurkish (495)
\$LwpLanguagesUkrainian (496)
\$LwpLanguagesUniversal (435)
\$LwpLanguagesUrdu (497)
\$LwpLanguagesVoorkeur (441)

Usage

{button ,AL(`H_LANGUAGE_PROPERTY_MEMDEF',0)} [See related topics](#)

Word Pro: CurrentlyPrintingInBackground property

{button ,AL('H_PRINTMANAGER_CLASS',0)} [See list of classes](#)

{button ,AL('H_CURRENTLYPRINTINGINBACKGROUND_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-only)

This language element has not yet been defined. You can check for a newer Help file on the Lotus web site.

Data Type

Integer

Syntax

currentlyprintinginbackgroundvalue = [objectreference].CurrentlyPrintingInBackground

Legal values

This property is read-only. The value of this property cannot be set by a script.

Usage

Word Pro: CurrentRow property

{button ,AL(^H_WPAPPLICATION_CLASS;H_BASSETABLE_CLASS;H_FOOTNOTETABLE_CLASS;H_PARALLELCOLUMNS_CLASS;H_GLOSSARY_CLASS;H_TABLEHEADING_CLASS;H_TABLE_CLASS';0)} [See list of classes](#)

(Read-only) Returns the layout object for the row(s) uppermost in the focus.

Data Type

[RowLayout](#)

Syntax

currentrowvalue = [objectreference].CurrentRow

Legal values

Can contain an instance of the Layout class or any of its derived classes.

Usage

If there is no table in the focus, this property is empty. If only one cell is selected, this property contains the CellLayout object for that cell. If more than one cell in the same row is selected, this property contains a CellGroupLayout object representing the cells selected in that row. If more than one cell is selected across multiple rows, this property contains a CellGroupLayout object representing the cells selected in that row.

Word Pro: CurrentRunningScriptName property

{button ,AL('H_APPLICATIONWINDOW_CLASS',0)} [See list of classes](#)

{button ,AL('H_CURRENTRUNNINGSCRIPTNAME_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-only)

Data Type

Data type is String.

Syntax

currentrunningscriptnamevalue = [objectreference].CurrentRunningScriptName

Legal values

The name of the currently running script.

Usage

Word Pro: CurrentRunningScriptPath property

{button ,AL('H_APPLICATIONWINDOW_CLASS',0)} [See list of classes](#)

{button ,AL('H_CURRENTRUNNINGSCRIPTPATH_PROPERTY_EXSCRIPT',1)} [See example](#)

{Read-only}

Data Type

Data type is String.

Syntax

currentrunningscriptpathvalue = [objectreference].CurrentRunningScriptPath

Legal values

The full path for the document containing the script currently running.

Usage

Word Pro: CurrentVersion property

{button ,AL('H_VERSIONMANAGER_CLASS',0)} [See list of classes](#)

{button ,AL('H_CURRENTVERSION_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-only)

Data Type

[Version](#)

Syntax

currentversionvalue = [objectreference].CurrentVersion

Legal values

Always contains an instance of the Version class.

Usage

Word Pro: CustomLength property

{button ,AL(^H_FOOTNOTESEPOPT_CLASS;H_FOOTNOTECONTSEP_CLASS;H_FOOTNOTESEPARATOR_CLASSES',0)} [See list of classes](#)

{button ,AL(^H_CUSTOMLENGTH_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Determines the length of a custom footnote separator line.

Data Type

Long

Syntax

customlengthvalue = [objectreference].CustomLength

[objectreference].CustomLength = customlengthvalue

Legal values

Data type is Long but the unit of measurement used for this property is Twips. There are 1440 Twips per inch.

Usage

Equivalent to the "Custom length" box on the Separators panel of the Footnote and Endnote Options dialog box.

Word Pro: CustomViewLevel property

{button ,AL('H_WINVIEWPREFS_CLASS',0)} [See list of classes](#)

{button ,AL('H_CUSTOMVIEWLEVEL_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Sets or returns the current custom view level percentage value.

Data Type

[Integer](#)

Syntax

customviewlevelvalue = [objectreference].CustomViewLevel

[objectreference].CustomViewLevel = customviewlevelvalue

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Equivalent to the "Custom level" box on the Zoom tab of the View Preferences dialog box. To set the custom view to display you can use the ViewType property

{button ,AL('H_VIEWTYPE_PROPERTY_MEMDEF',0)} [See related topics](#)

Word Pro: DataFileFieldNames property

{button ,AL(^H_MERGEOPTIONS_CLASS',0)} [See list of classes](#)

(Read-only) The names of merge fields in the Merge data file.

Data Type

String

Syntax

datafilefieldnamesvalue = [objectreference].DataFileFieldNames

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Equivalent to the field names on a new or existing Merge data file.

Word Pro: DataFileName property

{button ,AL('H_MERGEOPTIONS_CLASS',0)} [See list of classes](#)

(Read-write) The path and name of the Merge data file.

Data Type

String

Syntax

datafilenamevalue = [objectreference].DataFileName

[objectreference].DataFileName = datafilenamevalue

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Equivalent to the path and file name of a new or existing Merge data file.

Word Pro: DataFormat property

{button ,AL(^H_GRAPHICOLEBJECT_CLASS;H_GRAPHIC_CLASS;H_OLEOBJECT_CLASS',0)} [See list of classes](#)

(Read-write)

Data Type

Data type is String.

Syntax

[objectreference].DataFormat = dataformatvalue

dataformatvalue = [objectreference].DataFormat

Legal values

Usage

Word Pro: DataNames property

{button ,AL(^H_SCRIPTDATASET_CLASS;H_WPDATASET_CLASS',0)} [See list of classes](#)

(Read-only) All of the variable names in a data set.

Data Type

[StringCollection](#)

Syntax

datanamesvalue = [objectreference].DataNames

Legal values

Always contains an instance of the StringCollection class.

Usage

Use this property to return all the existing variable names in a specific data set.

Word Pro: DateCreatedValue property

{button ,AL('H_VERSION_CLASS',0)} [See list of classes](#)

{button ,AL('H_DATECREATEDVALUE_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-only)

Data Type

[Long](#)

Syntax

datecreatedvaluevalue = [objectreference].DateCreatedValue

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Word Pro: DateRevisedValue property

{button ,AL('H_VERSION_CLASS',0)} [See list of classes](#)

{button ,AL('H_DATEREVISIEDVALUE_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-only)

Data Type

[Long](#)

Syntax

daterevisedvaluevalue = [objectreference].DateRevisedValue

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Word Pro: DdeEnabled property

{button ,AL('H_PREFERENCES_CLASS',0)} [See list of classes](#)

{button ,AL('H_DDEENABLED_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write)

Data Type

Data type is Integer.

Syntax

[objectreference].DDEEnabled = ddeenabledvalue

ddeenabledvalue = [objectreference].DDEEnabled

Legal values

Usage

Word Pro: DdeLinkManager property

{button ,AL(^H_DIVISION_CLASS;H_TEXTDOCUMENT_CLASS',0)} [See list of classes](#)

(Read-only)

Data Type

[DdeLinkManager](#)

Syntax

ddelinkmanagervalue = [objectreference].DdeLinkManager

Legal values

Always contains an instance of the DdeLinkManager class.

Usage

Word Pro: DdeLinks property

{button ,AL(^H_DDELINKMANAGER_CLASS',0)} [See list of classes](#)

(Read-only) Contains all DdeLinks in a division and lists them by their readable names.

Data Type

[DdeLinkCollection](#)

Syntax

ddelinksvalue = [objectreference].DdeLinks

Legal values

As String (name). Always contains an instance of the DdeLinkCollection class.

Usage

Provides the readable names of all DdeLinks in a division.

In order to access all DDE links in a document, you must iterate the collection of divisions within the document to access each division's DdeLinkManager object.

Word Pro: DdeOutboundInfo property

{button ,AL('H_DIVISION_CLASS;H_TEXTDOCUMENT_CLASS',0)} [See list of classes](#)

(Read-write)

Data Type

Long

Syntax

ddeoutboundinfovalue = [objectreference].DdeOutboundInfo

[objectreference].DdeOutboundInfo = ddeoutboundinfovalue

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Word Pro: DebugVariable property

{button ,AL('H_MACRO_CLASS',0)} [See list of classes](#)

(Read-write)

Data Type

Integer

Syntax

debugvariablevalue = [objectreference].DebugVariable

[objectreference].DebugVariable = debugvariablevalue

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Word Pro: DefaultBinName property

{button ,AL(^H_PRINTMANAGER_CLASS',0)} [See list of classes](#)

(Read-only)

Data Type

String

Syntax

defaultbinnamevalue = [objectreference].DefaultBinName

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Word Pro: DefaultCellStyleDescription property

{button ,AL('H_PREFERENCES_CLASS',0)} [See list of classes](#)

(Read-write)

Data Type

String

Syntax

defaultcellstyledescriptionvalue = [objectreference].DefaultCellStyleDescription

[objectreference].DefaultCellStyleDescription = defaultcellstyledescriptionvalue

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Word Pro: DefaultColumnName property

{button ,AL('H_PREFERENCES_CLASS',0)} [See list of classes](#)

(Read-write)

Data Type

String

Syntax

defaultcolumnstylevalue = [objectreference].DefaultColumnName

[objectreference].DefaultColumnName = defaultcolumnstylevalue

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Word Pro: DefaultDropCapStyleDescription property

{button ,AL('H_PREFERENCES_CLASS',0)} [See list of classes](#)

(Read-write)

Data Type

Data type is String.

Syntax

[objectreference].DefaultDropCapStyleDescription = defaultdropcapdescriptionvalue

defaultdropcapdescriptionvalue = [objectreference].DefaultDropCapStyleDescription

Legal values

Usage

Word Pro: DefaultFilePath property

{button ,AL(^H_APPLICATION_CLASS;H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

(Read-only) Returns the path for the current working directory for Word Pro. Equivalent to the Location property on WPAApplication.

Data Type

String

Syntax

defaultfilepathvalue = [objectreference].DefaultFilePath

Legal values

The value of this property cannot be set by a script.

Usage

If you use LotusScript to create a file without specifying a directory for that file, Word Pro stores that file in the current working directory.

Note The working directory is usually the same as the directory in which the Word Pro executable is installed. However, you can change the working directory by specifying the path to the directory in the Properties dialog box for the Word Pro executable (WORDPRO.EXE). In Windows 3.1, select the WORDPRO.EXE file within the Windows Program Manager, choose File - Properties, and specify the path in the "Working Directory" box. In Windows 95, right-click the WORDPRO.EXE file, choose Properties, then specify the path in the "Start In" box.

Word Pro: DefaultFileType property

{button ,AL('H_TEXTDOCUMENT_CLASS',0)} [See list of classes](#)

{button ,AL('H_DEFAULTFILETYPE_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) The default file type to which Word Pro saves a SmartMaster.

Data Type

String

Syntax

defaultfiletypevalue = [objectreference].DefaultFileType

[objectreference].DefaultFileType = defaultfiletypevalue

Legal values

String

A String value that indicates the file type in which the document will be saved. A null string saves the document as a Word Pro file. Some of the usual file types are listed in the table below, but each user's list of available file types is derived from the list of text filters installed during the Word Pro installation.

DCA/RFT	Lotus Manuscript 2.x	MS Word for Windows 1.0
DIF	Lotus Organizer 1.x	MS Word for Windows 2.0
DisplayWrite	Lotus Word Pro	MS Word for Windows 6.0
HTML	Lotus Word Pro SmartMaster	MS Word for Windows95 7.0
Lotus 1-2-3	MS Excel	MS WordPad 1.0
Lotus 1-2-3 for OS/2	MS Excel 3.0	OfficeWriter 4,5,6
Lotus 1-2-3 R3	MS Excel 4.0	Rich Text Format(RTF)
Lotus 1-2-3 R4,5	MS Excel 5.0	SAMNA Word
Lotus 1-2-3 R6	MS Excel 7.0	WordPerfect 5.0
Lotus Ami Pro	MS Windows Write 3.x	WordPerfect 5.1
Lotus Ami Pro 3.x Macro	MS Word for DOS 3,4,5,6	WordPerfect 6.x
Lotus Ami Pro 3.x Styles	MS Word for OS/2	WordStar 2000 R3

Usage

Equivalent to choosing a file type in the Initial Save Format drop-down box in the Save As SmartMaster Options dialog box. To get to this dialog box, choose Save As - Word Pro SmartMaster (.MWP file) and then OK.

When a document created with this SmartMaster is initially saved, this file type will be selected in the "Save as type" box in the Save As dialog box.

Word Pro: DefaultFrameStyleDescription property

{button ,AL('H_PREFERENCES_CLASS',0)} [See list of classes](#)

(Read-write)

Data Type

String

Syntax

defaultframestyledescriptionvalue = [objectreference].DefaultFrameStyleDescription

[objectreference].DefaultFrameStyleDescription = defaultframestyledescriptionvalue

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Word Pro: DefaultLatinFont property

{button ,AL('H_PREFERENCES_CLASS',0)} [See list of classes](#)

(Read-write)

Data Type

String

Syntax

defaultlatinfontvalue = [objectreference].DefaultLatinFont

[objectreference].DefaultLatinFont = defaultlatinfontvalue

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Word Pro: DefaultLeftColumnStyleName property

{button ,AL('H_PARALLELCOLUMNS_CLASS;H_GLOSSARY_CLASS;H_PREFERENCES_CLASS',0)} [See list of classes](#)

(Read-write) The name of the default style for the left column of a parallel column.

Data Type

String

Syntax

defaultleftcolumnstylevalue = [objectreference].DefaultLeftColumnStyleName

[objectreference].DefaultLeftColumnStyleName = defaultleftcolumnstylevalue

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Word Pro: DefaultMenu property

{button ,AL('H_PREFERENCES_CLASS',0)} [See list of classes](#)

{button ,AL('H_DEFAULTMENU_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write)

The name of the menu set to use as the default.

Data Type

String

Syntax

defaultmenuvalue = [objectreference].DefaultMenu

[objectreference].DefaultMenu = defaultmenuvalue

Legal values

Any value of type String.

Usage

This property holds the name of a menu set such as those that appear in the "Available menu sets" box in the Menu Customization dialog box. It does not hold the filename for a menu set (CUSTOM.MNU).

Equivalent to choosing File - User Setup - Menu Customization, selecting a menu set, and clicking Make Default.

Word Pro: DefaultNewCategory property

{button ,AL('H_USERINTERFACEPREFS_CLASS',0)} [See list of classes](#)

(Read-write) Sets or returns the default SmartMaster category to be used for new documents.

Data Type

String

Syntax

defaultnewcategoryvalue = [objectreference].DefaultNewCategory

[objectreference].DefaultNewCategory = defaultnewcategoryvalue

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Determines which SmartMaster category is selected in the Create from any SmartMaster panel of the New Document dialog box. This property must be set to a valid SmartMaster category in order to have an effect.

Word Pro: DefaultNonLatinFont property

{button ,AL('H_PREFERENCES_CLASS',0)} [See list of classes](#)

(Read-write)

Data Type

String

Syntax

defaultnonlatinfontvalue = [objectreference].DefaultNonLatinFont

[objectreference].DefaultNonLatinFont = defaultnonlatinfontvalue

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Word Pro: DefaultPageHeight property

{button ,AL('H_PRINTMANAGER_CLASS',0)} [See list of classes](#)

(Read-only)

Data Type

Long

Syntax

defaultpageheightvalue = [objectreference].DefaultPageHeight

Legal values

Data type is Long but the unit of measurement used for this property is Twips. There are 1440 Twips per inch.

Usage

Word Pro: DefaultPageStyleDescription property

{button ,AL('H_PREFERENCES_CLASS',0)} [See list of classes](#)

(Read-write)

Data Type

String

Syntax

defaultpagestyledescriptionvalue = [objectreference].DefaultPageStyleDescription

[objectreference].DefaultPageStyleDescription = defaultpagestyledescriptionvalue

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Word Pro: DefaultPageWidth property

{button ,AL(^H_PRINTMANAGER_CLASS',0)} [See list of classes](#)

(Read-only)

Data Type

Long

Syntax

defaultpagewidthvalue = [objectreference].DefaultPageWidth

Legal values

Data type is Long but the unit of measurement used for this property is Twips. There are 1440 Twips per inch.

Usage

Word Pro: DefaultPitch property

{button ,AL(^H_FONT_CLASS',0)} [See list of classes](#)

(Read-only)

Data Type

Integer

Syntax

defaultpitchvalue = [objectreference].DefaultPitch

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Word Pro: DefaultRightColumnName property

{button ,AL('H_PARALLELCOLUMNS_CLASS;H_GLOSSARY_CLASS;H_PREFERENCES_CLASS',0)} [See list of classes](#)

(Read-write) The name of the default style for the right column of a parallel column.

Data Type

String

Syntax

defaultrightcolumnstylevalue = [objectreference].DefaultRightColumnName

[objectreference].DefaultRightColumnName = defaultrightcolumnstylevalue

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Word Pro: DefaultTableStyleDescription property

{button ,AL('H_PREFERENCES_CLASS',0)} [See list of classes](#)

(Read-write)

Data Type

String

Syntax

defaulttablestyledescriptionvalue = [objectreference].DefaultTableStyleDescription

[objectreference].DefaultTableStyleDescription = defaulttablestyledescriptionvalue

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Word Pro: DefaultTextStyleDescription property

{button ,AL('H_PREFERENCES_CLASS',0)} [See list of classes](#)

(Read-write)

Data Type

String

Syntax

defaulttextstyledescriptionvalue = [objectreference].DefaultTextStyleDescription

[objectreference].DefaultTextStyleDescription = defaulttextstyledescriptionvalue

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Word Pro: DefCellStyleName property

{button ,AL(^H_BASETABLE_CLASS;H_FOOTNOTETABLE_CLASS;H_PARALLELCOLUMNS_CLASS;H_GLOSSARY_CLASS;H_TABLEHEADING_CLASS;H_TABLE_CLASS',0)} [See list of classes](#)

(Read-write) The default style layout assigned to a cell object.

Data Type

String

Syntax

defcellstylevalue = [objectreference].DefCellStyleName

[objectreference].DefCellStyleName = defcellstylevalue

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Specifies the layout style of cell layout objects which are created after you set this property. For example, if you create a table, the default cell style is assigned to all cells within the table. If you modify the DefCellStyleName property of the table, any new cell layout objects that are created use the cell layout style that you specify.

Word Pro: DefColWidth property

{button ,AL(^H_BASSETABLE_CLASS;H_FOOTNOTETABLE_CLASS;H_PARALLELCOLUMNS_CLASS;H_GLOSSARY_CLASS;H_TABLEHEADING_CLASS;H_TABLE_CLASS',0)} [See list of classes](#)

{button ,AL(^H_DEFCOLWIDTH_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) The default width of column objects inserted within a table object.

Data Type

Long

Syntax

defcolwidthvalue = [objectreference].DefColWidth

[objectreference].DefColWidth = defcolwidthvalue

Legal values

Data type is Long but the unit of measurement used for this property is Twips. There are 1440 Twips per inch.

Usage

Determines the default width, or resets the default width, of a column object.

Word Pro: Definition property

{button ,AL(^H_CHARACTERSTYLE_CLASS;H_LAYOUT_CLASS;H_CELLGROUPLAYOUT_CLASS;H_CELLLAYO
UT_CLASS;H_CONNECTEDLAYOUT_CLASS;H_COLUMNGROUPLAYOUT_CLASS;H_FOOTERLAYOUT_CLA
SS;H_FRAMELAYOUT_CLASS;H_FRAMEGROUPLAYOUT_CLASS;H_DROPCAPLAYOOUT_CLASS;H_GROUPL
AYOUT_CLASS;H_HEADERLAYOUT_CLASS;H_NOTELAYOUT_CLASS;H_PAGELAYOUT_CLASS;H_ROWGR
OUPLAYOUT_CLASS;H_ROWLAYOUT_CLASS;H_RUBYLAYOUT_CLASS;H_SUPERTABLEGROUPLAYOUT_
CLASS;H_SUPERTABLELAYOUT_CLASS;H_TABLELAYOUT_CLASS;H_ENDNOTELAYOUT_CLASS;H_FOOT
NOTELAYOUT_CLASS;H_TABLEHEADINGLAYOUT_CLASS;H_TOCSUPERTABLELAYOUT_CLASS;H_PARAG
RAPHSTYLE_CLASS',0)} [See list of classes](#)

{button ,AL(^H_DEFINITION_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-only)

CharacterStyle includes attributes, facename, size info, and so on (look in hierarchy box when creating a character style).

ParagraphStyle includes attributes and things that comprise the style (look in hierarchy box when creating a paragraph style).

[Layout]

Indicates what style properties are received from another style, based on the style hierarchy.

Data Type

Data type is Long, which allows the value of this parameter to be one of the constants listed below or its hexadecimal equivalent (in parentheses). You can combine these values when you want Word Pro to combine the features listed below. Use the OR operator to combine values.

Syntax

definitionvalue = [objectreference].Definition

Legal values

[Layout]

<u>Value</u>	<u>Effect</u>
LwpLayStyOverSize (&H1)	A layout object's size settings are based on its layout style.
LwpLayStyOverPlacement (&H2)	A layout object's placement settings are based on its layout style.
LwpLayStyOverMargins (&H4)	A layout object's margin settings are based on its layout style.
LwpLayStyOverBorders (&H8)	A layout object's border settings are based on its layout style.
LwpLayStyOverBackground (&H10)	A layout object's background settings are based on its layout style.
LwpLayStyOverJoins (&H20)	A layout object's join settings are based on its layout style.
LwpLayStyOverShadow (&H40)	A layout object's shadow settings are based on its layout style.
LwpLayStyOverChildren (&H10000)	A layout object's children are based on its layout style.
LwpLayStyOverColumns (&H400)	A layout object's column settings are based on its layout style.
LwpLayStyOverContents (&H20000)	A layout object's content settings are based on its layout style.
LwpLayStyOverLeaders (&H2000)	A layout object's leader settings are based on its layout style.
LwpLayStyOverMisc (&H8000)	A layout object's misc settings are based on its layout style.

LwpLayStyOverNumerics (&H200)	A layout object's numeric settings are based on its layout style.
LwpLayStyOverOrientation (&H4000)	A layout object's orientation settings are based on its layout style.
LwpLayStyOverRotation (&H1000)	A layout object's rotation settings are based on its layout style.
LwpLayStyOverScaling (&H800)	A layout object's scaling settings are based on its layout style.
LwpLayStyOverScript (&H100)	A layout object's script settings are based on its layout style.
LwpLayStyOverSizeAndPlacement (&H3)	A layout object's size and placement settings are based on its layout style.
LwpLayStyOverTabs (&H80)	A layout object's tab settings are based on its layout style.

Usage

[Layout]

Layout objects, such as frames, table cells, and pages are based on styles. For example, when you create a frame in Word Pro, it is usually based on a style called Default Frame. When you create a table in Word Pro, the table is usually based on a style called Default Table, and the table cells are usually based on a style called Default Cell.

When you create a style in Word Pro, it is always based on an existing style. This is known as style hierarchy. Style hierarchy is represented in Word Pro on the Style panel of the InfoBox for layout objects. If you click Create Style from the Style panel, then choose Hierarchy, you see the list of available style properties. These style properties correspond with the property values listed above, under Legal Values.

The Definition property of a layout object represents all style properties which come from a style that is higher up in the style hierarchy. In the Style Hierarchy Definition dialog box, they are represented as unchecked style properties.

Word Pro: DefRowHeight property

{button ,AL(^H_BASETABLE_CLASS;H_FOOTNOTETABLE_CLASS;H_PARALLELCOLUMNS_CLASS;H_GLOSSARY_CLASS;H_TABLEHEADING_CLASS;H_TABLE_CLASS',0)} [See list of classes](#)

(Read-write) The default height of row objects inserted within a table object.

Data Type

Long

Syntax

defrowheightvalue = [objectreference].DefRowHeight

[objectreference].DefRowHeight = defrowheightvalue

Legal values

Data type is Long but the unit of measurement used for this property is Twips. There are 1440 Twips per inch.

Usage

Determines the default height, or resets the default height, of a row object.

Word Pro: DelayedGreeting property

{button ,AL('H_DOCCONTROL_CLASS',0)} [See list of classes](#)

Indicates whether or not a document's greeting message displays when the document is opened.

Data Type

Integer

Syntax

delayedgreeting = [objectreference].DelayedGreeting

[objectreference].DelayedGreeting = delayedgreeting

Legal values

The legal values for this property are -1 and 0 but you may use the LotusScript constants of True (-1) and False (0) instead of the integer values. The default value for this property is False.

Usage

A True setting causes a document to open without displaying its greeting message. Word Pro sets this property internally when a document is opened hidden. For example, when a Word Pro document is launched via OLE automation, this property is set to True so that the document's greeting does not display. Later, if the document is made visible, Word Pro sets this property to False, and the greeting is allowed to display.

Note OLE is not supported under OS/2.

Word Pro: DeleteFont property

{button ,AL(^H_EDITOR_CLASS;H_REVISIONDISPLAY_CLASS',0)} [See list of classes](#)

{button ,AL(^H_DELETEFONT_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-only) The collection of attributes associated with a font object that marks deleted text in a document.

Data Type

[Font](#)

Syntax

deletefontvalue = [objectreference].DeleteFont

Legal values

Always contains an instance of the Font class.

Usage

[Editor]

Equivalent to the "Markup for deletions" options in the Markup Options dialog box. You can reach this dialog box by clicking Markup Options on the General panel of the Word Pro Preferences dialog box.

Word Pro: DemandLoad property

{button ,AL('H_DIVISION_CLASS;H_TEXTDOCUMENT_CLASS',0)} [See list of classes](#)

(WriteOnly)

Data Type

Variant (Enumerated)

ReservedParam

Syntax

[objectreference].DemandLoad = demandloadvalue

Legal values

\$LwpReservedParamDefault (1707)

Usage

Word Pro: Descent property

{button ,AL(^H_FONT_CLASS,0)} [See list of classes](#)

(Read-only) Represents the descent characteristic of a font in points and fractions of points.

Data Type

Single

Syntax

descentvalue = [objectreference].Descent

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

The descent of a font is the distance from the baseline to the bottom of the characters.

There are 72 points per inch.

Word Pro: DescriptionFileName property

{button ,AL(^H_MERGEOPTIONS_CLASS',0)} [See list of classes](#)

(Read-write) If you choose DescriptionFile when opening an external data file from another application, Word Pro stores the field names and separator of that file in a separate description file. The name of that file is stored in this property.

Data Type

String

Syntax

descriptionfilenamevalue = [objectreference].DescriptionFileName

[objectreference].DescriptionFileName = descriptionfilenamevalue

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Word Pro uses the DescriptionFile to get the separator and field names for an external data file.

If you are not using a description file for your merge, you do not need to set a value for this property.

Word Pro: Description property

{button ,AL('H_BASEOBJECT_CLASS',0)} [See list of classes](#)

{button ,AL('H_DESCRIPTION_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) The description of an object. This property is defined in the BaseObject class and inherited by all Word Pro objects. It is not, however, used by all Word Pro objects. Objects that make use of this property usually use it to store a description of the object.

Data Type

String

Syntax

descriptionvalue = [objectreference].Description

[objectreference].Description = descriptionvalue

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

You will most likely use this property in a TextDocument object. When you call this property from a TextDocument object, it returns the value of the Description field for the document represented by that object. When you call this property from the WPAApplication object, it returns the value of the Description field for the currently active document. Most other objects do not use this description property.

Word Pro: DiagonalLines property

{button ,AL(^H_TABLELINE_CLASS',0)} [See list of classes](#)

{button ,AL(^H_DIAGONALLINES_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-only) Allows you to access the diagonal line information for a specific TableLine object.

Data Type

[BorderLines](#)

Syntax

diagonallinesvalue = [objectreference].DiagonalLines

Legal values

Always contains an instance of the BorderLines class.

Usage

Word Pro: DirectionDown property

```
{button ,AL('H_LAYOUT_CLASS;H_CELLGROUPLAYOUT_CLASS;H_CELLLAYOUT_CLASS;H_CONNECTEDLAYOUT_CLASS;H_COLUMNGROUPLAYOUT_CLASS;H_FOOTERLAYOUT_CLASS;H_FRAMELAYOUT_CLASS;H_FRAMEGROUPLAYOUT_CLASS;H_DROP_CAP_LAYOUT_CLASS;H_GROUPLAYOUT_CLASS;H_HEADERLAYOUT_CLASS;H_NOTELAYOUT_CLASS;H_PAGELAYOUT_CLASS;H_ROWGROUPLAYOUT_CLASS;H_ROWLAYOUT_CLASS;H_RUBY_LAYOUT_CLASS;H_SUPERTABLEGROUPLAYOUT_CLASS;H_SUPERTABLELAYOUT_CLASS;H_TABLELAYOUT_CLASS;H_ENDNOTELAYOUT_CLASS;H_FOOTNOTELAYOUT_CLASS;H_TABLEHEADINGLAYOUT_CLASS;H_TOCSUPERTABLELAYOUT_CLASS',0)} See list of classes
```

```
{button ,AL('H_DIRECTIONDOWN_PROPERTY_EXSCRIPT',1)} See example
```

(Read-write) Specifies whether a layout will autogrow or autosize in a downward direction.

Data Type

Data type is [Variant](#), which allows the value of this property to be one of the constants listed below or its hexadecimal equivalent (in parentheses). You can combine these values when you want Word Pro to combine the features listed below. Use the OR operator to combine values.

Syntax

directiondownvalue = [objectreference].DirectionDown

[objectreference].DirectionDown = directiondownvalue

Legal values

<u>Value</u>	<u>Effect</u>
LwpLayoutDirectionUseDirection (&H1)	Specifies that a layout object will adhere to other bit values set for this property.
LwpLayoutDirectionAutoGrow (&H2)	Specifies that a layout object autogrows in a downward direction.
LwpLayoutDirectionAutoSize (&H4)	Specifies that a layout object autosizes in a downward direction.
LwpLayoutDirectionTocontainer (&H8)	Specifies that a layout object autogrows or autosizes to the bottom edge of its container.
LwpLayoutDirectionAllDirections (&HF)	Simultaneously autogrows or autosizes a layout object in all directions (down, up, left, and right).

Usage

Make sure to set the &H1 bit value if you want to enable any of the grow or size options for a layout object. For example, if you want to specify that a layout object should automatically grow, it is not sufficient to set the property value to &H2. You must also set the &H1 bit in order for the other bit settings to be valid. An example statement is shown below:

```
[objectreference].DirectionDown = &H1 OR &H2
```

Word Pro: DirectionLeft property

```
{button ,AL(^H_LAYOUT_CLASS;H_CELLGROUPLAYOUT_CLASS;H_CELLLAYOUT_CLASS;H_CONNECTEDLAY  
OUT_CLASS;H_COLUMNGROUPLAYOUT_CLASS;H_FOOTERLAYOUT_CLASS;H_FRAMELAYOUT_CLASS;H  
_FRAMEGROUPLAYOUT_CLASS;H_DROP_CAPLAYOUT_CLASS;H_GROUPLAYOUT_CLASS;H_HEADERLAY  
OUT_CLASS;H_NOTELAYOUT_CLASS;H_PAGELAYOUT_CLASS;H_ROWGROUPLAYOUT_CLASS;H_ROWLA  
YOUT_CLASS;H_RUBYLAYOUT_CLASS;H_SUPERTABLEGROUPLAYOUT_CLASS;H_SUPERTABLELAYOUT  
_CLASS;H_TABLELAYOUT_CLASS;H_ENDNOTELAYOUT_CLASS;H_FOOTNOTELAYOUT_CLASS;H_TABLE  
HEADINGLAYOUT_CLASS;H_TOCSUPERTABLELAYOUT_CLASS',0)} See list of classes
```

(Read-write) Indicates whether a layout will autogrow or autosize toward the left.

Data Type

Data type is [Variant](#), which allows the value of this property to be one of the constants listed below or its hexadecimal equivalent (in parentheses). You can combine these values when you want Word Pro to combine the features listed below. Use the OR operator to combine values.

Syntax

```
directionleftvalue = [objectreference].DirectionLeft
```

```
[objectreference].DirectionLeft = directionleftvalue
```

Legal values

<u>Value</u>	<u>Effect</u>
LwpLayoutDirectionUseDirection (&H1)	Specifies that a layout object will adhere to other bit values set for this property.
LwpLayoutDirectionAutoGrow (&H2)	Specifies that a layout object autogrows toward the left.
LwpLayoutDirectionAutoSize (&H4)	Specifies that a layout object autosizes toward the left.
LwpLayoutDirectionTocontainer (&H8)	Specifies that a layout object autogrows or autosizes to the left edge of its container.
LwpLayoutDirectionAllDirections (&HF)	Simultaneously autogrows or autosizes a layout object in all directions (down, up, left, and right).

Usage

Make sure to set the &H1 bit value if you want to enable any of the grow or size options for a layout object. For example, if you want to specify that a layout object should automatically grow, it is not sufficient to set the property value to &H2. You must also set the &H1 bit in order for the other bit settings to be valid. An example statement is shown below:

```
[objectreference].DirectionLeft = &H1 OR &H2
```


Word Pro: DirectionRight property

{button ,AL(^H_LAYOUT_CLASS;H_CELLGROUPLAYOUT_CLASS;H_CELLLAYOUT_CLASS;H_CONNECTEDLAYOUT_CLASS;H_COLUMNGROUPLAYOUT_CLASS;H_FOOTERLAYOUT_CLASS;H_FRAMELAYOUT_CLASS;H_FRAMEGROUPLAYOUT_CLASS;H_DROP_CAP_LAYOUT_CLASS;H_GROUPLAYOUT_CLASS;H_HEADERLAYOUT_CLASS;H_NOTELAYOUT_CLASS;H_PAGELAYOUT_CLASS;H_ROWGROUPLAYOUT_CLASS;H_ROWLAYOUT_CLASS;H_RUBY_LAYOUT_CLASS;H_SUPERTABLEGROUPLAYOUT_CLASS;H_SUPERTABLELAYOUT_CLASS;H_TABLELAYOUT_CLASS;H_ENDNOTELAYOUT_CLASS;H_FOOTNOTELAYOUT_CLASS;H_TABLEHEADINGLAYOUT_CLASS;H_TOCSUPERTABLELAYOUT_CLASS',0)} [See list of classes](#)

(Read-write) Indicates whether a layout will autogrow or autosize toward the right.

Data Type

Data type is [Variant](#), which allows the value of this property to be one of the constants listed below or its hexadecimal equivalent (in parentheses). You can combine these values when you want Word Pro to combine the features listed below. Use the OR operator to combine values.

Syntax

directionrightvalue = [objectreference].DirectionRight

[objectreference].DirectionRight = directionrightvalue

Legal values

<u>Value</u>	<u>Effect</u>
LwpLayoutDirectionUseDirection (&H1)	Specifies that a layout object will adhere to other bit values set for this property.
LwpLayoutDirectionAutoGrow (&H2)	Specifies that a layout object autogrows toward the right.
LwpLayoutDirectionAutoSize (&H4)	Specifies that a layout object autosizes toward the right.
LwpLayoutDirectionTocontainer (&H8)	Specifies that a layout object autogrows or autosizes to the right edge of its container.
LwpLayoutDirectionAllDirections (&HF)	Simultaneously autogrows or autosizes a layout object in all directions (down, up, left, and right).

Usage

Make sure to set the &H1 bit value if you want to enable any of the grow or size options for a layout object. For example, if you want to specify that a layout object should automatically grow, it is not sufficient to set the property value to &H2. You must also set the &H1 bit in order for the other bit settings to be valid. An example statement is shown below:

```
[objectreference].DirectionRight = &H1 OR &H2
```

Word Pro: DirectionUp property

```
{button ,AL(^H_LAYOUT_CLASS;H_CELLGROUPLAYOUT_CLASS;H_CELLLAYOUT_CLASS;H_CONNECTEDLAYOUT_CLASS;H_COLUMNGROUPLAYOUT_CLASS;H_FOOTERLAYOUT_CLASS;H_FRAMELAYOUT_CLASS;H_FRAMEGROUPLAYOUT_CLASS;H_DROP_CAPLAYOUT_CLASS;H_GROUPLAYOUT_CLASS;H_HEADERLAYOUT_CLASS;H_NOTELAYOUT_CLASS;H_PAGELAYOUT_CLASS;H_ROWGROUPLAYOUT_CLASS;H_ROWLAY  
OUT_CLASS;H_RUBYLAYOUT_CLASS;H_SUPERTABLEGROUPLAYOUT_CLASS;H_SUPERTABLELAYOUT  
_CLASS;H_TABLELAYOUT_CLASS;H_ENDNOTELAYOUT_CLASS;H_FOOTNOTELAYOUT_CLASS;H_TABLE  
HEADINGLAYOUT_CLASS;H_TOCSUPERTABLELAYOUT_CLASS',0)} See list of classes
```

(Read-write) Indicates whether a layout will autogrow or autosize in an upward direction.

Data Type

Data type is [Variant](#), which allows the value of this property to be one of the constants listed below or its hexadecimal equivalent (in parentheses). You can combine these values when you want Word Pro to combine the features listed below. Use the OR operator to combine values.

Syntax

```
directionupvalue = [objectreference].DirectionUp
```

```
[objectreference].DirectionUp = directionupvalue
```

Legal values

<u>Value</u>	<u>Effect</u>
LwpLayoutDirectionUseDirection (&H1)	Specifies that a layout object will adhere to other bit values set for this property.
LwpLayoutDirectionAutoGrow (&H2)	Specifies that a layout object autogrows in a upward direction.
LwpLayoutDirectionAutoSize (&H4)	Specifies that a layout object autosizes in a upward direction.
LwpLayoutDirectionTocontainer (&H8)	Specifies that a layout object autogrows or autosizes to the top edge of its container.
LwpLayoutDirectionAllDirections (&HF)	Simultaneously autogrows or autosizes a layout object in all directions (down, up, left, and right).

Usage

Make sure to set the &H1 bit value if you want to enable any of the grow or size options for a layout object. For example, if you want to specify that a layout object should automatically grow, it is not sufficient to set the property value to &H2. You must also set the &H1 bit in order for the other bit settings to be valid. An example statement is shown below:

```
[objectreference].DirectionUp = &H1 OR &H2
```

Word Pro: DisableClickHereAction property

{button ,AL('H_DOCCONTROL_CLASS',0)} [See list of classes](#)

(Read-write)

This language element has not yet been defined. You can check for a newer Help file on the Lotus web site.

Data Type

Integer

Syntax

disableclickhereactionvalue = [objectreference].DisableClickHereAction

[objectreference].DisableClickHereAction = disableclickhereactionvalue

Legal values

The legal values for this property are -1 and 0 but you may use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Word Pro: DisableClickHeres property

{button ,AL('H_DOCCONTROL_CLASS',0)} [See list of classes](#)

{button ,AL('H_DISABLECLICKHERES_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Allows you to edit Click Here prompt text on-screen.

Data Type

[Integer](#)

Syntax

disableclickheresvalue = [objectreference].DisableClickHeres

[objectreference].DisableClickHeres = disableclickheresvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Equivalent to choosing File - TeamSecurity, and selecting "Edit Click Here Block prompts on-screen" on the Other Protection panel.

Word Pro: DisableConsistencyCheck property

{button ,AL('H_USERINTERFACEPREFS_CLASS',0)} [See list of classes](#)

{button ,AL('H_DISABLECONSISTENCYCHECK_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-Write) Sets or returns whether Consistency Check is disabled.

Data Type

[Integer](#)

Syntax

disableconsistencycheckvalue = [objectreference].DisableConsistencyCheck

[objectreference].DisableConsistencyCheck = disableconsistencycheckvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values. Default is False (0).

Usage

Equivalent to the "Consistency Check" option in the "Disable" field on the General panel of the Word Pro Preferences dialog box. If this property is set to True (-1), Word Pro does not perform a consistency check. If set to False (0), Word Pro performs a consistency check.

For more information on Consistency Check, see the [ConsistencyCheck](#) method.

Word Pro: DisableExportToNotes property

{button ,AL('H_DOCCONTROL_CLASS',0)} [See list of classes](#)

{button ,AL('H_DISABLEEXPORTTONOTES_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Prevents information in a document from being exported to Notes.

Data Type

[Integer](#)

Syntax

disableexporttonotesvalue = [objectreference].DisableExportToNotes

[objectreference].DisableExportToNotes = disableexporttonotesvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Equivalent to choosing File - TeamSecurity, and selecting "Disable Notes/FX of TeamSecurity fields" on the Other Protection panel. Checking this option disables Notes/FX and sets this property to False (0); deselecting the option sets it to True (-1).

Word Pro: DisableHarmlessMessages property

{button ,AL('H_USERINTERFACEPREFS_CLASS',0)} [See list of classes](#)

(Read-write)

Data Type

Integer

Syntax

disableharmlessmessagesvalue = [objectreference].DisableHarmlessMessages

[objectreference].DisableHarmlessMessages = disableharmlessmessagesvalue

Legal values

The legal values for this property are -1 and 0 but you may use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Word Pro: DisableVersionReview property

{button ,AL('H_DOCCONTROL_CLASS',0)} [See list of classes](#)

{button ,AL('H_DISABLEVERSIONREVIEW_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Prevents you from viewing versions other than the current version of the document.

Data Type

[Integer](#)

Syntax

disableversionreviewvalue = [objectreference].DisableVersionReview

[objectreference].DisableVersionReview = disableversionreviewvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

A Word Pro document can have multiple editors who are able to view each other's version of a document. However, you can set this property to restrict an editor to a specific version of the document. A True setting is equivalent to choosing File - TeamSecurity, and selecting "Disable version review" on the Other Protection panel. A False setting is equivalent to choosing File - TeamSecurity, and deselecting this option.

Word Pro: DisplayablePageNum property

```
{button ,AL(^H_BASECONTAINER_CLASS;H_CELLCONTAINER_CLASS;H_FRAMECONTAINER_CLASS;H_NOTE  
CONTAINER_CLASS;H_RUBYCONTAINER_CLASS;H_DROPDOWNCONTAINER_CLASS;H_PAGECONTAINER_  
CLASS;H_SUBPAGECONTAINER_CLASS;H_SUPERPAGECONTAINER_CLASS;H_ROWCONTAINER_CLASS;  
H_SUPERTABLECONTAINER_CLASS;H_TABLECONTAINER_CLASS;H_PARALLELCOLSCONTAINER_CLASS  
;H_TABLEONLYCONT_CLASS',0)} See list of classes
```

(Read-only)

This language element has not yet been defined. You can check for a newer Help file on the Lotus web site.

Data Type

Integer

Syntax

```
displayablepagenumvalue = [objectreference].DisplayablePageNum
```

Legal values

This property is read-only. The value of this property cannot be set by a script.

Usage

Word Pro: DisplayAsIcon property

{button ,AL(^H_OLEOBJECT_CLASS',0)} [See list of classes](#)

(Read-write)

Data Type

Data type is Integer.

Syntax

displayasiconvalue = [objectreference].DisplayAsIcon

Legal values**Usage**

Word Pro: DistanceFromLeftMargin property

{button ,AL(^H_LINENUMBEROPTIONS_CLASS',0)} [See list of classes](#)

(Read-write)

Data Type

Long

Syntax

distancefromleftmarginvalue = [objectreference].DistanceFromLeftMargin

[objectreference].DistanceFromLeftMargin = distancefromleftmarginvalue

Legal values

Data type is Long but the unit of measurement used for this property is Twips. There are 1440 Twips per inch.

Usage

Word Pro: DivisionInfo property

{button ,AL(^H_BASECONTAINER_CLASS;H_CELLCONTAINER_CLASS;H_FRAMECONTAINER_CLASS;H_NOTECONTAINER_CLASS;H_RUBYCONTAINER_CLASS;H_DROPDOWNCONTAINER_CLASS;H_PAGECONTAINER_CLASS;H_SUBPAGECONTAINER_CLASS;H_SUPERPAGECONTAINER_CLASS;H_ROWCONTAINER_CLASS;H_SUPERTABLECONTAINER_CLASS;H_TABLECONTAINER_CLASS;H_PARALLELCOLSCONTAINER_CLASS;H_TABLEONLYCONT_CLASS;H_DIVISION_CLASS;H_MARKER_CLASS;H_POWERFIELD_CLASS;H_CLICKHERE_CLASS;H_TEXTMARKER_CLASS;H_RUBYMARKER_CLASS;H_TABLEMARKER_CLASS;H_TEXTDOCUMENT_CLASS',0)} [See list of classes](#)

(Read-only) Returns the division info object from any container, division, or marker object.

Data Type

[DivisionInfo](#)

Syntax

divisioninfovalue = [objectreference].DivisionInfo

Legal values

Always contains an instance of the DivisionInfo class.

Usage

Word Pro: DivisionNames property

{button ,AL(^H_DIVISION_CLASS;H_TEXTDOCUMENT_CLASS',0)} [See list of classes](#)

(Read-only)

Data Type

[StringCollection](#)

Syntax

divisionnamesvalue = [objectreference].DivisionNames

Legal values

Always contains an instance of the StringCollection class.

Usage

Word Pro: DivisionName property

{button ,AL('H_BASECONTAINER_CLASS;H_CELLCONTAINER_CLASS;H_FRAMECONTAINER_CLASS;H_NOTECONTAINER_CLASS;H_RUBYCONTAINER_CLASS;H_DROPDOWNCONTAINER_CLASS;H_PAGECONTAINER_CLASS;H_SUBPAGECONTAINER_CLASS;H_SUPERPAGECONTAINER_CLASS;H_ROWCONTAINER_CLASS;H_SUPERTABLECONTAINER_CLASS;H_TABLECONTAINER_CLASS;H_PARALLELCOLSCONTAINER_CLASS;H_TABLEONLYCONT_CLASS;H_TOCSUPERTABLELAYOUT_CLASS',0)} [See list of classes](#)

{button ,AL('H_DIVISIONNAME_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-only) Returns the external user name for a division object.

Data Type

String

Syntax

divisionnamevalue = [objectreference].DivisionName

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Use this property to return the division name that displays on a division tab.

Word Pro: DivisionOptions property

{button ,AL('H_DIVISION_CLASS;H_TEXTDOCUMENT_CLASS',0)} [See list of classes](#)

{button ,AL('H_DIVISIONOPTIONS_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-only)

Data Type

[DivisionOptions](#)

Syntax

divisionoptionsvalue = [objectreference].DivisionOptions

Legal values

Always contains an instance of the DivisionOptions class.

Usage

Word Pro: Divisions property

{button ,AL(^H_WPAPPLICATION_CLASS;H_DIVISION_CLASS;H_TEXTDOCUMENT_CLASS;H_PRINTSETTINGS_CLASS',0)} [See list of classes](#)

(Read-only) Contains a DivisionCollection object. This DivisionCollection object contains Division objects.

Data Type

[DivisionCollection](#)

Syntax

divisionsvalue = [objectreference].Divisions

Legal values

Always contains an instance of the DivisionCollection class.

Usage

When you call this property from the WPAApplication object, the DivisionCollection object contains all the divisions in the currently active document.

Word Pro: Division property

{button ,AL(^H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

(Read-only) Contains the currently active Division object.

Data Type

[Division](#)

Syntax

divisionvalue = [objectreference].Division

Legal values

Always contains an instance of the Division class.

Usage

Use this property to access the currently active Division object and any of its members.

Word Pro: DocControlPassword property

{button ,AL('H_DOCCONTROL_CLASS','0')} [See list of classes](#)

(Write-only)

The password for the TeamSecurity dialog box.

Data Type

String

Syntax

```
doccontrolpasswordvalue = [objectreference].DocControlPassword
```

Legal values

Any value of type String.

This property is Write-only so script cannot discover the existing value.

Usage

To prevent a script from circumventing TeamSecurity measures already in place, this property will only work under one of the following conditions:

```
.ActiveDocument.DocControl.DocControlProtection = $LwpDocProtectTypeNone
```

or

```
.ActiveDocument.DocControl.DocControlProtection = $LwpDocProtectTypeOnlyeditor  
and .ActiveDocument.DocControl.DocControlRestrictedToEditor matches the current User Name  
(.Application.Preferences.UserName).
```

Word Pro: DocControlProtection property

{button ,AL('H_DOCCONTROL_CLASS','0)} [See list of classes](#)

{button ,AL('H_DOCCONTROLPROTECTION_PROPERTY_EXSCRIPT',1)} [See example](#)

(Write-only) Determines the status of the protection of the TeamSecurity dialog box.

Data Type

The data type for this property is [Variant](#) which allows the value of this property to be one of the constants listed below or its numeric equivalent (in parentheses).

Syntax

```
doccontrolprotectionvalue = [objectreference].DocControlProtection
```

Legal values

The legal values for this property are listed below. You can use the constant or the numeric value when setting this property. Word Pro always returns the numeric value. You can specify only one of these values at any time.

<u>Value</u>	<u>Effect</u>
\$LwpDocProtectTypeNone (187)	No protection of the TeamSecurity dialog box.
\$LwpDocProtectTypeOnlyEditor (188)	Only the editor has access to the TeamSecurity dialog box.
\$LwpDocProtectTypePassword (189)	Only those with the TeamSecurity dialog box password can access the TeamSecurity dialog box.

Usage

You can use this property in conjunction with other DocControl properties to set the document password.

For example, you could use the following lines of code to set the document password:

```
.ActiveDocument.DocControl.FileProtectionType = $LwpFileProtectTypePassword  
.ActiveDocument.DocControl.FilePassword = "whatever"
```

Setting the password will only work when the DocControlProtection property has the value \$LwpDocProtectTypeNone or it has a value of \$LwpDocProtectTypeOnlyeditor and the DocControlRestrictedToEditor property matches the current User Name. The current User Name is accessible in the UserName property of the Preferences class.

The reason for this restriction is so that script can not circumvent the security already in place on a document. In fact, none of the DocControl properties (including this one) are accessible to script if this condition is not met.

Note When you set DocControlProtectionType to a value other than \$LwpDocProtectTypeNone, LotusScript access to the other DocControl properties is immediately restricted. Therefore, make any necessary changes to properties of the DocControl class before modifying the DocControlProtectionType property.

Word Pro: DocControlRestrictedToEditor property

{button ,AL('H_DOCCONTROL_CLASS',0)} [See list of classes](#)

{button ,AL('H_DOCCONTROLRESTRICTEDTOEDITOR_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-only) Allows only a specific editor to set or modify document control options for a document.

Data Type

[String](#)

Syntax

doccontrolrestrictedtoeditorvalue = [objectreference].DocControlRestrictedToEditor

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Equivalent to choosing File - TeamSecurity, selecting "Only" and then the editor's name in the "Only" box on the Access panel.

Word Pro: DocControl property

{button ,AL(^H_DIVISION_CLASS;H_TEXTDOCUMENT_CLASS',0)} [See list of classes](#)

{button ,AL(^H_DOCCONTROL_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-only)

Data Type

[DocControl](#)

Syntax

doccontrolvalue = [objectreference].DocControl

Legal values

Always contains an instance of the DocControl class.

Usage

Word Pro: DocInfo property

{button ,AL(^H_DIVISION_CLASS;H_TEXTDOCUMENT_CLASS',0)} [See list of classes](#)

(Read-only)

Data Type

[DocInfo](#)

Syntax

docinfovalue = [objectreference].DocInfo

Legal values

Always contains an instance of the DocInfo class.

Usage

Word Pro: DocOptions property

{button ,AL(^H_DIVISION_CLASS;H_TEXTDOCUMENT_CLASS',0)} [See list of classes](#)

(Read-only)

Data Type

[Options](#)

Syntax

docoptionsvalue = [objectreference].DocOptions

Legal values

Always contains an instance of the Options class.

Usage

Word Pro: DocPath property

{button ,AL(^H_USERINTERFACEPREFS_CLASS',0)} [See list of classes](#)

(Read-write) Sets or returns the default path (drive and directory) to which Word Pro looks when opening documents.

Data Type

String

Syntax

docpathvalue = [objectreference].DocPath

[objectreference].DocPath = docpathvalue

Legal values

A valid path including drive and deirectory.

Usage

Equivalent to the "Documents" field on the Locations panel of the Word Pro Preferences dialog box. In Word Pro, the "Documents" field can contain multiple paths. You can use this property to clear all paths before setting the default or first document path, or you can use the property, DocumentPaths, to read any paths entered by the user.

Word Pro: DocSize property

{button ,AL('H_DOCINFO_CLASS',0)} [See list of classes](#)

{button ,AL('H_DOCSIZE_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read only) Returns the size of a document in bytes.

Data Type

Long

Syntax

docsizevalue = [objectreference].DocSize

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Word Pro: DocumentLevel property

{button ,AL('H_PARAGRAPHSTYLE_CLASS',0)} [See list of classes](#)

{button ,AL('H_DOCUMENTLEVEL_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write)

Data Type

[Integer](#)

Syntax

documentlevelvalue = [objectreference].DocumentLevel

[objectreference].DocumentLevel = documentlevelvalue

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Word Pro: DocumentPaths property

{button ,AL('H_USERINTERFACEPREFS_CLASS',0)} [See list of classes](#)

(Read-only) Stores multiple paths (drive and directory) where Word Pro looks for documents to open.

Data Type

[StringCollection](#)

Syntax

documentpathsvalue = [objectreference].DocumentPaths

Legal values

Always contains an instance of the StringCollection class.

Usage

Equivalent to the "Documents" field on the Locations panel of the Word Pro Preferences dialog box. In Word Pro, the "Documents" field can contain multiple document paths. This property returns a collection of String objects that contains the names of all default Word Pro document paths.

Word Pro: Documents property

{button ,AL(^H_APPLICATION_CLASS;H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

(Read-only) Contains a Documents object. This Documents object is a collection object that contains all the currently open TextDocument objects.

Data Type

[Documents](#)

Syntax

documentsvalue = [objectreference].Documents

Legal values

Always contains an instance of the Documents class.

Usage

Use this property when you want to access a TextDocument object which is open but not currently active. For more information about collection classes, see [Overview: Word Pro LotusScript Collection Classes](#).

Word Pro: Document property

{button ,AL(^H_DOCWINDOW_CLASS',0)} [See list of classes](#)

(Read-only)

Data Type

[TextDocument](#)

Syntax

documentvalue = [objectreference].Document

Legal values

Always contains an instance of the TextDocument class.

Usage

Word Pro: DocVersionID property

{button ,AL(^H_VERSION_CLASS;H_VERSIONMANAGER_CLASS',0)} [See list of classes](#)

(Read-only)

Data Type

Long

Syntax

docversionidvalue = [objectreference].DocVersionID

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Word Pro: DocWindows property

{button ,AL('H_APPLICATIONWINDOW_CLASS',0)} [See list of classes](#)

(Read-only) An instance of the DocWindowCollection class that is a collection of all open document windows in an application.

Data Type

[DocWindowCollection](#)

Syntax

docwindowsvalue = [objectreference].DocWindows

Legal values

Always contains an instance of the DocWindowCollection class.

Usage

Use this property to get a list of all open document windows in the currently active application window.

Word Pro: DoInitialCaps property

{button ,AL('H_SMARTCORRECT_CLASS',0)} [See list of classes](#)

{button ,AL('H_DOINITIALCAPS_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Determines whether or not two initial capital letters are replaced by a single initial capital letter.

Data Type

[Integer](#)

Syntax

doinitialcapsvalue = [objectreference].DoInitialCaps

[objectreference].DoInitialCaps = doinitialcapsvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Equivalent to the "Correct TWo INitial CAPitals" option in the SmartCorrect dialog box.

Word Pro: DoSentenceInitialCaps property

{button ,AL('H_SMARTCORRECT_CLASS',0)} [See list of classes](#)

{button ,AL('H_DOSENTENCEINITIALCAPS_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Determines if SmartCorrect automatically capitalizes first word in sentence

Data Type

Integer

Syntax

dosentenceinitialcapsvalue = [objectreference].DoSentenceInitialCaps

[objectreference].DoSentenceInitialCaps = dosentenceinitialcapsvalue

Legal values

The legal values for this property are -1 and 0 but you may use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Word Pro: DoSmartLinks property

{button ,AL('H_SMARTCORRECT_CLASS',0)} [See list of classes](#)

{button ,AL('H_DOSMARTLINKS_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write)

This language element has not yet been defined. You can check for a newer Help file on the Lotus web site.

Data Type

Integer

Syntax

dosmartlinksvalue = [objectreference].DoSmartLinks

[objectreference].DoSmartLinks = dosmartlinksvalue

Legal values

The legal values for this property are -1 and 0 but you may use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Word Pro: DoSmartQuotes property

{button ,AL('H_SMARTCORRECT_CLASS',0)} [See list of classes](#)

{button ,AL('H_DOSMARTQUOTES_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Determines whether or not Word Pro replaces straight quotes with smart quotes.

Note Smart quotes are not supported in Word Pro for OS/2 Warp 4.

Data Type

Integer

Syntax

dosmartquotesvalue = [objectreference].DoSmartQuotes

[objectreference].DoSmartQuotes = dosmartquotesvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Smart quotes are typographical replacements for straight quotes. If this property is set to True, Word Pro replaces straight quotes with smart quotes, if they occur at the beginning or end of a word.

Word Pro: DoubleUnderline property

{button ,AL('H_FONT_CLASS',0)} [See list of classes](#)

{button ,AL('H_DOUBLEUNDERLINE_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write)

Data Type

[Integer](#)

Syntax

doubleunderlinevalue = [objectreference].DoubleUnderline

[objectreference].DoubleUnderline = doubleunderlinevalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Word Pro: DownloadGraphics property

{button ,AL('H_HTMLOPTIONS_CLASS',0)} [See list of classes](#)

{button ,AL('H_DOWNLOADGRAPHICS_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write)

This language element has not yet been defined. You can check for a newer Help file on the Lotus web site.

Data Type

Integer.

Syntax

downloadgraphicsvalue = [objectreference].DownloadGraphics

[objectreference].DownloadGraphics = downloadgraphicsvalue

Legal values

The legal values for this property are -1 and 0 but you may use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Equivalent to choosing File - Internet - HTML Import Options, and selecting "Download graphics from the Internet."

Word Pro: DragDropOn property

{button ,AL('H_PREFERENCES_CLASS',0)} [See list of classes](#)

{button ,AL('H_DRAGDROPON_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write)

Data Type

[Integer](#)

Syntax

dragdroponvalue = [objectreference].DragDropOn

[objectreference].DragDropOn = dragdroponvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Word Pro: DriverName property

{button ,AL('H_PRINTMANAGER_CLASS',0)} [See list of classes](#)

(Read-write)

Data Type

Data type is String.

Syntax

[objectreference].DriverName = drivenamevalue

drivenamevalue = [objectreference].DriverName

Legal values

Usage

Word Pro: DropCapPosition property

{button ,AL(^H_LAYOUT_CLASS;H_CELLGROUPLAYOUT_CLASS;H_CELLLAYOUT_CLASS;H_CONNECTEDLAY
OUT_CLASS;H_COLUMNGROUPLAYOUT_CLASS;H_FOOTERLAYOUT_CLASS;H_FRAMELAYOUT_CLASS;H
_FRAMEGROUPLAYOUT_CLASS;H_DROPcapLAYOUT_CLASS;H_GROUPLAYOUT_CLASS;H_HEADERLAY
OUT_CLASS;H_NOTELAYOUT_CLASS;H_PAGELAYOUT_CLASS;H_ROWGROUPLAYOUT_CLASS;H_ROWLA
YOUT_CLASS;H_RUBY_LAYOUT_CLASS;H_SUPERTABLEGROUPLAYOUT_CLASS;H_SUPERTABLELAYOUT
_CLASS;H_TABLELAYOUT_CLASS;H_ENDNOTELAYOUT_CLASS;H_FOOTNOTELAYOUT_CLASS;H_TABLE
HEADINGLAYOUT_CLASS;H_TOCSUPERTABLELAYOUT_CLASS',0)} [See list of classes](#)

(Read-write) Controls where a DropCap layout object is placed in relation to the first line of text.

Data Type

Data type is [Variant](#), which allows the value of this property to be one of the constants listed below or its numeric equivalent (in parentheses).

Syntax

dropcappositionvalue = [objectreference].DropCapPosition

[objectreference].DropCapPosition = dropcappositionvalue

Legal values

<u>Value</u>	<u>Effect</u>
\$LwpDropcapAbove (2235)	Equivalent to choosing "Above first line" in the Drop Cap dialog box.
\$LwpDropcapBelow (2234)	Equivalent to choosing "Below first line" in the Drop Cap dialog box.
\$LwpDropcapBeside (2236)	Equivalent to choosing "Beside paragraph" in the Drop Cap dialog box.

Usage

Word Pro: DropCapStyleName property

{button ,AL(^H_PREFERENCES_CLASS',0)} [See list of classes](#)

(Read-write)

Data Type

Data type is String.

Syntax

[objectreference].DropCapStyleName = dropcapstylenamevalue

dropcapstylenamevalue = [objectreference].DropCapStyleName

Legal values

Usage

Word Pro: DropCapStyles property

{button ,AL('H_FOUNDRY_CLASS',0)} [See list of classes](#)

(Read-only)

Data Type

DropCapLayoutCollection

Syntax

dropcapstylesvalue = [objectreference].DropCapStyles

Legal values

Usage

Word Pro: DropCaps property

{button ,AL('H_FOUNDRY_CLASS',0)} [See list of classes](#)

(Read-only)

Data Type

DropCapLayoutCollection

Syntax

dropcapsvalue = [objectreference].DropCaps

Legal values**Usage**

Word Pro: Editable property

{button ,AL(^H_BULLET_CLASS',0)} [See list of classes](#)

(Read-write)

Data Type

Variant (Enumerated)

CommandState

Syntax

editablevalue = [objectreference].Editable

[objectreference].Editable = editablevalue

Legal values

\$LwpCommandStateOff (151)

\$LwpCommandStateOn (152)

\$LwpCommandStateStyle (153)

Usage

Word Pro: EditorInitials property

{button ,AL('H_EDITOR_CLASS',0)} [See list of classes](#)

(Read-only) Returns the initials of the editor assigned to a document.

Data Type

String

Syntax

editorinitialsvalue = [objectreference].EditorInitials

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Equivalent to the "Initials" box on the Personal panel of the Word Pro Preferences dialog box.

Word Pro: EditorManager property

{button ,AL(^H_DIVISION_CLASS;H_TEXTDOCUMENT_CLASS',0)} [See list of classes](#)

{button ,AL(^H_EDITORMANAGER_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-only)

Data Type

[EditorManager](#)

Syntax

editormanagervalue = [objectreference].EditorManager

Legal values

Always contains an instance of the EditorManager class.

Usage

Word Pro: EditorName property

```
{button ,AL(^H_DIVISION_CLASS;H_LAYOUT_CLASS;H_CELLGROUPLAYOUT_CLASS;H_CELLLAYOUT_CLASS;  
H_CONNECTEDLAYOUT_CLASS;H_COLUMNGROUPLAYOUT_CLASS;H_FOOTERLAYOUT_CLASS;H_FRAM  
ELAYOUT_CLASS;H_FRAMEGROUPLAYOUT_CLASS;H_DROPAPLAYOUT_CLASS;H_GROUPLAYOUT_CL  
ASS;H_HEADERLAYOUT_CLASS;H_NOTELAYOUT_CLASS;H_PAGELAYOUT_CLASS;H_ROWGROUPLAYOU  
T_CLASS;H_ROWLAYOUT_CLASS;H_RUBYLAYOUT_CLASS;H_SUPERTABLEGROUPLAYOUT_CLASS;H_S  
UPERTABLELAYOUT_CLASS;H_TABLELAYOUT_CLASS;H_ENDNOTELAYOUT_CLASS;H_FOOTNOTELAYOU  
T_CLASS;H_TABLEHEADINGLAYOUT_CLASS;H_TOCSUPERTABLELAYOUT_CLASS;H_TEXTDOCUMENT_C  
LASS;H_VERSION_CLASS',0)} See list of classes
```

```
{button ,AL(^H_EDITORNAME_PROPERTY_EXSCRIPT',1)} See example
```

(Read-only) Returns the user name of the person who created a particular layout object.

Data Type

String

Syntax

editornamevalue = [objectreference].EditorName

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Stores the name of the person who created a layout object. In Word Pro, the editor name value can be accessed in the "User name" box on the Personal panel of the Word Pro Preferences dialog box.

Word Pro: Editors property

{button ,AL(^H_EDITORMANAGER_CLASS;H_VERSION_CLASS',0)} [See list of classes](#)

(Read-only) The collection of editors assigned to a document.

Data Type

[EditorCollection](#)

Syntax

editorsvalue = [objectreference].Editors

Legal values

Always contains an instance of the EditorCollection class.

Usage

[EditorManager]

The Editors property contains all editors that are assigned to a document. Equivalent to the editors listed in the Editing Rights panel of the TeamSecurity dialog.

Word Pro: EditorVerificationType property

{button ,AL('H_DOCCONTROL_CLASS',0)} [See list of classes](#)

{button ,AL('H_EDITORVERIFICATIONTYPE_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-only) Verifies how an editor gains access to a document by using either e-mail login, operating system login, or user name.

Data Type

Data type is [Variant](#), which allows the value of this property to be one of the constants listed below or its numeric equivalent (in parentheses).

Syntax

editorverificationtypevalue = [objectreference].EditorVerificationType

Legal values

<u>Value</u>	<u>Effect</u>
\$LwpEditorVerifyTypeByEmailLogin (236)	An editor must use an e-mail login to access a document.
\$LwpEditorVerifyTypeByOpsysLogin (235)	An editor must use an operating system login to access a document.
\$LwpEditorVerifyTypeByUserName (234)	An editor must use a user name to access a document.

Usage

Equivalent to choosing File - TeamSecurity and selecting a verification type (e-mail login, operating system login, Word Pro user name) in the "Verify editors using" box on the Access panel.

Word Pro: EffectiveColumnWidth property

{button ,AL(^H_CLICKHERE_CLASS;H_TEXTMARKER_CLASS;H_TEXT_CLASS',0)} [See list of classes](#)

(Read-only) The column width less any indentation for the paragraph in which the insertion point is located.

Data Type

Long

Syntax

effectivecolumnwidthvalue = [objectreference].EffectiveColumnWidth

Legal values

Data type is Long but the unit of measurement used for this property is Twips. There are 1440 Twips per inch.

Usage

Word Pro: EMail property

{button ,AL('H_PREFERENCES_CLASS',0)} [See list of classes](#)

(Read-write)

Data Type

String

Syntax

emailvalue = [objectreference].EMail

[objectreference].EMail = emailvalue

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Word Pro: Embedded property

{button ,AL(^H_DOCUMENT_CLASS;H_TEXTDOCUMENT_CLASS',0)} [See list of classes](#)

(Read-only)

Data Type

Integer

Syntax

embeddedvalue = [objectreference].Embedded

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Word Pro: EmbedFonts property

{button ,AL('H_EMBEDFONTS_PROPERTY_EXSCRIPT',1)} [See example](#)

{button ,AL('H_OPTIONS_CLASS',0)} [See list of classes](#)

(Read-write) Enables Word Pro to embed fonts that are used in the current document within the current document when it is saved.

Data Type

Integer

Syntax

embedfontsvvalue = [objectreference].EmbedFonts

[objectreference].EmbedFonts = embedfontsvvalue

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

1 - enables "Embed Fonts in document"

0 - disables "Embed Fonts in document"

Usage

Equivalent to choosing Document Properties - Document, clicking the Options tab, and selecting "Embed Fonts in document."

Enabling this option saves the fonts used in the current file with the file. When another user opens the document, Word Pro automatically installs the embedded fonts on that user's computer (in the Windows System folder), provided the fonts are not already installed. Enabling this option also ensures that all users who open the file see the fonts you used, even if not everyone has installed those fonts.

Note Word Pro cannot embed read-only or protected fonts. Word Pro does not display a message when it cannot embed a font.

Word Pro: EmitsEvents property

{button ,AL(^H_OLEOBJECT_CLASS',0)} [See list of classes](#)

(Read-only)

Indicates whether or not an OLEObject emits events.

Data Type

Integer

Syntax

emitseventsvalue = [objectreference].EmitsEvents

Legal values

A value of -1 indicates that the OLEObject emits events. A value of 0 indicates that the object does not emit events. This property is read-only. The value of this property cannot be set by a script.

Usage

It is usually OLE controls which emit events. Most OLE controls such as command buttons, edit boxes, radio buttons, and so on emit events for which you can write scripts. Once a control is named, it appears in the Object list in the Script Editor. Its events appear in the Script list when that control is selected.

Word Pro: Enabled property

{button ,AL('H_MENUITEM_CLASS;H_LWPTIMER_CLASS',0)} [See list of classes](#)

(Read-write)

[MenuItem]

Indicates if a menu item is not active (grayed) or active (not grayed).

Data Type

Integer

Syntax

enabledvalue = [objectreference].Enabled

[objectreference].Enabled = enabledvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

To gray a menu item, set this property to False. To ungray a menu item, set this property to True.

Setting this property is available only for custom created menu items. You cannot set this property for predefined Word Pro menu items, because Word Pro dynamically sets the Enabled property for predefined menu items, based on the current context.

Word Pro: EndingColOfSelection property

{button ,AL(^H_BASSETABLE_CLASS;H_FOOTNOTETABLE_CLASS;H_PARALLELCOLUMNS_CLASS;H_GLOSSARY_CLASS;H_TABLEHEADING_CLASS;H_TABLE_CLASS',0)} [See list of classes](#)

(Read-only) Returns the number of the last column included in a selection of table cells.

Data Type

Integer

Syntax

endingcolofselectionvalue = [objectreference].EndingColOfSelection

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

The column ID is a zero-based value, which means that the first column in a table has a zero ID value.

Word Pro: EndingRowOfSelection property

{button ,AL(^H_BASETABLE_CLASS;H_FOOTNOTETABLE_CLASS;H_PARALLELCOLUMNS_CLASS;H_GLOSSARY_CLASS;H_TABLEHEADING_CLASS;H_TABLE_CLASS',0)} [See list of classes](#)

(Read-only) Returns the number of the last row included in a selection of table cells.

Data Type

Integer

Syntax

endingrowofselectionvalue = [objectreference].EndingRowOfSelection

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

The row ID is a zero-based value, which means that the first row in a table has a zero ID value.

Word Pro: EndnoteDivisionGroupNum property

{button ,AL('H_FOOTNOTEOPTIONS_CLASS',0)} [See list of classes](#)

{button ,AL('H_ENDNOTEDIVISIONGROUPNUM_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-only) Sets or returns the EndnoteDivisionGroupNum object of the specified object.

Data Type

[EndnoteDivisionGroupNum](#)

Syntax

endnotedivisiongroupnumvalue = [objectreference].EndnoteDivisionGroupNum

Legal values

Always contains an instance of the EndnoteDivisionGroupNum class.

Usage

Word Pro: EndnoteDivisionNum property

{button ,AL('H_FOOTNOTEOPTIONS_CLASS',0)} [See list of classes](#)

{button ,AL('H_ENDNOTEDIVISIONNUM_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-only) Sets or returns the EndnoteDivisionNum object for the specified object.

Data Type

[EndnoteDivisionNum](#)

Syntax

endnotedivisionnumvalue = [objectreference].EndnoteDivisionNum

Legal values

Always contains an instance of the EndnoteDivisionNum class.

Usage

Word Pro: EndnoteDocNum property

{button ,AL('H_FOOTNOTEOPTIONS_CLASS',0)} [See list of classes](#)

{button ,AL('H_ENDNOTEDOCNUM_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-only) Sets or returns the EndnoteDocNum object for the specified object.

Data Type

[EndnoteDocNum](#)

Syntax

endnotedocnumvalue = [objectreference].EndnoteDocNum

Legal values

Always contains an instance of the EndnoteDocNum class.

Usage

Word Pro: Endnotes property

{button ,AL('H_FOUNDRY_CLASS',0)} [See list of classes](#)

(Read-only) An object created from the EndnoteLayoutCollection class. This object provides access to EndnoteLayoutCollection objects.

Data Type

[EndnoteLayoutCollection](#)

Syntax

endnotesvalue = [objectreference].Endnotes

Legal values

Always contains an instance of the EndnoteLayoutCollection class.

Usage

When accessed through the Foundry property on a Division object, the collection object in this property provides access to all the EndnoteLayoutCollection objects contained in that Division object.

When accessed through the AppFoundry property on the WPAApplication object, this collection object provides access to all the EndnoteLayoutCollection objects contained in the Word Pro Clipboard.

When accessed through the TempFoundry property on the WPAApplication object, this collection object provides access to all the EndnoteLayoutCollection objects placed in TempFoundry by WordPro or a script.

When accessed through the Foundry property on the WPAApplication object, this collection object provides access to all the EndnoteLayoutCollection objects contained in the currently active Division object.

This property is not used in the Foundry property on the TextDocument object.

For more information about collection classes, see [Overview: Word Pro LotusScript Collection Classes](#).

Word Pro: EntireDocument property

{button ,AL('H_HTMLOPTIONS_CLASS',0)} [See list of classes](#)

{button ,AL('H_ENTIREDOCUMENT_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write)

This language element has not yet been defined. You can check for a newer Help file on the Lotus web site.

Data Type

Integer.

Syntax

entiredocumentvalue = [objectreference].EntireDocument

[objectreference].EntireDocument = entiredocumentvalue

Legal values

The legal values for this property are -1 and 0 but you may use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Equivalent to choosing File - Internet - HTML Export Assistant, and on the Content panel, selecting "Entire document" or "Current division" in the "Export" section.

Word Pro: Enumeration property

{button ,AL('H_FOUNDRY_CLASS',0)} [See list of classes](#)

{button ,AL('H_ENUMERATION_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Tells Word Pro which objects to include in the specified Foundry's collection objects.

Data Type

Variant (Enumerated)

EnumScope

Syntax

enumerationvalue = [objectreference].Enumeration

[objectreference].Enumeration = enumerationvalue

Legal values

<u>Value</u>	<u>Effect</u>
\$LwpEnumScopeAll (237)	Includes both types of objects in this Foundry's collection objects.
\$LwpEnumScopeLocal (238)	Includes only those objects which are not part of a SmartMaster.
\$LwpEnumScopeStyle (239)	Includes only those objects which are part of a SmartMaster.

Note Do not use quotation marks around the string values above. You can also use the numeric equivalents for these values (shown in parentheses).

Usage

Each Word Pro document is first created from a SmartMaster. All objects that comprise the SmartMaster are known as style objects. These objects might include ParagraphStyle, CharacterStyle, PageLayout, TableLayout, FrameLayout, and so on. When you add new objects to your document, such as a new paragraph of text, a new paragraph style, or a new page layout that were not part of the original SmartMaster, Word Pro sees that object as a local object. Both style objects and local objects can be instantiated from the same class, but Word Pro sees them as either a part or not a part of the SmartMaster.

For example, when you create a document whose SmartMaster contains a division with a paragraph style named Body Text, you can create a new paragraph style called Title. When you looked into the ParagraphStyleCollection object for that division, you would see two ParagraphStyle objects. Word Pro differentiates automatically between the style objects and the local objects. However, it treats them the same within the context of LotusScript.

You may find it useful to exclude objects which originated with the SmartMaster from your collections. Setting the value of the Enumeration property to \$LwpEnumScopeLocal hides the SmartMaster objects from the Script Editor. By default, Word Pro stores both the style objects and the local objects in their appropriate collection objects.

Word Pro: Epoch property

{button ,AL(^H_DIVISION_CLASS;H_TEXTDOCUMENT_CLASS',0)} [See list of classes](#)

(Read-only)

Data Type

String

Syntax

epochvalue = [objectreference].Epoch

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Word Pro: EqnFontHeight property

{button ,AL(^H_GRAPHIC_CLASS',0)} [See list of classes](#)

(Read-only)

Data Type

Long

Syntax

eqnfontheightvalue = [objectreference].EqnFontHeight

Legal values

Data type is Long but the unit of measurement used for this property is Twips.

Usage

Word Pro: Expandable property

{button ,AL('H_TEXTMARKER_CLASS',0)} [See list of classes](#)

{button ,AL('H_EXPANDABLE_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write)

Data Type

[Integer](#)

Syntax

expandablevalue = [objectreference].Expandable

[objectreference].Expandable = expandablevalue

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

{button ,AL('H_MARK_METHOD_MEMDEF;H_ISCOLLAPSIBLE_PROPERTY_MEMDEF',0)} [See related topics](#)

Word Pro: ExportToNotesFX property

{button ,AL('H_DOCINFOFIELD_CLASS',0)} [See list of classes](#)

{button ,AL('H_EXPORTTONOTESFX_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Indicates whether or not a DocInfo field should be available for exchange through Notes/FX.

Data Type

[Integer](#)

Syntax

exporttonotesFXvalue = [objectreference].ExportToNotesFX

[objectreference].ExportToNotesFX = exporttonotesFXvalue

Legal values

The legal values for this property are 0 and 1.

Usage

This property allows you to determine whether a user-defined document field is exported as Notes/FX.

For example, if you have a DocInfo field named Client and you want to export that field to Notes, you can set the field's ExportToNotesFX property to 1. Notes then makes the Client field available for exchange through Notes/FX.

Set this property to 0 if you do not want a document field to be available for exchange through Notes/FX.

{button ,AL('H_EXPORTASNOTESFX_METHOD_MEMDEF',0)} [See related topics](#)

Word Pro: ExternallyControlledUndo property

{button ,AL(^H_GRAPHICOLEBJECT_CLASS;H_GRAPHIC_CLASS;H_OLEOBJECT_CLASS',0)} [See list of classes](#)

(Read-only)

Data Type

Data type is Integer.

Syntax

externallycontrolledundovalue = [objectreference].ExternallyControlledUndo

Legal values**Usage**

Word Pro: ExternalName property

{button ,AL(^H_DIVISIONINFO_CLASS;H_SECTIONTABS_CLASS',0)} [See list of classes](#)

{button ,AL(^H_EXTERNALNAME_PROPERTY_EXSCRIPT',1)} [See example](#)

[SectionTabs]

(Read-only) The user-assigned name that displays on the current section or division tab.

[DivisionInfo]

(Read-write)

Data Type

String

Syntax

externalnamevalue = [objectreference].ExternalName

[objectreference].ExternalName = externalnamevalue

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

[SectionTabs]

You cannot change the name of a section or division tab by using this LotusScript property. You can change the section or division tab name by double clicking the section or division tab, typing a name, and clicking outside the tab or pressing ENTER. If you want to change the name of a division or section with LotusScript, you can use the Name property of the DivisionInfo class.

Word Pro: ExternalType property

{button ,AL('H_DIVISIONINFO_CLASS',0)} [See list of classes](#)

(Read-write)

Data Type

String

Syntax

externaltypevalue = [objectreference].ExternalType

[objectreference].ExternalType = externaltypevalue

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Word Pro: FaceNames property

{button ,AL('^H_PRINTMANAGER_CLASS',0)} [See list of classes](#)

(Read-only)

Data Type

[StringCollection](#)

Syntax

facenamesvalue = [objectreference].FaceNames

Legal values

Always contains an instance of the StringCollection class.

Usage

Word Pro: FaceName property

{button ,AL(^H_FONT_CLASS',0)} [See list of classes](#)

(Read-write)

Data Type

String

Syntax

[objectreference].FaceName = facenamevalue

facenamevalue = [objectreference].FaceName

Legal values**Usage**

Word Pro: FaceStyleName property

{button ,AL('H_CHARACTERSTYLE_CLASS;H_PARAGRAPHSTYLE_CLASS',0)} [See list of classes](#)

(Read-write) The name of the style of

Data Type

String

Syntax

[objectreference].FaceStyleName()

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Word Pro: FaxNumber property

{button ,AL('H_PREFERENCES_CLASS',0)} [See list of classes](#)

{button ,AL('H_FAXNUMBER_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write)

Data Type

String

Syntax

faxnumbervalue = [objectreference].FaxNumber

[objectreference].FaxNumber = faxnumbervalue

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Word Pro: FieldDelimiterText property

{button ,AL('H_SORTOPTIONS_CLASS',0)} [See list of classes](#)

{button ,AL('H_FIELDDELIMITERTEXT_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Allows you to specify the string character used to delimit sort fields in a document.

Data Type

[String](#)

Syntax

fielddelimitertextvalue = [objectreference].FieldDelimiterText

[objectreference].FieldDelimiterText = fielddelimitertextvalue

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Equivalent to choosing Text - Sort, selecting "Text" and entering a string character in the box next to it.

Word Pro: FieldDelimiter property

{button ,AL('H_SORTOPTIONS_CLASS',0)} [See list of classes](#)

{button ,AL('H_FIELDDELIMITER_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Allows you to indicate whether a string character or a tab is used to find sort fields in a document.

Data Type

Data type is [Variant](#), which allows the value of this property to be one of the constants listed below or its numeric equivalent (in parentheses).

Syntax

fielddelimitervalue = [objectreference].FieldDelimiter

[objectreference].FieldDelimiter = fielddelimitervalue

Legal values

<u>Value</u>	<u>Effect</u>
\$LwpDelimiterTypeTabdelimited (180)	Indicates that a tab is being used to find sort fields in a document.
\$LwpDelimiterTypeTextdelimited (181)	Indicates that a string character is being used to find sort fields in a document.

Usage

Equivalent to choosing Text - Sort and selecting "Tab" or "Text" under "Field delimiter."

Word Pro: FieldManager property

{button ,AL('H_DOCINFO_CLASS',0)} [See list of classes](#)

(Read-only) The DocInfoFieldManager object for a document.

Data Type

[DocInfoFieldManager](#)

Syntax

fieldmanagervalue = [objectreference].FieldManager

Legal values

Always contains an instance of the DocInfoFieldManager class.

Usage

You can use this property to access any of the DocInfo fields in a document.

Word Pro: FieldNumber property

{button ,AL('H_SORTKEY_CLASS',0)} [See list of classes](#)

(Read-write) Specifies the field or column number on which you want to sort.

Data Type

Integer

Syntax

fieldnumbervalue = [objectreference].FieldNumber

[objectreference].FieldNumber = fieldnumbervalue

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Fields can be separated by tabs or a character that is not used elsewhere within the text. If the data to be sorted is in a table, the column numbers represent field numbers. For example, the first column within a table would represent field number one.

Equivalent to choosing Text - Sort and selecting a number from the "Field/col." box in either Level 1 "First sort by," Level 2 "Then by," or Level 3 "Then by:"

Word Pro: Fields property

{button ,AL('H_DOCINFOFIELDMANAGER_CLASS',0)} [See list of classes](#)

(Read-only) A collection of document fields.

Data Type

[DocInfoFieldCollection](#)

Syntax

fieldsvalue = [objectreference].Fields

Legal values

Always contains an instance of the DocInfoFieldCollection class.

Usage

In Word Pro, document fields are created in the Fields panel of the Document Properties dialog box.

Word Pro: FieldType property

{button ,AL(^H_CLICKHERE_CLASS;H_TEXTMARKER_CLASS;H_TEXT_CLASS',0)} [See list of classes](#)

(Read-only) The name of the Powerfield at the insertion point.

Data Type

Data type is String.

Syntax

fieldtypevariablevalue = [objectreference].FieldType

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Word Pro: FileFormat property

{button ,AL(^H_OLEOBJECT_CLASS',0)} [See list of classes](#)

(Read-only)

Note This property is not supported for the class OLEObject within OS/2.

Data Type

Data type is Long.

Syntax

fileformatvalue = [objectreference].FileFormat

Legal values**Usage**

Word Pro: Created event

{button ,AL('H_TEXTDOCUMENT_CLASS',0)} [See list of classes](#)

Arguments

Parameters

Created(String StyleSheet)

Usage

Word Pro: DocumentClosed event

{button ,AL(^H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

Raised by Word Pro each time Word Pro closes a document.

Arguments

Source

The object which raised this event. Use this argument to determine which object raised this event.

DocName

A String expression that specifies the name of the document which Word Pro closed.

Usage

You can write a script for this event which checks the name of the document and performs a set of actions after Word Pro has closed the document. This event is useful for cleaning up the application workspace.

Word Pro: DocumentClose event

{button ,AL(^H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

Raised by Word Pro each time Word Pro receives an instruction to close an open document.

Arguments

Source

The object which raised this event. Use this argument to determine which object raised this event.

DocName

A String expression that specifies the name of the document which Word Pro is instructed to close.

Usage

You can write a script for this event that checks the name of the document and performs a set of actions before allowing Word Pro to close the document. You can also use the object in the Source argument to access the WPAApplication object. This event is also useful for error-checking, prompting for saves, and cleaning up the application workspace.

Word Pro: DocumentCreated event

{button ,AL(^H_APPLICATION_CLASS;H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

Raised each time Word Pro completes the creation of a new Word Pro document.

Arguments

Source

The object which raised this event. Use this argument to determine which object raised this event.

StyleSheet

A String representing the name of the SmartMaster used in creating the new document.

Usage

Use this event to detect the completion of the creation of a new document in Word Pro. You can write a script for this event which checks the name of the SmartMaster and performs a set of actions after Word Pro has created the document. This event is also useful for error-checking, prompting for optional actions, and preparing the application workspace.

Word Pro: DocumentCreate event

{button ,AL(^H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

Raised by Word Pro each time Word Pro receives an instruction to create a document.

Arguments

Source

The object which raised this event. Use this argument to determine which object raised this event.

StyleSheet

A String expression that specifies the name of the SmartMaster which Word Pro is instructed to use in creating the new document.

Usage

You can write a script for this event which checks the name of the SmartMaster and performs a set of actions before allowing Word Pro to create the document. This event is also useful for error-checking, prompting for optional actions, and preparing the application workspace.

Word Pro: DocumentExported event

{button ,AL(^H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

Raised each time Word Pro exports a document. Equivalent to choosing File - Import/Export and exporting a document.

Arguments

Source

The object which raised this event. Use this argument to determine which object raised this event.

DocName

A String expression that specifies the name of the document which Word Pro exported.

DocType

A String expression that specifies the file type of the exported document.

Usage

Word Pro: DocumentExport event

{button ,AL(^H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

Raised each time Word Pro receives an instruction to export a document to a format other than Word Pro or Ami Pro.

Arguments

Source

The object which raised this event. Use this argument to determine which object raised this event.

DocName

A String expression that specifies the name of the document which Word Pro is instructed to export.

DocType

A String expression that specifies the file type which Word Pro will use for the exported document.

Usage

In Word Pro, this event is raised by choosing Import/Export from the File menu, then exporting to a file format other than Word Pro or Ami Pro.

Word Pro: DocumentImported event

{button ,AL(^H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

Raised each time Word Pro imports a document. Equivalent to choosing File - Import/Export and importing a document.

Arguments

Source

The object which raised this event. Use this argument to determine which object raised this event.

DocName

A String expression that specifies the name of the document which Word Pro imported.

DocType

A String expression that specifies the file type of the imported document.

Usage

Word Pro: DocumentImport event

{button ,AL(^H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

Raised each time Word Pro receives an instruction to import a document. Equivalent to choosing File - Import/Export and importing a document.

Arguments

Source

The object which raised this event. Use this argument to determine which object raised this event.

DocName

A String expression that specifies the name of the document which Word Pro is instructed to import.

DocType

A String expression that specifies the file type of the document which Word Pro is supposed to import.

Usage

Word Pro: DocumentInserted event

{button ,AL(^H_WPAPPLICATION_CLASS;H_TEXTDOCUMENT_CLASS',0)} [See list of classes](#)

Raised each time Word Pro inserts a document.

Arguments

Source

The object which raised this event. Use this argument to determine which object raised this event.

DocName

A String expression that specifies the name of the document which Word Pro has inserted.

Usage

Word Pro: DocumentInsert event

{button ,AL(^H_WPAPPLICATION_CLASS;H_TEXTDOCUMENT_CLASS',0)} [See list of classes](#)

Raised each time Word Pro receives an instruction to insert a document.

Arguments

Source

The object which raised this event. Use this argument to determine which object raised this event.

DocName

A String expression that specifies the name of the document which Word Pro is instructed to insert.

Usage

Word Pro: DocumentOpened event

{button ,AL(^H_APPLICATION_CLASS;H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

Raised each time Word Pro opens a document.

Arguments

Source

The object which raised this event. Use this argument to determine which object raised this event.

DocName

A String expression that specifies the name of the document which Word Pro opened.

Usage

Use this event to detect when Word Pro opens a document. You can use the DocName argument to check the name of the document Word Pro opened.

Word Pro: DocumentOpen event

{button ,AL(^H_APPLICATION_CLASS;H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

Raised each time Word Pro receives an instruction to open a document.

Arguments

Source

The object which raised this event. Use this argument to determine which object raised this event.

DocName

A String expression that specifies the name of the document which Word Pro is instructed to open.

Usage

Use this event to detect when Word Pro receives a command to open a document.

Word Pro: DocumentPrinted event

{button ,AL(^H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

Raised each time Word Pro prints a document.

Arguments*Source*

The object which raised this event. Use this argument to determine which object raised this event.

DocName

A String expression that specifies the name of the document which Word Pro printed.

Usage

Word Pro: DocumentPrint event

{button ,AL(^H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

Raised each time Word Pro receives an instruction to print a document.

Arguments

Source

The object which raised this event. Use this argument to determine which object raised this event.

DocName

A String expression that specifies the name of the document which Word Pro is instructed to print.

Usage

Word Pro: DocumentSaveAs event

{button ,AL(^H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

Raised each time Word Pro receives an instruction to save a document as another type of document or under another name. Equivalent to choosing File - Save As.

Arguments

Source

The object which raised this event. Use this argument to determine which object raised this event.

DocName

A String expression that specifies the name of the document which Word Pro is instructed to save as another type of document or under another name.

Usage

Word Pro: DocumentSavedAs event

{button ,AL(^H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

Raised each time Word Pro saves a document as another type of document or under another name. Equivalent to choosing File - Save As.

Arguments

Source

The object which raised this event. Use this argument to determine which object raised this event.

DocName

A String expression that specifies the name of the document which Word Pro saved in another file type or under another name.

Usage

Word Pro: DocumentSaved event

{button ,AL(^H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

Raised each time Word Pro saves a document. Equivalent to choosing File - Save.

Arguments

Source

The object which raised this event. Use this argument to determine which object raised this event.

DocName

A String expression that specifies the name of the document which Word Pro saved.

Usage

Word Pro: DocumentSave event

{button ,AL(^H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

Raised each time Word Pro receives an instruction to save a document. Equivalent to choosing File - Save.

Arguments

Source

The object which raised this event. Use this argument to determine which object raised this event.

DocName

A String expression that specifies the name of the document which Word Pro is instructed to save.

Usage

Word Pro: EnterClickHere event

{button ,AL(^H_WPAPPLICATION_CLASS;H_DIVISION_CLASS;H_CLICKHERE_CLASS;H_TEXTDOCUMENT_CLASS',0)} [See list of classes](#)

Raised each time the insertion point is moved from somewhere in the document into a ClickHere block.

Arguments*Source*

The object which raised this event. Use this argument to determine which object raised this event.

ClickHereName

A String expression which specifies the name of the ClickHere block that was entered.

Usage

Word Pro: EnterLayout event

```
{button ,AL(^H_WPAPPLICATION_CLASS;H_DIVISION_CLASS;H_LAYOUT_CLASS;H_CELLGROUPLAYOUT_CL  
ASS;H_CELLLAYOUT_CLASS;H_CONNECTEDLAYOUT_CLASS;H_COLUMNGROUPLAYOUT_CLASS;H_FOO  
TERLAYOUT_CLASS;H_FRAMELAYOUT_CLASS;H_FRAMEGROUPLAYOUT_CLASS;H_DROP_CAPLAYOUT_  
CLASS;H_GROUPLAYOUT_CLASS;H_HEADERLAYOUT_CLASS;H_NOTELAYOUT_CLASS;H_PAGELAYOUT_  
CLASS;H_ROWGROUPLAYOUT_CLASS;H_ROWLAYOUT_CLASS;H_RUBYLAYOUT_CLASS;H_SUPERTABLE  
GROUPLAYOUT_CLASS;H_SUPERTABLELAYOUT_CLASS;H_TABLELAYOUT_CLASS;H_ENDNOTELAYOUT_  
CLASS;H_FOOTNOTELAYOUT_CLASS;H_TABLEHEADINGLAYOUT_CLASS;H_TOCSUPERTABLELAYOUT_C  
LASS;H_TEXTDOCUMENT_CLASS',0)} See list of classes
```

Raised each time the insertion point is moved into a layout.

Arguments

Source

The object which raised this event. This could be WPAApplication or the layout itself. Use this argument to determine which object raised this event.

LayoutEntered

A String expression which specifies the name of the layout which was entered.

Usage

Each part of a document has its own layout. For example, a page, header, footer, and frame each has its own layout, even though they are all on the same page. When you move the insertion point from one of these layouts to another by clicking in the header, the footer, or some other part of the document, Word Pro raises an EnterLayout event. You can check the Source argument to determine which object is responsible for raising this event.

Word Pro: ExitClickHere event

{button ,AL(^H_WPAPPLICATION_CLASS;H_DIVISION_CLASS;H_CLICKHERE_CLASS;H_TEXTDOCUMENT_CLASS',0)} [See list of classes](#)

{button ,AL(^H_EXITCLICKHERE_EVENT_EXSCRIPT',1)} [See example](#)

Raised each time the insertion point is moved out of a ClickHere block.

Arguments

Source

The object which raised this event. Use this argument to determine which object raised this event.

ClickHereName

A String expression which specifies the name of the exited ClickHere block.

Usage

Word Pro: GotFocus event

{button ,AL(`H_TEXTDOCUMENT_CLASS',0)} [See list of classes](#)

Raised when the focus moves from one document window to another document window.

Arguments

TextDocument

The name of the document that is gaining the focus

Usage

When scripting with this element, do not use message boxes, dialog boxes and other types of interfaces that gain the focus after the script executes. Using these will cause an error in your script.

Word Pro: ImportInserted event

{button ,AL(^H_WPAPPLICATION_CLASS;H_TEXTDOCUMENT_CLASS',0)} [See list of classes](#)

Raised each time Word Pro imports a document and inserts the contents at the current insertion point. Equivalent to choosing File - Import/Export and specifying the option, "Import at the current insertion point."

Arguments

Source

The object which raised this event. Use this argument to determine which object raised this event.

DocName

A String expression which specifies the name of the document that Word Pro inserted.

DocType

A String expression which specifies the file type of the document that Word Pro inserted.

Usage

Word Pro: ImportInsert event

{button ,AL(^H_WPAPPLICATION_CLASS;H_TEXTDOCUMENT_CLASS',0)} [See list of classes](#)

Raised each time Word Pro receives an instruction to import a document and insert the contents at the current insertion point. Equivalent to choosing File - Import/Export and specifying the option, "Import at the current insertion point."

Arguments

Source

The object which raised this event. Use this argument to determine which object raised this event.

DocName

A String expression which specifies the name of the document that Word Pro has been instructed to import.

DocType

A String expression which specifies the file type of the document that Word Pro is supposed to import.

Usage

Word Pro: KeyStroke event

```
{button ,AL(^H_WPAPPLICATION_CLASS;H_DIVISION_CLASS;H_LAYOUT_CLASS;H_CELLGROUPLAYOUT_CL  
ASS;H_CELLLAYOUT_CLASS;H_CONNECTEDLAYOUT_CLASS;H_COLUMNGROUPLAYOUT_CLASS;H_FOO  
TERLAYOUT_CLASS;H_FRAMELAYOUT_CLASS;H_FRAMEGROUPLAYOUT_CLASS;H_DROP_CAPLAYOUT_  
CLASS;H_GROUPLAYOUT_CLASS;H_HEADERLAYOUT_CLASS;H_NOTELAYOUT_CLASS;H_PAGELAYOUT_  
CLASS;H_ROWGROUPLAYOUT_CLASS;H_ROWLAYOUT_CLASS;H_RUBYLAYOUT_CLASS;H_SUPERTABLE  
GROUPLAYOUT_CLASS;H_SUPERTABLELAYOUT_CLASS;H_TABLELAYOUT_CLASS;H_ENDNOTELAYOUT_  
CLASS;H_FOOTNOTELAYOUT_CLASS;H_TABLEHEADINGLAYOUT_CLASS;H_TOCSUPERTABLELAYOUT_C  
LASS;H_TEXTDOCUMENT_CLASS',0)} See list of classes
```

Raised each time a user presses and releases a key.

Arguments

Source

The object which raised this event. Use this argument to determine which object raised this event. You can use Source to retrieve information about or change the properties of the object in which the keystroke occurred. When you write a script for a keystroke event in a Layout object, this argument returns the Layout object in which the keystroke event was raised. When you write a script for a keystroke event in the WPAApplication object, this argument returns the WPAApplication object.

Key

An Integer which specifies the ASCII value of the key that was pressed. This code can be translated into a character, using the Chr function.

Modifier

An Integer which specifies the ASCII value of the modifier key that was pressed.

ReceivingLayout

A String expression representing the name of the layout object in which the Keystroke event was raised.

Usage

Word Pro can detect a Keystroke event in four different types of objects: WPAApplication, TextDocument, Division, and Layout. This means that you can write a script that runs every time a keystroke is detected anywhere within Word Pro (WPAApplication), anywhere within a particular document (TextDocument), anywhere within a particular division (Division), or anywhere within a particular layout (Layout). By placing your script in the Keystroke event section for a particular object, you allow the script to run only when a Keystroke event is detected within that object. You can also use the End statement to limit the number of events that Word Pro raises for a single keystroke.

Word Pro: LostFocus event

{button ,AL(`H_TEXTDOCUMENT_CLASS',0)} [See list of classes](#)

Raised when the focus moves from one document window to another document window.

Arguments

TextDocument

The name of the document that is losing the focus

Usage

When scripting with this element, do not use message boxes, dialog boxes and other types of interfaces that gain the focus after the script executes. Using these will cause an error in your script.

Word Pro: MouseDown event

```
{button ,AL(^H_WPAPPLICATION_CLASS;H_DIVISION_CLASS;H_LAYOUT_CLASS;H_CELLGROUPLAYOUT_CL  
ASS;H_CELLLAYOUT_CLASS;H_CONNECTEDLAYOUT_CLASS;H_COLUMNGROUPLAYOUT_CLASS;H_FOO  
TERLAYOUT_CLASS;H_FRAMELAYOUT_CLASS;H_FRAMEGROUPLAYOUT_CLASS;H_DROP_CAPLAYOUT_  
CLASS;H_GROUPLAYOUT_CLASS;H_HEADERLAYOUT_CLASS;H_NOTELAYOUT_CLASS;H_PAGELAYOUT_  
CLASS;H_ROWGROUPLAYOUT_CLASS;H_ROWLAYOUT_CLASS;H_RUBYLAYOUT_CLASS;H_SUPERTABLE  
GROUPLAYOUT_CLASS;H_SUPERTABLELAYOUT_CLASS;H_TABLELAYOUT_CLASS;H_ENDNOTELAYOUT_  
CLASS;H_FOOTNOTELAYOUT_CLASS;H_TABLEHEADINGLAYOUT_CLASS;H_TOCSUPERTABLELAYOUT_C  
LASS;H_TEXTDOCUMENT_CLASS',0)} See list of classes
```

Raised each time the user presses down on a mouse button.

Arguments

Source

The object which raised this event. Use this argument to determine which object raised this event.

GeneralModifier

The key or keys which were pressed when the mouse button was pressed down. Data type is Variant, which allows the value of the argument to be one of the constants listed below or its hexadecimal equivalent (in parentheses). You can combine these values when you want Word Pro to combine the features listed below. Use the OR operator to combine values.

<u>Value</u>	<u>Effect</u>
LwpGeneralModifierNone (&H0)	No keys were down. The SHIFT key was down.
LwpGeneralModifierShift (&H1)	The ALT key was down.
LwpGeneralModifierAlt (&H2)	The CTRL key was down.
LwpGeneralModifierCtrl (&H4)	The COMMAND key was down.
LwpGeneralModifierCommand (&H8)	The OPTION key was down.
LwpGeneralModifierOption (&H10)	The SYSRQ key was down.
LwpGeneralModifierSys (&H20)	The CAPS LOCK key was down.
LwpGeneralModifierCapslock (&H40)	The NUM LOCK key was down.
LwpGeneralModifierNumlock (&H80)	The SCROLL LOCK key was down.
LwpGeneralModifierScrolllock (&H100)	The HELP key was down.
LwpGeneralModifierHelp (&H200)	

SpecificModifier

Specifies which mouse button was pressed. Data type is Variant, which allows the value of the argument to be one of the constants listed below or its hexadecimal equivalent (in parentheses). You can combine these values when you want Word Pro to combine the features listed below. Use the OR operator to combine values.

LwpSpecificModifierNone (&H0)

LwpSpecificModifierMbutton1 (&H1)

LwpSpecificModifierMbutton2 (&H2)

LwpSpecificModifierMbutton3 (&H4)

LwpSpecificModifierMbuttonmask (&H7)

LwpSpecificModifierMbuttondown (&H8)

LwpSpecificModifierMbuttonup (&H10)

LwpSpecificModifierDoublebyte (&H1)

LwpSpecificModifierGoingdown (&H2)

LwpSpecificModifierComingup (&H4)

LwpSpecificModifierVirtual (&H8)

LwpSpecificModifierClicked (&H2)

LwpSpecificModifierNomove (&H4)

LwpSpecificModifierNovolatile (&H8)

LwpSpecificModifierReset (&H10)

LwpPopcheckpoint (&H20)

ReceivingLayout

A String expression which specifies the name of the layout over which the mouse button was pressed.

Usage

Word Pro: MouseUp event

```
{button ,AL(^H_WPAPPLICATION_CLASS;H_DIVISION_CLASS;H_LAYOUT_CLASS;H_CELLGROUPLAYOUT_CL  
ASS;H_CELLLAYOUT_CLASS;H_CONNECTEDLAYOUT_CLASS;H_COLUMNGROUPLAYOUT_CLASS;H_FOO  
TERLAYOUT_CLASS;H_FRAMELAYOUT_CLASS;H_FRAMEGROUPLAYOUT_CLASS;H_DROP CAPLAYOUT_  
CLASS;H_GROUPLAYOUT_CLASS;H_HEADERLAYOUT_CLASS;H_NOTELAYOUT_CLASS;H_PAGELAYOUT_  
CLASS;H_ROWGROUPLAYOUT_CLASS;H_ROWLAYOUT_CLASS;H_RUBYLAYOUT_CLASS;H_SUPERTABLE  
GROUPLAYOUT_CLASS;H_SUPERTABLELAYOUT_CLASS;H_TABLELAYOUT_CLASS;H_ENDNOTELAYOUT_  
CLASS;H_FOOTNOTELAYOUT_CLASS;H_TABLEHEADINGLAYOUT_CLASS;H_TOCSUPERTABLELAYOUT_C  
LASS;H_TEXTDOCUMENT_CLASS',0)} See list of classes
```

Raised each time the user releases a mouse button after pressing it.

Arguments

Source

The object which raised this event. Use this argument to determine which object raised this event.

GeneralModifier

The key or keys pressed when the mouse button was released. Data type is Variant ,which allows the value of the argument to be one of the constants listed below or its hexadecimal equivalent (in parentheses). You can combine these values when you want Word Pro to combine the features listed below. Use the OR operator to combine values.

<u>Value</u>	<u>Effect</u>
LwpGeneralModifierNone (&H0)	No keys were down. The SHIFT key was down.
LwpGeneralModifierShift (&H1)	The ALT key was down.
LwpGeneralModifierAlt (&H2)	The CTRL key was down.
LwpGeneralModifierCtrl (&H4)	The COMMAND key was down.
LwpGeneralModifierCommand (&H8)	The OPTION key was down.
LwpGeneralModifierOption (&H10)	The SYSRQ key was down.
LwpGeneralModifierSys (&H20)	The CAPS LOCK key was down.
LwpGeneralModifierCapslock (&H40)	The NUM LOCK key was down.
LwpGeneralModifierNumlock (&H80)	The SCROLL LOCK key was down.
LwpGeneralModifierScrolllock (&H100)	The HELP key was down.
LwpGeneralModifierHelp (&H200)	

SpecificModifier

Specifies which mouse button was released. Data type is Variant, which allows the value of the argument to be one of the constants listed below or its hexadecimal equivalent (in parentheses). You can combine these values when you want Word Pro to combine the features listed below. Use the OR operator to combine values.

LwpSpecificModifierNone (&H0)

LwpSpecificModifierMbutton1 (&H1)

LwpSpecificModifierMbutton2 (&H2)

LwpSpecificModifierMbutton3 (&H4)

LwpSpecificModifierMbuttonmask (&H7)

LwpSpecificModifierMbuttondown (&H8)

LwpSpecificModifierMbuttonup (&H10)

LwpSpecificModifierDoublebyte (&H1)

LwpSpecificModifierGoingdown (&H2)

LwpSpecificModifierComingup (&H4)

LwpSpecificModifierVirtual (&H8)

LwpSpecificModifierClicked (&H2)

LwpSpecificModifierNomove (&H4)

LwpSpecificModifierNovolatile (&H8)

LwpSpecificModifierReset (&H10)

LwpPopcheckpoint (&H20)

ReceivingLayout

A String expression which specifies the name of the layout over which the mouse button was released.

Usage

Word Pro: Moved event

{button ,AL(^H_WINDOW_CLASS;H_APPLICATIONWINDOW_CLASS;H_STATUSBAR_CLASS;H_DOCWINDOW_CLASS;H_DOCWINDOW_CLASS',0)} [See list of classes](#)

Word Pro raises this event when the containing object is moved.

Arguments

Source

Contains the object which was moved.

Usage

Word Pro: Opened event

{button ,AL('H_TEXTDOCUMENT_CLASS',0)} [See list of classes](#)

Arguments

Parameters

Opened(String DocName)

Usage

Word Pro: PreClose event

{button ,AL(^H_DOCUMENT_CLASS;H_TEXTDOCUMENT_CLASS',0)} [See list of classes](#)

Arguments

Parameters

PreClose(String DocName)

Usage

Word Pro: PrePrint event

{button ,AL('H_TEXTDOCUMENT_CLASS',0)} [See list of classes](#)

Arguments

Parameters

PrePrint(String DocName)

Usage

Word Pro: Printed event

{button ,AL('H_TEXTDOCUMENT_CLASS',0)} [See list of classes](#)

Arguments

Parameters

Printed(String DocName)

Usage

Word Pro: Quit event

{button ,AL(^H_APPLICATION_CLASS;H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

Raised each time Word Pro receives an instruction to exit the Word Pro application. The event handler in the Script Editor processes this event and any related scripts before Word Pro closes.

Arguments*Source*

The object which raised this event. Use this argument to determine which object raised this event.

Usage

Use this event to run a script any time you exit Word Pro. The document that contains the Quit event script must be open when you exit Word Pro.

Word Pro: SaveAs event

{button ,AL(^H_DOCUMENT_CLASS;H_TEXTDOCUMENT_CLASS',0)} [See list of classes](#)

Arguments

Parameters

SaveAs(String DocName)

Usage

Word Pro: SavedAs event

{button ,AL(^H_DOCUMENT_CLASS;H_TEXTDOCUMENT_CLASS',0)} [See list of classes](#)

Arguments

Parameters

SavedAs(String DocName)

Usage

Word Pro: Saved event

{button ,AL('H_DOCUMENT_CLASS;H_TEXTDOCUMENT_CLASS',0)} [See list of classes](#)

Arguments

Parameters

Saved(String DocName)

DocName

A String representing the name of the saved document.

Usage

Word Pro: Save event

{button ,AL('H_DOCUMENT_CLASS;H_TEXTDOCUMENT_CLASS',0)} [See list of classes](#)

Arguments

Parameters

Save(String DocName)

Usage

Word Pro: StatusBarButtonClicked event

{button ,AL(^H_STATUSBAR_CLASS;H_STATUSBARBUTTON_CLASS',0)} [See list of classes](#)

This event is emitted when you click on a button, or when following the SimulateButtonClick method..

If you choose the StatusBar object in the LotusScript window, you can't see the emitted events assigned to it. If you choose one event and write a routine within it, and, if any value other than False (0) is returned, that event is handled by LotusScript, not Word Pro.

Arguments

ButtonName = name of button that is clicked.

Parameters

StatusBarButtonClicked(String ButtonName)

Usage

This event applies only to clickable buttons. Popup type buttons receive the StatusBarButtonFillPopupList event instead.

In the script for this event, write the code that you want executed in response to the button click. You can determine which button was clicked by looking at the ButtonName parameter.

Word Pro: StatusBarButtonFillPopupList event

{button ,AL('H_STATUSBAR_CLASS;H_STATUSBARBUTTON_CLASS',0)} [See list of classes](#)

This event is emitted when you click a popup button. You can write instructions in this event to fill up the popup list.

If you choose the StatusBar object in the LotusScript window, you can see the emitted events assigned to it. If you choose one event and write a routine within it, and, if any value other than False (0) is returned, that event is handled by LotusScript, not Word Pro.

Arguments

Parameters

StatusBarButtonFillPopupList(String ButtonName)

Usage

The list is always empty when this event is emitted. There are several methods that you can call when responding to this event:

<u>Method</u>	<u>Effect</u>
AddPopupGraphicItem	Only used for graphic buttons. Can be called multiple times, once for each graphic in the list.
AddPopupTextItem	Only used for text popup buttons. Can be called multiple times, once for each string in the list.
SetPopupAlignment	Lets you align the button's contents.
SetPopupIndex	Lets you select which item in the button's list will be highlighted by default.
SetPopupWidth	Lets you set a width for the popup list.
SetPopupWidthType	Lets you specify how the width of the popup list will be calculated.

If you want Word Pro to do its normal processing for a button property, this event should return False. To prevent Word Pro from doing its normal processing, this event should return True.

Word Pro: StatusBarButtonItemSelected event

{button ,AL(^H_STATUSBARBUTTON_CLASS',0)} [See list of classes](#)

This event is emitted when you select an item from a popup list in the button.

If you choose the StatusBar object in the LotusScript window, you can't see the emitted events assigned to it. If you choose one event and write a routine within it, and, if any value other than False (0) is returned, that event is handled by LotusScript, not Word Pro.

Arguments

Button name - Name of the button selected.

Index (0 based) - Indicates the item number selected.

String - The text of the selected item.

Parameters

StatusBarButtonItemSelected(String ButtonName, Integer Index, String SelectedItem)

Usage

This event applies only to clickable buttons. Popup type buttons receive the StatusBarButtonFillPopupList event instead.

In the script for this event, write the code that you want executed in response to the button click. You can determine which button was clicked by looking at the ButtonName parameter.

Word Pro: StatusBarButtonOverrideGraphic event

{button ,AL(^H_STATUSBARBUTTON_CLASS',0)} [See list of classes](#)

This event is emitted when you refresh a graphic button.

If you choose the StatusBar object in the LotusScript window, you can see the emitted events assigned to it. If you choose one event and write a routine within it, and, if any value other than False (0) is returned, that event is handled by LotusScript, not Word Pro.

Arguments

Parameters

StatusBarButtonOverrideGraphic(String ButtonName)

Usage

You must provide the graphic in this event; otherwise, the button appears blank. The SetOverrideGraphic method must be called.

Word Pro: StatusBarButtonOverrideTextAndGraphic event

{button ,AL(^H_STATUSBARBUTTON_CLASS',0)} [See list of classes](#)

This event is emitted when you update a text and graphic button.

If you choose the StatusBar object in the LotusScript window, you can see the emitted events assigned to it. If you choose one event and write a routine within it, and, if any value other than False (0) is returned, that event is handled by LotusScript, not by Word Pro.

Arguments

Parameters

StatusBarButtonOverrideTextAndGraphic(String ButtonName)

Usage

You must provide the graphic and the text in this event; otherwise, the button appears blank. The SetOverrideGraphic and/or the SetOverrideText method must be called.

Note If the text on the status bar button is never going to change, you can use the LwpButtonNoTextFromHost (&H800) parameter when the button is created.

Word Pro: StatusBarButtonOverrideText event

{button ,AL(^H_STATUSBAR_CLASS;H_STATUSBARBUTTON_CLASS',0)} [See list of classes](#)

This event is emitted when you update a text button.

If you choose the StatusBar object in the LotusScript window, you can see the emitted events assigned to it. If you choose one event and write a routine within it, and, if any value other than False (0) is returned, that event is handled by LotusScript, not Word Pro.

Arguments

Parameters

StatusBarButtonOverrideText(StatusBarButton Source, String ButtonName)

Usage

You must provide the text in this event; otherwise, the button appears blank. The SetOverrideText method must be called.

Note If the text on the status bar button is never going to change, you can use the LwpButtonNoTextFromHost (&H800) parameter when the button is created.

Word Pro: TimerTick event

{button ,AL(^H_LWPTIMER_CLASS',0)} [See list of classes](#)

Arguments

Parameters

TimerTick()

Usage

Word Pro: WMCommand event

{button ,AL(^H_WPAPPLICATION_CLASS;H_TEXTDOCUMENT_CLASS',0)} [See list of classes](#)

Raised each time a WMCommand is issued in Word Pro. Occurs when a user chooses a menu or menu item, and when a script calls the WMCommand method.

Arguments*Source*

The object that raised this event. Use this argument to determine which object raised this event.

cmd

An Integer expression which specifies which menu item or WMCommand was chosen or called. You can find the WMCommands and their corresponding integers in the file named WPBITMSK.LSS, which came with Word Pro.

Usage

'Example: DefaultFileType property

'This example sets the default Save As type for a SmartMaster as Lotus Ami Pro.

'The SaveAs method is then called to save the document as a SmartMaster

```
.ActiveDocWindow.Document.DefaultFileType = "Lotus Ami Pro"
```

```
.SaveAs "C:\lotus\Smasters\wordpro\AmiPro.mwp", "", "Lotus Word Pro SmartMaster",  
False, True, False
```

'Example: DefaultMenu property

' This example sets the default menu set to the Ami Pro menu set, then

' resets the default menu to the Word Pro default.

```
.Application.Preferences.DefaultMenu = "Lotus Ami Pro"
```

'now reset this to the Word Pro standard menu by setting the property to blank

```
.Application.Preferences.DefaultMenu = ""
```

```
'Example: DefColWidth property
' This example creates a table, then prints the default column width of the
' table to the LotusScript output panel.
.CreateTable
TwipColWid = .BaseTable.DefColWidth
Print "The default column width is " & TwipColWid/1440 & " inches."
```

```

'Example: Definition property
'This example determines the hierarchy of the current paragraph style's
'attributes attribute.  If attributes are inherited, the user is given a
'chance to make it local.

If .Text.ParagraphStyle.AttrStyleName = "" Then 'this style's attributes are local
    MsgBox .Text.ParagraphStyle.Name & " stores its attributes locally.", MB_OK,
    "Example Script"
Else ' Give user a chance to make it local
    Stat = MsgBox (.Text.ParagraphStyle.Name & " inherits its attributes from "
    & .Text.ParagraphStyle.AttrStyleName & ". Do you want to make it local?", 36, "Example
    Script")
    If stat = 6 Then 'user said yes
        'get the current hierarchy, and add attributes to the local attrs.
        StyleAttrs = .Text.ParagraphStyle.Definition + &H4
        .SetStyle $LwpStyleTypeParagraph, .Text.ParagraphStyle.Name, StyleAttrs
        ' set the attribute style name to the empty string.
        .Text.ParagraphStyle.AttrStyleName = ""
    End If
End If

```

```
'Example: DeleteButton method
' This example removes all custom statusbar buttons and repaints the
' statusbar.
' RUNTIME DEPENDENCIES: You must have a document open for this script to work.
```

```
Dim StatBar As StatusBar
Set StatBar = .ApplicationWindow.StatusBar
Forall Button In StatBar.StatusBarButtons
    If Button.GetButtonType = 153 Then
        Button.DeleteButton()
    End If
End Forall
StatBar.InvalidateWholeBar
```

```
'Example: DeleteChars method
' This example inserts two words into the current document. After the
' message box is closed, the first five characters are deleted.
' RUNTIME DEPENDENCIES: You must have a document open for this script to work.

.Text.InsertText "Some text."
  MessageBox "Click OK to delete the first 5 characters.",MB_OK,"Example Script"
.Text.Backward $LwpNavigateObjectTypeWord, 3
.Text.DeleteChars 5
```


'Example: DeleteClickHere method
'This example inserts a Click Here block into the current document,
'then deletes the block..

```
MyClickHere = .InsertClickHere()  
.Division.Foundry.ClickHeres(MyClickHere).Prompt.Clear  
.Division.Foundry.ClickHeres(MyClickHere).Prompt.InsertText "Click here to type Text",  
False, $LwpTextTypeNative  
.Division.Foundry.ClickHeres(MyClickHere).MirrorName = ""  
.Division.Foundry.ClickHeres(MyClickHere).Action = 1  
.Division.Foundry.ClickHeres(MyClickHere).HelpText = "Click here to type Text"  
.Division.Foundry.ClickHeres(MyClickHere).UsesHelp = False  
.Division.Foundry.ClickHeres(MyClickHere).TabOrder = 1  
.Division.Foundry.ClickHeres(MyClickHere).TabExits = True  
.Division.Foundry.ClickHeres(MyClickHere).ReturnExits = False  
.Division.Foundry.ClickHeres(MyClickHere).Name = MyClickHere  
.Division.Foundry.ClickHeres(MyClickHere).AllowListEdit = True  
.Division.Foundry.ClickHeres(MyClickHere).AllowListMultiValues = False  
.Division.Foundry.ClickHeres(MyClickHere).RemoveNamedProperty  
.Division.Foundry.ClickHeres(MyClickHere).SetNamedProperty "Collect", "Off"  
.Division.Foundry.ClickHeres(MyClickHere).SetNamedProperty "Required", "Off"  
.Division.Foundry.ClickHeres(MyClickHere).SetNamedProperty "Notes/FX", "Off"  
.UpdateUI  
'Move into the click here we just created, then delete it.  
MessageBox "Click OK to delete the Click Here block.", MB_OK, "Example Script"  
.Type "[Left]"  
.DeleteClickHere
```

```
'Example: DeleteContainer method
'This example imports a graphic into the current document, then deletes the
'graphic, and ultimately the frame.
GraphicFileName = .ApplicationWindow.UserInterfacePrefs.GraphicPath & "\_Wpdon.gif"
.ImportGraphic GraphicFileName, "", False, False
MessageBox "Click OK to delete the graphic.", MB_OK, "Example Script"
.Graphic.DeleteContent
MessageBox "Click OK to delete the frame.", MB_OK, "Example Script"
.Frame.DeleteContainer
```

'Example: DeleteContents method

' This example deletes the contents of a Click Here Block.

' RUNTIME DEPENDENCIES: You must have a document open and a click here

' named 'ClickHere1' for this script to work.

.Division.Foundry.ClickHeres.Item("ClickHere1").DeleteContents

```
'Example: DeleteContent method
'This example imports a graphic into the current document, then deletes the
'graphic, and ultimately the frame.
GraphicFileName = .ApplicationWindow.UserInterfacePrefs.GraphicPath & "\_Wpdon.gif"
.ImportGraphic GraphicFileName, "", False, False
MessageBox "Click OK to delete the graphic.", MB_OK, "Example Script"
.Graphic.DeleteContent
MessageBox "Click OK to delete the frame.", MB_OK, "Example Script"
.Frame.DeleteContainer
```

```
'Example: DeleteDivision method
' This example creates a new division based on the "DEFAULT.MWP" SmartMaster.
' It is placed after the current division.
' After the division is created it is deleted.
' RUNTIME DEPENDENCIES: You must have a document open for this script to work.
```

```
Dim SmartMaster as String
Dim NewDivId as String
```

```
SmartMaster = .ApplicationWindow.UserInterfacePrefs.StylePath & "\DEFAULT.MWP"
.CreateDivision SmartMaster, "", $LWPDivLocInsertAfterCurrentDiv, "", ""
```

```
' Get the hexadecimal name of the division so we know which one to delete.
NewDivId = .Division.Name
MessageBox "Click OK to delete the division", MB_OK, "Example Script"
.DeleteDivision(NewDivId)
```

```
'Example: DeleteField method
' This example adds a new field named 'ExampleField' for the current document.
' The field's contents are inserted and the field is then deleted.
' RUNTIME DEPENDENCIES: You must have a document open for this script to work.

.ActiveDocument.DocInfo.FieldManager.AddField "ExampleField", "Some data for
ExampleField ", 1
.InsertDocInfo $LwpDocVarField, "ExampleField"
MessageBox "Click OK to delete the field.", MB_OK, "Example Script"
.ActiveDocument.DocInfo.FieldManager.DeleteField "ExampleField"
```

'Example: DeleteFont property

'This example script sets the revision deletion font properties for the current editor.

' change the font color to red

.ActiveDocument.EditorManager.CurrentEditor.**DeleteFont**.FontColor.RevertToStyle

.ActiveDocument.EditorManager.CurrentEditor.**DeleteFont**.FontColor.Red = 255

.ActiveDocument.EditorManager.CurrentEditor.**DeleteFont**.FontColor.Blue = 0

.ActiveDocument.EditorManager.CurrentEditor.**DeleteFont**.FontColor.Green = 0

.ActiveDocument.EditorManager.CurrentEditor.**DeleteFont**.FontColor.Override =
\$LwpColorOverrideRgb

'Set up overstrike, with the hyphen character

.ActiveDocument.EditorManager.CurrentEditor.**DeleteFont**.RevertToStyle
\$LwpFontPropertyStrikethru

.ActiveDocument.EditorManager.CurrentEditor.**DeleteFont**.RevertToStyle
\$LwpFontPropertyOverstrike

.ActiveDocument.EditorManager.CurrentEditor.**DeleteFont**.OverstrikeCharacter = 45

```
'Example: DeleteItem method
' This example creates a new menu item named 'Example Menu' on the File
' menu. The mnuMenuSub subroutine is assigned to run each time the new
' menu item is selected
' RUNTIME DEPENDENCIES: You must have a document open for this script to work.

Dim NewMenu As MenuItem
Dim MenuName as String
Dim MenuSpacer as String

MenuName = "&Example Menu"
MenuSpacer = Chr$(8)

' Set menu object
Set NewMenu = .ApplicationWindow.LwpMenuBar.Items.Item("&File")

' Create a new menu off of the File Menu and before the Save option
' Delete it first to prevent duplicates
NewMenu.DeleteItem MenuName
NewMenu.NewItem MenuName,"!mnuMenuSub",0,"&Save" & MenuSpacer & "Ctrl+S"

Sub mnuMenuSub
' This sub will run whenever the user clicks on the new menu item.
    MessageBox "Example Menu was called.", MB_OK, "Example Script"
End Sub
```



```
'Example: DeleteKey method
' This example positions the insertion point at the beginning of the current
' paragraph. Some text is inserted and then each character is deleted.
' RUNTIME DEPENDENCIES: You must have a document open for this script to work.
```

```
Dim Message As String
Dim LenMessage As Integer
Dim x As Integer

Message = "This is some text"
LenMessage = Len(Message)
.Text.MoveToStart $LwpLocationTypeParagraph
.Type Message
.Text.MoveToStart $LwpLocationTypeParagraph
For x = 1 To LenMessage
    .DeleteKey
Next
```

'Example: DeleteLayout method

'This example creates a table, and changes the background color of cell A1.

'The layout of cell A1 is then deleted, which removes the background color.

```
.CreateTable False, "Default Table", 2, 2
```

```
.Table.TableFill.Background.Pattern = $LtsFillSolid
```

```
.Table.TableFill.Background.Color.Red = 255
```

```
.Table.TableFill.Background.Color.Blue = 0
```

```
.Table.TableFill.Background.Color.Green = 0
```

```
.Table.TableFill.Background.Color.Override = $LwpColorOverrideRgb
```

```
.Table.TableFill.Background.BackColor.Red = 0
```

```
.Table.TableFill.Background.BackColor.Blue = 18
```

```
.Table.TableFill.Background.BackColor.Green = 232
```

```
.Table.TableFill.Background.BackColor.Override = $LwpColorOverrideBlack
```

```
.Table.TableFill.FillStyle = $LwpTableFillStyleAll
```

```
MessageBox "Click OK to delete this cell layout.", MB_OK, "Example Script"
```

```
.Table.CellLayout.DeleteLayout
```

'Example: DeleteMarker method
'This example inserts some text into the current document, and creates
'a default marker. The marker is then extended to the end of the paragraph.
'The contents of the marker is displayed in a message box.
'Finally the marker is deleted.

```
.Text.InsertText "This is an example of markers. "  
.Text.Backward $LwpNavigateObjectTypeParagraph, 1  
MyMarker = .Text.Mark($LwpMarkerTypeDefault)  
.Text.Forward $LwpNavigateObjectTypeParagraph, 1  
.Text.Mark $LwpMarkerTypeDefault, MyMarker, $LwpRangePartEnd  
MyText = .Foundry.Markers(MyMarker).GetMarkedText  
MessageBox MyText, MB_OK, "Example Script"  
.Division.Foundry.Markers(MyMarker).DeleteMarker
```

```
'Example: DeleteParallelColumns method
' This example creates then deletes a parallel column for the current
' document.
' RUNTIME DEPENDENCIES: You must have a document open for this script to work.

.CreateParallelColumns 3, $LtsAlignmentHorizCenter
MessageBox "Click OK to delete the columns.", MB_OK, "Example Script"
.DeleteParallelColumns
```

```
'Example: DeleteSection method
' This example inserts several sections in the active division and then
' deletes all sections.
' RUNTIME DEPENDENCIES: You must have a document open for this script to work.

.InsertSection "Default Page", True, True, $LwpStartTypeNextpage, False, True
.InsertSection "Default Page", True, True, $LwpStartTypeNextpage, False, True
MessageBox "Click OK to delete all the sections.", MB_OK, "Example Script"
Forall Section In .Division.Foundry.Sections
    Section.DeleteSection
End Forall
```

```
'Example: DeleteSmartCorrect method  
' This example deletes an entry in the SmartCorrect list.  
' RUNTIME DEPENDENCIES: You must have a document open for this script to work.
```

```
Language = "English (United States)"  
.Application.SmartCorrects(Language).DeleteSmartCorrect "lts"
```

```
'Example: DeleteTable method
' This example creates a table with 4 rows and 5 columns, fills the table with data,
' then deletes portions of the table when you click OK in the message box.
' RUNTIME DEPENDENCIES: You must have a document open for this script to work.

.CreateTable False, "Default Table", 5, 4
For i = 0 To 3
    For j = 0 To 4
        .Table.CellLayout(i,j).GoToLayout
        .Type "Column " & j+1 & " Row " & i+1
    Next
Next

Messagebox "Click OK to delete the current row.",MB_OK,"Example Script"
.Table.DeleteTable $LwpTableDelTypeRow

Messagebox "Click OK to delete the second column.",MB_OK,"Example Script"
.Table.DeleteTable $LwpTableDelTypeColumn, 1, 1

Messagebox "Click OK to delete the current table.",MB_OK,"Example Script"
.DeleteTable
```

'Example: DeleteTab method

'This example inserts tabs at 2, 3, and 4 inches from the left edge of
'the page. The middle tab is then deleted.

```
.ActiveDocWindow.WinViewPrefs.IsViewHorzRuler = True
```

```
.Text.TabRack.InsertOne 2880, $LwpTabTypeLeft, $LwpTabLeaderNone, $LwpTabRelativeLeft,  
0
```

```
.Text.TabRack.InsertOne 4320, $LwpTabTypeLeft, $LwpTabLeaderNone, $LwpTabRelativeLeft,  
0
```

```
.Text.TabRack.InsertOne 5760, $LwpTabTypeLeft, $LwpTabLeaderNone, $LwpTabRelativeLeft,  
0
```

'Argument is 0 based, so use 1 to delete 2nd tab

```
.Text.TabRack.DeleteTab 1
```



```
'Example: DeleteVersion method
' This example creates a version for the current document then deletes the
' version.
' RUNTIME DEPENDENCIES: You must have a document open for this script to work.

.Division.VersionManager.CreateVersion "NewVersion"
MessageBox "Click OK to delete the version.", MB_OK, "Example Script"
Forall Version In .ActiveDocument.VersionManager.Versions
    If Version.name = "NewVersion" Then
        .ActiveDocument.VersionManager.DeleteVersion Version.DocVersionId
    End If
End Forall
```

```
'Example: DemoteOutlineLevel method
'This example inserts numbered lines of text at different outline levels.

.Text.Bullet.Name = "Default Outline"
.Text.Numbering.Position = 3
.Text.Indent.First = 360
.Text.Indent.Rest = 360
.Type "This is outline level 1[Enter]"
.DemoteOutlineLevel
.Type "This is outline level 2[Enter]"
.DemoteOutlineLevel
.Type "This is outline level 3[Enter]"
.PromoteOutlineLevel
.Type "Back to outline level 2[Enter]"
```

```
'Example: Description property
'This example adds a document description if one has not been created,
'then prints the description to the LotusScript output panel.
If .ActiveDocument.Description = "" Then
    ActiveDocument.Description = "This is the description for this document"
End If
Print .ActiveDocument.Description
```

```
'Example: Deselect method
' This example inserts some text into the current document which is then
' selected. After the message box is closed, the text is deselected.
' RUNTIME DEPENDENCIES: You must have a document open for this script to work.

.Text.InsertText "This is some text."
MessageBox "Click OK to select the text.",MB_OK,"Example Script"
.Select $LwpSelectObjectTypeParagraph
MessageBox "Click OK to deselect the text.",MB_OK,"Example Script"
.Text.Deselect
```

'Example: DestroyDocWindow method

'This example uses the DestroyDocWindow method to remove a newly created
'document from the screen.

.NewDocument

.Type "We will not be given a chance to save this text."

.ActiveDocWindow.**DestroyDocWindow**

'Example: DirectionDown property
'This example creates a table and disables the auto row height property,
'then creates a frame, and sizes it to fit the page height.

.CreateTable

.Table.CurrentRow.**DirectionDown** = &H0 ' disable auto row height

.Table.CurrentRow.Height = 288 ' and set a fixed height.

.NewFrame 959, 666, 1693, 6589, "Default Frame" ' create the frame

.Frame.Anchor \$LwpAnchorWhereLayout, \$LwpConditionTypeOnlyspecificpage,
\$LwpRelativeTypeLytParent 'position 'on current page'

.QuickAlignFrame \$LwpQuickLayoutAlignTop ' align with top of page

.Frame.Layout.DirectionUp = &H0

.Frame.Layout.**DirectionDown** = &H5 ' enable auto sizing downward

```
'Example: DisableClickHeres property
'This example asks the user whether Click Here blocks should be edited on screen'
'or whether they should be filled in, and then sets the appropriate option.

stat = MessageBox ("Do you want to edit Click Here blocks?", 36, "Example Script")
If stat = 6 Then ' user said yes
    .ActiveDocument.DocControl.DisableClickHeres = True
Else
    .ActiveDocument.DocControl.DisableClickHeres = False
End If
```

```
'Example: DisableConsistencyCheck property
'This example asks the user whether to disable checking for errors
'when opening documents, and then sets the appropriate option.

stat = MessageBox ("Do you want to disable checking document consistency?", 36,
"Example Script")
If stat = 6 Then ' user said yes
    .ApplicationWindow.UserInterfacePrefs.DisableConsistencyCheck = True
Else
    .ApplicationWindow.UserInterfacePrefs.DisableConsistencyCheck = False
End If
```



```
'Example: DisableExportToNotes property
'This example asks the user whether to disable exporting of Notes/FX TeamSecurity
fields,
'and then sets the appropriate option.

stat = MessageBox ("Do you want to disable Notes/FX?", 36, "Example Script")
If stat = 6 Then ' user said yes
    .ActiveDocument.DocControl.DisableExportToNotes = False
Else
    .ActiveDocument.DocControl.DisableExportToNotes = True
End If
```

'Example: DisableVersionReview property
'This example asks the user whether to keep editors from looking at prior versions,
'and then sets the appropriate option.

```
stat = MessageBox ("Do you want to disable version review?", 36, "Example Script")
If stat = 6 Then ' user said yes
    .ActiveDocument.DocControl.DisableVersionReview = True
Else
    .ActiveDocument.DocControl.DisableVersionReview = False
End If
```

```
'Example: DisconnectCells method
' This example creates a table with 5 columns and 4 rows based on the
' Default Table style, selects the first column, and connects the cells.
' RUNTIME DEPENDENCIES: You must have a document open for this script to work.

.CreateTable False, "Default Table", 5, 4
.SelectColumn
.ConnectCells
MessageBox "Click OK to disconnect cells .",MB_OK,"Example Script"
.DisconnectCells
```

```
'Example: Disconnect method
' This example creates a table and connects the first row of cells. After the
' message box is closed, the table cells are disconnected.
' RUNTIME DEPENDENCIES: You must have a document open for this script to work.

.CreateTable False, "Default Table", 5,5
.SelectRow
.ConnectCells
MessageBox "Click OK to disconnect cells. ", MB_OK, "Example Script"
.TableContainer.Disconnect
```

'Example: DivisionName property

'This example displays the current division name in a message box.

```
DivName = .Container.DivisionName
```

```
MessageBox "The current division name is " & DivName, MB_OK, "Example Script"
```

```
'Example: DivisionOptions property
'This example toggles the option to show hidden text in the division.
.Text.Attributes.HiddenMode = True
.Type "Now is the time for all good men to come to the aid of their party."
.Text.Attributes.HiddenMode = False
CurrentDocument.DivisionOptions.ShowHiddenText = False
MessageBox "Click OK to show hidden text", MB_OK, "Example Script"
CurrentDocument.DivisionOptions.ShowHiddenText = True
```

```
'Example: DocControlPassword property
'This example changes access to the TeamSecurity dialog box to password protected
'only. Note that this script will fail if the current access to the
'TeamSecurity dialog box is set to password protected.

' First, define the password
.ActiveDocument.DocControl.DocControlPassword = "Secret"
' Then change the security option
.ActiveDocument.DocControl.DocControlProtection = $LwpDocProtectTypePassword
```

```
'Example: DocControlProtection property
'This example changes access to the TeamSecurity dialog box to the current
'editor only. Note that this script will fail if the current access to the
'TeamSecurity dialog box is set to password protected.

' First, set the name of the editor who can access TeamSecurity
.ActiveDocument.DocControl.DocControlRestrictedToEditor = .Preferences.UserName
' Then change the security option
.ActiveDocument.DocControl.DocControlProtection = $LwpDocProtectTypeOnlyEditor
```



```
'Example: DocControlRestrictedToEditor property
'This example changes access to the TeamSecurity dialog box to the current
'editor only. Note that this script will fail if the current access to the
'TeamSecurity dialog box is set to password protected.

' First, set the name of the editor who can access TeamSecurity
.ActiveDocument.DocControl.DocControlRestrictedToEditor = .Preferences.UserName
' Then change the security option
.ActiveDocument.DocControl.DocControlProtection = $LwpDocProtectTypeOnlyEditor
```

'Example: DocControl property

'This example changes access to the TeamSecurity dialog box to the current editor only. Note that this script will fail if the current access to the TeamSecurity dialog box is set to password protected.

' First, set the name of the editor who can access TeamSecurity

.ActiveDocument.**DocControl**.DocControlRestrictedToEditor = .Preferences.UserName

' Then change the security option

.ActiveDocument.**DocControl**.DocControlProtection = \$LwpDocProtectTypeOnlyEditor

'Example: DocSize property

'This example prints the size of the current document to the LotusScript output panel.

Print "The size of the document is " & .ActiveDocument.DocInfo.**DocSize** & " bytes."

'Example: DocumentLevel property

'This example assigns an outline level of 1 to the Heading 1 paragraph style

'and an outline level of 2 to the Heading 2 paragraph style.

```
.Division.Foundry.ParagraphStyles("Heading 1").DocumentLevel = 1
```

'Make this a heading style

```
.Division.Foundry.ParagraphStyles("Heading 1").Heading = 1
```

'Assign the Default Outline numbering sequence, and set the numbering position to 1.

```
.Division.Foundry.ParagraphStyles("Heading 1").Bullet.Name = "Default Outline"
```

```
.Division.Foundry.ParagraphStyles("Heading 1").NumberingPosition = 1
```

```
.Division.Foundry.ParagraphStyles("Heading 2").DocumentLevel = 2
```

```
.Division.Foundry.ParagraphStyles("Heading 2").Heading = 1
```

```
.Division.Foundry.ParagraphStyles("Heading 2").Bullet.Name = "Default Outline"
```

```
.Division.Foundry.ParagraphStyles("Heading 2").NumberingPosition = 2
```

'Example: DoesMarkerNameMatch method

'This example creates a table, then places a default marker in a specific cell.

'Each cell layout is examined to determine if it contains the marker. When a

'match is found, the insertion point is placed in the layout.

```
.CreateTable False, "Default Table", 4, 4
```

```
Temp = .Table.CellLayout(2,3).Mark($LwpMarkerTypeDefault)
```

```
For i = 0 To 3
```

```
    For j = 0 To 3
```

```
        If .Table.CellLayout(i,j).DoesMarkerNameMatch(Temp) Then
```

```
            .Table.CellLayout(i,j).GoToLayout
```

```
        End If
```

```
    Next
```

```
Next
```

```
'Example: DoInitialCaps property
'This example asks the user whether to have SmartCorrect fix errors in initial
'capitalization, and then sets the appropriate option.

stat = MessageBox ("Do you want to fix TWo initial caps errors?", 36, "Example
Script")
If stat = 6 Then ' user said yes
    .Application.SmartCorrect.DoInitialCaps = True
Else
    .Application.SmartCorrect.DoInitialCaps = False
End If
```

```
'Example: DoSentenceInitialCaps property
'This example asks the user whether to have SmartCorrect automatically
'capitalize the first word in each sentence, and then sets the appropriate option.

stat = MessageBox ("Do you want to automatically capitalize sentences?", 36, "Example
Script")
If stat = 6 Then ' user said yes
    .Application.SmartCorrect.DoSentenceInitialCaps = True
Else
    .Application.SmartCorrect.DoSentenceInitialCaps = False
End If
```

```
'Example: DoSmartLinks property
'This example asks the user whether to have SmartCorrect automatically create
'hyperlinks for phrases beginning with http://, and then sets the appropriate option.

stat = MessageBox ("Do you want to automatically add Hyperlinks?", 36, "Example
Script")
If stat = 6 Then ' user said yes
    .Application.SmartCorrect.DoSmartLinks = True
Else
    .Application.SmartCorrect.DoSmartLinks = False
End If
```



```
'Example: DoSmartQuotes property
'This example asks the user whether to have SmartCorrect change regular quotes
'to SmartQuotes, and then sets the appropriate option.
'This example is not viable for OS/2.

stat = Messagebox ("Do you want to change quotes to SmartQuotes?", 36, "Example
Script")
If stat = 6 Then   ' user said yes
    .Application.SmartCorrect.DoSmartQuotes = True
Else
    .Application.SmartCorrect.DoSmartQuotes = False
End If
```

'Example: DoubleUnderline property
'This example enables double underlining, inserts double underlined text,
'then removes the attribute.

```
.Text.Font.DoubleUnderline = True  
.Text.InsertText "This is double underlined text."  
.Text.Font.DoubleUnderline = False
```

'Example: DoVerb method

'This example imports a bitmap from the Word Pro directory. The bitmap is then placed in edit mode using the DoVerb method.

```
.ImportGraphic .path & "\helpbutn.bmp", ".bmp", False, False, "Default Graphic/OLE"
```

```
Msgbox "Click OK to edit the Bitmap Object."
```

```
.DoVerb 0
```

```
'Example: DownloadGraphics property
'This example asks the user whether to download graphics from the Internet
'when importing HTML, and then sets the appropriate option.

stat = MessageBox ("Do you want to download graphics from the Internet?", 36, "Example
Script")
If stat = 6 Then ' user said yes
    .ApplicationWindow.UserInterfacePrefs.HTMLOptions.DownloadGraphics = True
Else
    .ApplicationWindow.UserInterfacePrefs.HTMLOptions.DownloadGraphics = False
End If
```

```
'Example: DragDropOn property
'This example asks the user whether to disable drag and drop,
'and then sets the appropriate option.

stat = MessageBox ("Do you want to disable drag and drop?", 36, "Example Script")
If stat = 6 Then ' user said yes
    .Application.Preferences.DragDropOn = False
Else
    .Application.Preferences.DragDropOn = True
End If
```

```
'Example: EditorManager property
'This example reviews all the editors for the document, and sets their rights to "not
allowed".
Forall DocEditor In .ActiveDocument.EditorManager.Editors
    If DocEditor.Name <> .ActiveDocument.EditorManager.CurrentEditor.Name Then
        DocEditor.Abilities =$LwpEditAbilEditingNotAllowed
    End If
End Forall
```

'Example: EditorName property

'This example displays the name of the person who created the document,

'and the name of the person who created the page layout.

EdName = .ActiveDocument.**EditorName**

MessageBox "The person who created this document was " & EdName & ".", MB_OK, "Example
Script"

EdName = .Layout.**EditorName**

MessageBox "The person who created this layout was " & EdName & ".", MB_OK, "Example
Script"

'Example: EditorVerificationType property
'This example displays a message box with the current verification
'type. The type is then changed, and finally reset to its original value
'NOTE: The TeamSecurity dialog box must not be password protected in
'order to use this example.

```
CurType = .ActiveDocument.DocControl.EditorVerificationType
MessageBox "The current editor verification type is " & Curtype, MB_OK, "Example
Script"
' set the type to OS Login
.ActiveDocument.DocControl.EditorVerificationType = $LwpEditorVerifyTypeByOpsysLogin
MessageBox "The current editor verification type is "
& .ActiveDocument.DocControl.EditorVerificationType , MB_OK, "Example Script"
' reset to original type
.ActiveDocument.DocControl.EditorVerificationType = CurType
```


Example: EditWrap method

'This example imports a graphic into the current document,
'then displays the frame's wrap handles.

```
.ImportGraphic "E:\lotus\wordpro\graphics\apple.sdw", ".sdw", False, False, "Default  
Graphic/OLE"  
.Layout.WrapType = $LwpWrapTypeLayoutWrapIrregBiggest  
MessageBox "Click OK to display the frame's irregular wrap handles.", MB_OK, "Example  
Script"  
.Frame.EditWrap
```

'Example: EmbedFonts property

'This example asks the user whether to embed fonts in the document

'and then sets the appropriate option.

```
stat = MessageBox ("Do you want to disable embed fonts?", 36, "Example Script")
```

```
If stat = 6 Then ' user said yes
```

```
    .ActiveDocument.DocOptions.EmbedFonts = 1
```

```
Else
```

```
    .ActiveDocument.DocOptions.EmbedFonts = 0
```

```
End If
```

```
'Example: EndChange method
' This example creates a table with 5 columns and 5 rows. Several background
' table cell properties are changed all at one using the BeginChange and
' EndChange methods.
' RUNTIME DEPENDENCIES: You must have a document open for this script to work.
```

```
Dim ExampleTable As Table
Dim ExampleCell As CellLayout

.CreateTable False, "Default Table", 5,5
Set ExampleTable = .Table
ExampleTable.CellLayout(1,0).GotoLayout

Set ExampleCell = ExampleTable.CellLayout(1,0)

.BeginChange
ExampleCell.Content.InsertText "Hello"
With ExampleCell.BackGround
    .Pattern = $LtsFillSolid
    .Color.Red = 255
    .Color.Blue = 194
    .Color.Green = 255
    .Color.Override = $LwpColorOverrideRgb
    .BackColor.Red = 65
    .BackColor.Blue = 176
    .BackColor.Green = 0
End With

.EndChange
```

'Example: Ending method

' This example moves the insertion point to the end of the document.

' RUNTIME DEPENDENCIES: You must have a document open and the insertion point

' within a frame for this script to work.

.CreateFrame False, "Default Frame",1440, 1440

.Page.Ending \$LwpContainerEndEndOfDocument

'Example: EndnoteDivisionGroupNum property

'This example sets options for numbering footnotes at the end of a division group.

```
.Division.FootnoteOptions.EndnoteDivisionGroupNum.StartingNumber = 1  
.Division.FootnoteOptions.EndnoteDivisionGroupNum.LeadingText = "("  
.Division.FootnoteOptions.EndnoteDivisionGroupNum.TrailingText = ")"  
.Division.FootnoteOptions.EndnoteDivisionGroupNum.UseSuperscriptReferenceNum = False  
.Division.FootnoteOptions.EndnoteDivisionGroupNum.ResetWhen =  
$LwpResetOptionEachDivisiongroup
```

'Example: EndnoteDivisionNum property

'This example sets numbering options for footnotes at the end of a division.

.Division.FootnoteOptions.**EndnoteDivisionNum**.StartingNumber = 1

.Division.FootnoteOptions.**EndnoteDivisionNum**.LeadingText = ""

.Division.FootnoteOptions.**EndnoteDivisionNum**.TrailingText = ""

.Division.FootnoteOptions.**EndnoteDivisionNum**.UseSuperscriptReferenceNum = True

.Division.FootnoteOptions.**EndnoteDivisionNum**.ResetWhen = \$LwpResetOptionEachDivision

'Example: EndnoteDocNum property

'This example sets numbering options for endnotes at the end of the document.

.Division.FootnoteOptions.**EndnoteDocNum**.StartingNumber = 1

.Division.FootnoteOptions.**EndnoteDocNum**.LeadingText = ""

.Division.FootnoteOptions.**EndnoteDocNum**.TrailingText = ""

.Division.FootnoteOptions.**EndnoteDocNum**.UseSuperscriptReferenceNum = False

.Division.FootnoteOptions.**EndnoteDocNum**.ResetWhen = \$LwpResetOptionEachDoc

```
'Example: EntireDocument property
'This example asks the user if the entire document should be exported to
'HTML, then sets the appropriate option.

stat = MessageBox ("Do you want to export the entire document?", 36, "Example Script")
If stat = 6 Then ' user said yes
    .ApplicationWindow.UserInterfacePrefs.HTMLOptions.EntireDocument = True
Else ' export only the current division
    .ApplicationWindow.UserInterfacePrefs.HTMLOptions.EntireDocument = False
End If
```



```
'Example: EnumerateTerm method
' This example enumerates all the entries in the glossary named GLOSSARY.GLS.
' RUNTIME DEPENDENCIES: You must have a glossary named GLOSSARY.GLS and a
' document open for this script to work.

.ApplicationWindow.UserInterfacePrefs.OpenDocsVisible = False
.ApplicationWindow.UserInterfacePrefs.OpenReadOnly = True
.GlossaryOpen "GLOSSARY.GLS", "Lotus Word Pro"
.ApplicationWindow.UserInterfacePrefs.OpenDocsVisible = True
.ApplicationWindow.UserInterfacePrefs.OpenReadOnly = False
Forall Gloss In .Division.Foundry.Glossarys
    Count% = Gloss.NumRows
    For Item% = 1 To (Count% - 1)
        Print "Item" & Format$(Item%) & "= " & Gloss.EnumerateTerm(Item%)
    Next
End Forall
.Close
```

```
'Example: Enumeration property
'This example sets the value of the Enumeration property to $LwpEnumScopeLocal.
'This script excludes any ParagraphStyle objects which originated
'with the SmartMaster from the ParagraphStyleCollection object in
'the currently active division's Foundry. This script then
'prints the names of any locally created ParagraphStyle objects
'in the Output panel.
'RUNTIME DEPENDENCIES: You must create one or more ParagraphStyle
'objects in addition to those in the SmartMaster. Otherwise, the
'Output panel will remain empty. You must have a document open for
'this script to work.
'Paste this example script into Sub Main and run it.

.Foundry.Enumeration = $LwpEnumScopeLocal
Forall ParagraphStyle In .Foundry.ParagraphStyles
    Print ParagraphStyle.Name
End Forall
```

```
'Example: ExchangeItem method
' This example exchanges the menu locations for the File and Create menus.
' RUNTIME DEPENDENCIES: You must have a document open for this script to work.

Dim MenuBar As MenuItem
Set MenuBar = .ApplicationWindow.LwpMenuBar
MenuBar.Items("&File").ExchangeItem MenuBar.Items("&Create")
```

```
'Example: ExitClickHere event
Sub Exitclickhere(Source As Clickhere, Clickherename As String)
Const Maximum = 15
Temp$ = Source.GetMarkedText
If Len(Temp$) > Maximum Then
    MsgBox "Contents is too long, try again"
    Source.Goto(True)
'This would notify the user of the problem,
'then put them back in the ClickHere block. The parameter True on the
'Source.Goto line is to actually select the contents of the ClickHere block.
'False would simply put them back in the ClickHere.
End If
End Sub
```

'Example: Expandable property

'This example shows the effects of Expandable and Collapsible in extending the
'boundaries of a text marker. The final contents of the marker is displayed
'in a message box.

```
MyMark = .Text.Mark ($LwpMarkerTypeDefault)
.Text.InsertText "1: This is marked text. " ' not included, since expandable set to 0
.Division.Foundry.Markers(MyMark).Expandable =1
.Text.InsertText "2: This is marked text. " ' included in marker
.Division.Foundry.Markers(MyMark).Expandable =0
.Division.Foundry.Markers(MyMark).Collapsible = 1
.Text.InsertText "3: This is marked text" ' not included, collapsible = 1
DialogBox "The marked text is: " & .Division.Foundry.Markers(MyMark).GetMarkedText() &
".", MB_OK, "Example Script"
.Division.Foundry.Markers(MyMark).DeleteMarker
```

```
'Example: ExpandOutlineLevel method
'This example inserts text in an outline structure. It then contracts and
'expands the outline.

.Text.Bullet.Name = "Default Outline"
.Text.Numbering.Position = 3
.Text.Indent.First = 360
.Text.Indent.Rest = 360
.Type "This is outline 1[Enter]"
.DemoteOutlineLevel
.Type "This is outline 2[Enter]"
.DemoteOutlineLevel
.Type "This is outline 3[Enter]"
.DemoteOutlineLevel
.Type "This is outline 4[Up][Up]"
'move up 2 lines after typing this one
MessageBox"Click OK to hide subordinate outline levels", MB_OK, "Example Script"
.ContractOutlineLevel
MessageBox"Click OK to show subordinate outline levels", MB_OK, "Example Script"
.ExpandOutlineLevel
```

```
'Example: Expand method
' This example creates two child divisions and then contracts and expands the
' divider tabs.
' RUNTIME DEPENDENCIES: You must have a document open for this script to work.

' show the tabs, and create the child divisions
.ActiveDocWindow.WinViewPrefs.IsViewSectionTabs = True
.ApplicationWindow.SectionTabs.ConnectSectionTabs
.ApplicationWindow.SectionTabs.ConnectSectionTabs
MsgBox "Click OK to contract the tabs.", MB_OK, "Example Script"

.ApplicationWindow.SectionTabs.Contract
MsgBox "Click OK to expand the tabs.", MB_OK, "Example Script"
.ApplicationWindow.SectionTabs.Expand
```

```
'Example: ExportAllAsNotesFX method
' This example inserts two docinfo fields and makes them exportable to Lotus
' Notes.
' RUNTIME DEPENDENCIES: You must have a document open for this script to work.

.ActiveDocument.DocInfo.IsExportedAsNotesFX $LwpDocVarVersionnumrevisions
.InsertDocInfo $LwpDocVarCreatedby
.Text.SplitParagraph
.InsertDocInfo $LwpDocVarDatecreated
.ActiveDocument.DocInfo.ExportAllAsNotesFX True
.ActiveDocument.DocInfo.UpdateSelectedFields
```


'Example: ExportAsNotesFX method

'This example disables exporting the FileName document field to Notes/FX,

'and enables exporting the Date Created fields to Notes/FX.

.ActiveDocument.DocInfo.**ExportAsNotesFX** \$LwpDocVarFilename, False

.ActiveDocument.DocInfo.**ExportAsNotesFX** \$LwpDocVarDatecreated, True

'Example: ExportToNotesFX property

'This example creates a new document field named Field1, and allows the export
'of the newly created field to Notes/FX.

```
.ActiveDocument.DocInfo.FieldManager.AddField "Field1", "", 0
```

```
.ActiveDocument.DocInfo.FieldManager.Fields("Field1").ExportToNotesFX = 1
```

```
'Example: ExternalName property
'This example displays a message box with the internal and external names
'of the currently selected division. All divisions in the document are examined,
'and if an external division is found, the external file name of the
'division is displayed.

Messagebox "The user defined name of the current division is "
& .ApplicationWindow.SectionTabs.ExternalName _
& " and the internal name is " & .ApplicationWindow.SectionTabs.Name & ".", MB_OK,
"Example Script"
Forall Div In .ActiveDocument.Divisions
    If Div.IsDivisionExternal Then
        Messagebox "External Division " & Div.DivisionInfo.Name & " is derived from
the file " _
        & Div.DivisionInfo.ExternalName & ".", MB_OK, "Example Script"
    End If
End Forall
```

```
'Example: ExtractText method
' This example opens the glossary file named GLOSSARY.GLS and prints all of
' the glossary entries to the LotusScript Output panel.
' RUNTIME DEPENDENCIES: You must have a document open for this script to work.

.ApplicationWindow.UserInterfacePrefs.OpenDocsVisible = False
.ApplicationWindow.UserInterfacePrefs.OpenReadOnly = True
.GlossaryOpen "GLOSSARY.GLS", "Lotus Word Pro"
.ApplicationWindow.UserInterfacePrefs.OpenDocsVisible = True
.ApplicationWindow.UserInterfacePrefs.OpenReadOnly = False
Forall Gloss In .Division.Foundry.Glossarys
    Count% = Gloss.NumRows
    For Item% = 1 To (Count% - 1)
        GlossItem$ = Gloss.EnumerateTerm(Item%)
        GlossText$ = Gloss.ExtractText(GlossItem$)
        Print GlossText$
    Next
End Forall
.Documents("GLOSSARY.GLS").Close
```

'Example: FaxNumber property
'This example lets the user change the fax number found in user preferences.
'If the user changes the number, the new number is saved.

```
FaxNum = .Preferences.Faxnumber  
NewNum = Inputbox ("Type a new fax number:", "Example Script", FaxNum)  
If NewNum <> "" Then ' user didn't cancel  
    .Preferences.Faxnumber = NewNum  
End If
```

'Example: FieldDelimiterText property

'This example creates a comma delimited list, and sorts it.

```
.NewDocument
.Text.InsertText "Manufacturing, 555-1234", True
.Text.InsertText "Distribution, 555-6845", True
.Text.InsertText "Sales, 555-9425", True
.ActiveDocument.SortOptions.SortLevel1.FieldNumber = 1
.ActiveDocument.SortOptions.SortLevel1.SortType = $LwpSortTypeAlphanumeric
.ActiveDocument.SortOptions.SortLevel1.SortOrder = $LtsSortAscending
.ActiveDocument.SortOptions.NumParagraphs = 1
.ActiveDocument.SortOptions.SortNumbers = $LwpSortNumberOrderLast
.ActiveDocument.SortOptions.FieldDelimiter = $LwpDelimiterTypeTextdelimited
.ActiveDocument.SortOptions.FieldDelimiterText = ","
.SortParagraphs
```

'Example: FieldDelimiter property

'This example types a comma-delimited list, and sorts it on the first field.

'The field delimiter is set to the comma character.

.NewDocument

.Text.InsertText "Manufacturing, 555-1234", True

.Text.InsertText "Distribution, 555-6845", True

.Text.InsertText "Sales, 555-9425", True

.ActiveDocument.SortOptions.SortLevel1.FieldNumber = 1

.ActiveDocument.SortOptions.SortLevel1.SortType = \$LwpSortTypeAlphanumeric

.ActiveDocument.SortOptions.SortLevel1.SortOrder = \$LtsSortAscending

.ActiveDocument.SortOptions.NumParagraphs = 1

.ActiveDocument.SortOptions.SortNumbers = \$LwpSortNumberOrderLast

.ActiveDocument.SortOptions.**FieldDelimiter** = \$LwpDelimiterTypeTextdelimited

.ActiveDocument.SortOptions.FieldDelimiterText = ","

.SortParagraphs

Example: FilePassword property

'This example creates a password for the current document. First, the protection type is set to password protection, and then the password is changed to 'Secret'.

'This script will not run if access to the TeamSecurity dialog box is password protected, or if the current editor does not have access to the TeamSecurity dialog box.

```
.ActiveDocument.DocControl.FileProtectionType = $LwpFileProtectTypePassword  
.ActiveDocument.DocControl.FilePassword = "Secret"
```


'Example: FileProtectionType property

'This example sets the restricts editing of the current document to
'current editors only.

```
.ActiveDocument.DocControl.FileProtectionType = $LwpFileProtectTypeEditors
```

'Example: FillerPageText property

'This example changes the text used for filler pages to "I am blank by design".

.Division.DivisionInfo.**FillerPageText**.Clear ' delete old filler text

.Division.DivisionInfo.**FillerPageText**.InsertText "I am blank by design." ' and insert
new

```
'Example: FindAndReplace property
' This example inserts three identical sentences into the current document,
' clears the FindAndReplace settings, sets the FindString to "cat" and the
' ReplaceString to "dog", and then displays a message box.
' When you click OK, the script finds the first 'cat' then replaces it with
' 'dog'.
' RUNTIME DEPENDENCIES: You must have a document open for this script to work.

Dim SentenceCount As Integer
For SentenceCount = 1 To 3
    .Text.InsertText "The sleep-deprived fox jumped over the gravity-challenged cat."
    .Text.SplitParagraph
Next
.Application.ResetFindAndReplace

.Application.FindAndReplace.FindString = "cat"
.Application.FindAndReplace.ReplaceString = "dog"

.InitFindAndReplace True
MessageBox "Click OK to find and then replace.",MB_OK,"Example Script"
.Find
.ReplaceCmd
```

'Example: FindCellLayout method
'This example creates a table, and inserts information in cells A1 and A2,
'thereby creating layout objects for these cells.
'The FindCellLayout method is then called to determine if layouts exist
'in various cells.

```
.CreateTable False, "Default Table", 3, 2
.Type "Text[Down]1.234[Up]"
stat = .Table.FindCellLayout(0,1)
If stat = 0 Then
    MsgBox "There is no layout in cell B1", MB_OK, "Example Script"
Else
    MsgBox "There is a layout in cell B1", MB_OK, "Example Script"
End If
stat = .Table.FindCellLayout(0,0)
If stat = 1 Then
    MsgBox "There is a layout in cell A1", MB_OK, "Example Script"
Else
    MsgBox "There is no layout in cell A1", MB_OK, "Example Script"
End If
```

```
'Example: FindClass method
' This example obtains the header class name for the current document and
' uses that name print some header properties to the Lotus Script Output
' panel.
' RUNTIME DEPENDENCIES: You must have a document open for this script to work.

Dim HeaderComponent As String
HeaderClass = .Layout.FindClass("Header")
Print .Division.Foundry.Layouts.Item(HeaderClass).ClassName

Print .Division.Foundry.Layouts.Item(HeaderClass).MarginLeft
Print .Division.Foundry.Layouts.Item(HeaderClass).MarginTop
Print .Division.Foundry.Layouts.Item(HeaderClass).Background.Color.GetRGB()
Print .Division.Foundry.Layouts.Item(HeaderClass).Content.GetText($LwpGetObjectTypeParagraph, False)
```

'Example: FindExactCase property

'This example executes a find. Because the option to find exact case is set
'to True, only the second instance of 'white' is found.

```
.Text.InsertText "The White House is painted white."  
.Application.FindAndReplace.FindString = "white"  
.Application.FindAndReplace.FindExactCase = True  
.InitFindAndReplace True  
.Find
```

'Example: FindFont property

'This example types two identical sentences except for the font.

'The example then executes a find, specifying the font used in the second
'sentence.

```
.Type "Now is the time for all good men to come to the aid of their country.[Enter]"
```

```
.Text.Font.FontName = "Arial"
```

```
.Type "Now is the time for all good men to come to the aid of their country.[Enter]"
```

```
.Application.FindAndReplace.FindString = "country"
```

```
.Application.FindAndReplace.FindFont.Clear ' get rid of current attributes
```

```
.Application.FindAndReplace.FindFont.FontName = "Arial" ' set the font name
```

```
.Application.FindAndReplace.FindWithProperties = True
```

```
.Application.FindAndReplace.UseFindStyle = 0
```

```
.InitFindAndReplace True
```

```
.Find
```

```
'Example: FindString property
' This example inserts three identical sentences into the current document,
' clears the FindAndReplace settings, sets the FindString to "cat" and the
' ReplaceString to "dog", and then displays a message box.
' When you click OK, the script finds the first 'cat' then replaces it with
' 'dog'.
' RUNTIME DEPENDENCIES: You must have a document open for this script to work.

Dim SentenceCount As Integer
For SentenceCount = 1 To 3
    .Text.InsertText "The sleep-deprived fox jumped over the gravity-challenged cat."
    .Text.SplitParagraph
Next
.Application.ResetFindAndReplace

.Application.FindAndReplace.FindString = "cat"

.Application.FindAndReplace.ReplaceString = "dog"
.InitFindAndReplace True
MessageBox "Click OK to find and then replace.",MB_OK,"Example Script"
.Find
.ReplaceCmd
```



```

'Example: FindTerm method
' This example opens the glossary file named 'GLOSSARY.GLS', searches for the
' glossary item named 'TestItem'. If the item is found, its glossary value is
' printed to the Lotus Script Output panel.
' RUNTIME DEPENDENCIES: You must have a document open for this script to work.

.ApplicationWindow.UserInterfacePrefs.OpenDocsVisible = False
.ApplicationWindow.UserInterfacePrefs.OpenReadOnly = True
GlossName = .ApplicationWindow.UserInterfacePrefs.GlossaryPath & "\GLOSSARY.GLS"
.GlossaryOpen GlossName, "Lotus Word Pro"
.ApplicationWindow.UserInterfacePrefs.OpenDocsVisible = True
.ApplicationWindow.UserInterfacePrefs.OpenReadOnly = False
Forall Gloss In .Division.Foundry.Glossarys
    GlossItem$ = "TestItem"
    Status% = Gloss.FindTerm(GlossItem$)
    If Status% = 1 Then
        GlossText$ = Gloss.ExtractText(GlossItem$)
        Print GlossText$
        Exit Forall
    End If
End Forall
.Documents("GLOSSARY.GLS").Close

```

Word Pro: DropCapLayoutCollection class members

Properties

Application AS WPAApplication class

Count

Description

IsValid

Name

Parent AS BaseObject class

VersionID

Methods

IsEmpty

Item

Events

None

Word Pro: DropCapLayout class members

Properties

[AbsoluteOn](#)
[AbsoluteXPos](#)
[AbsoluteYPos](#)
[Accelerator](#)
[AllowResizeWhenCrop](#)
[AmtTether](#)
[AmtToRotateContent](#)
[AmtToTetherFrom](#)
[Application](#) AS [WPAApplication class](#)
[Background](#) AS [Background class](#)
[BaseLineOffset](#)
[BinName](#)
[BorderLines](#) AS [BorderLines class](#)
[BorderOffset](#)
[BottomExternalMargin](#)
[Center](#)
[ChildLayouts](#) AS [LayoutCollection class](#)
[ClassName](#)
[ColumnBalance](#)
[ColumnGap](#)
[ConditionType](#)
[Content](#)
[ContentName](#)
[ContentStyleName](#)
[Definition](#)
[Description](#)
[DirectionDown](#)
[DirectionLeft](#)
[DirectionRight](#)
[DirectionUp](#)
[DropCapPosition](#)
[EditorName](#)
[Footer](#) AS [Layout class](#)
[GridDistance](#)
[GridType](#)
[Gutter](#) AS [Gutter class](#)
[Header](#) AS [Layout class](#)
[Height](#)
[IsBreakable](#)
[IsChildSpannable](#)
[IsCollapsed](#)
[IsCollapsible](#)
[IsColumnBreakable](#)
[IsComplex](#)

IsConnected
IsErrorChecking
IsExpandDown
IsExpandLeft
IsExpandRight
IsExpandUp
IsIndexMarkAble
IsLocal
IsLocked
IsMarginSameAsParent
IsNotCopyable
IsNotGroupable
IsNoUICommAllowed
IsOverridden
IsOverride
IsPageBreak
IsPartOfGroup
IsPrintable
IsProtected
IsScrollable
IsSingleClickEntry
IsSizable
IsSnapTo
IsStyle
IsTableHeading
IsTOC
IsTOCMarkAble
IsValid
Join AS Join class
Justifiable
Layer AS Layout class
LayerName
LeaderDotType
LeftExternalMargin
LeftPage AS Layout class
LeftTopCellId
LineLocation
LinkFrame
MaintainAspectRatio
MarginBottom
MarginLeft
MarginRight
MarginTop
MasterName
MinBottomMargin
MinHeight
MinLeftMargin

[MinRightMargin](#)
[MinTopMargin](#)
[Name](#)
[NameBasedOnStyle](#)
[NumberOfLines](#)
[NumCols](#)
[NumColsSpannedOneCell](#)
[NumericFormat](#) AS [NumericFormat class](#)
[NumRowsSpannedOneCell](#)
[PageToUseLayoutOn](#)
[PaperName](#)
[Parent](#) AS [BaseObject class](#)
[RelativeType](#)
[RelativeXDistance](#)
[RelativeYDistance](#)
[RevisionType](#)
[RightExternalMargin](#)
[RightPage](#) AS [Layout class](#)
[ScaleHeight](#)
[ScaleMode](#)
[ScalePercentage](#)
[ScaleWidth](#)
[SelectType](#)
[Shadow](#) AS [Shadow class](#)
[Span](#)
[Style](#) AS [Layout class](#)
[StyleExceptions](#)
[TabRack](#) AS [TabRack class](#)
[TextOrient](#)
[Tile](#)
[TopExternalMargin](#)
[TopLeftCellRowId](#)
[UseFooter](#)
[UseHeader](#)
[UsePrinterSettings](#)
[UseWhen](#) AS [UseWhen class](#)
[VersionID](#)
[VertAlign](#)
[WasDeletedInRevMarkMode](#)
[WasInsertedInRevMarkMode](#)
[Where](#)
[Width](#)
[WPDataSets](#) AS [WPDataSetCollection class](#)
[WrapType](#)
[XOffset](#)
[XPosition](#)
[YOffset](#)

YPosition

Methods

AddChildToLayout

Backward

CreateLayer

DeleteContents

DeleteLayout

DoesMarkerNameMatch

FindClass

Forward

GetMarkerName

GetNamedProperty

GoToLayout

HasNamedProperty

ImportWatermarkGraphic

Mark

MirrorPage

MoveToBack

MoveToFront

RegisterWPDataSet

RemoveChildFromLayout

RemoveNamedProperty

RevisionAcceptLayoutChange

RevisionCancelLayoutChange

SetAllMargins

SetMinimumOrigin

SetNamedProperty

SetPaperNameAndUpdateSize

SetupForCropping

UnregisterWPDataSet

Update

Events

EnterLayout

KeyStroke

MouseDown

MouseUp

Word Pro: EditorCollection class members

Properties

Application AS WPAApplication class

Count

Description

IsValid

Name

Parent AS BaseObject class

VersionID

Methods

IsEmpty

Item

Events

None

Word Pro: EditorManager class members

Properties

Application AS WPAApplication class

CurrentEditor AS Editor class

Description

Editors AS EditorCollection class

IsValid

Name

Parent AS BaseObject class

VersionID

Methods

AddEditorManager

RemoveEditor

Events

None

Word Pro: Editor class members

Properties

Abilities

Application AS WPAApplication class

DeleteFont AS Font class

Description

EditorInitials

HiLiteColor AS Color class

InsertFont AS Font class

IsValid

Locks

Name

Parent AS BaseObject class

Suggestions

TextAttributes AS Attributes class

VersionID

Methods

None

Events

None

Word Pro: EndnoteDivisionGroupNum class members

Properties

Application AS WPAApplication class

Description

IsValid

LeadingText

Name

Parent AS BaseObject class

ResetWhen

StartingNumber

TrailingText

UseSuperscriptReferenceNum

VersionID

Methods

None

Events

None

Word Pro: EndnoteDivisionNum class members

Properties

Application AS WPAApplication class

Description

IsValid

LeadingText

Name

Parent AS BaseObject class

ResetWhen

StartingNumber

TrailingText

UseSuperscriptReferenceNum

VersionID

Methods

None

Events

None

Word Pro: EndnoteDocNum class members

Properties

Application AS WPAApplication class

Description

IsValid

LeadingText

Name

Parent AS BaseObject class

ResetWhen

StartingNumber

TrailingText

UseSuperscriptReferenceNum

VersionID

Methods

None

Events

None

Word Pro: EndnoteLayoutCollection class members

Properties

Application AS WPAApplication class

Count

Description

IsValid

Name

Parent AS BaseObject class

VersionID

Methods

IsEmpty

Item

Events

None

Word Pro: EndnoteLayout class members

Properties

[AbsoluteOn](#)
[AbsoluteXPos](#)
[AbsoluteYPos](#)
[Accelerator](#)
[AllowResizeWhenCrop](#)
[AmtTether](#)
[AmtToRotateContent](#)
[AmtToTetherFrom](#)
[Application](#) AS [WPAApplication class](#)
[Background](#) AS [Background class](#)
[BaseLineOffset](#)
[BinName](#)
[BorderLines](#) AS [BorderLines class](#)
[BorderOffset](#)
[BottomExternalMargin](#)
[Center](#)
[ChildLayouts](#) AS [LayoutCollection class](#)
[ClassName](#)
[ColumnBalance](#)
[ColumnGap](#)
[ConditionType](#)
[Content](#)
[ContentName](#)
[ContentStyleName](#)
[Definition](#)
[Description](#)
[DirectionDown](#)
[DirectionLeft](#)
[DirectionRight](#)
[DirectionUp](#)
[DropCapPosition](#)
[EditorName](#)
[Footer](#) AS [Layout class](#)
[GridDistance](#)
[GridType](#)
[Gutter](#) AS [Gutter class](#)
[Header](#) AS [Layout class](#)
[Height](#)
[IsBreakable](#)
[IsChildSpannable](#)
[IsCollapsed](#)
[IsCollapsible](#)
[IsColumnBreakable](#)
[IsComplex](#)

IsConnected
IsErrorChecking
IsExpandDown
IsExpandLeft
IsExpandRight
IsExpandUp
IsIndexMarkAble
IsLocal
IsLocked
IsMarginSameAsParent
IsNotCopyable
IsNotGroupable
IsNoUICommAllowed
IsOverridden
IsOverride
IsPageBreak
IsPartOfGroup
IsPrintable
IsProtected
IsScrollable
IsSingleClickEntry
IsSizable
IsSnapTo
IsStyle
IsTableHeading
IsTOC
IsTOCMarkAble
IsValid
Join AS Join class
Justifiable
Layer AS Layout class
LayerName
LeaderDotType
LeftExternalMargin
LeftPage AS Layout class
LeftTopCellId
LineLocation
LinkFrame
MaintainAspectRatio
MarginBottom
MarginLeft
MarginRight
MarginTop
MasterName
MinBottomMargin
MinHeight
MinLeftMargin

[MinRightMargin](#)
[MinTopMargin](#)
[Name](#)
[NameBasedOnStyle](#)
[NumberOfLines](#)
[NumCols](#)
[NumColsSpannedOneCell](#)
[NumericFormat](#) AS [NumericFormat class](#)
[NumRowsSpannedOneCell](#)
[PageToUseLayoutOn](#)
[PaperName](#)
[Parent](#) AS [BaseObject class](#)
[RelativeType](#)
[RelativeXDistance](#)
[RelativeYDistance](#)
[RevisionType](#)
[RightExternalMargin](#)
[RightPage](#) AS [Layout class](#)
[ScaleHeight](#)
[ScaleMode](#)
[ScalePercentage](#)
[ScaleWidth](#)
[SelectType](#)
[Shadow](#) AS [Shadow class](#)
[Span](#)
[Style](#) AS [Layout class](#)
[StyleExceptions](#)
[TabRack](#) AS [TabRack class](#)
[TextOrient](#)
[Tile](#)
[TopExternalMargin](#)
[TopLeftCellRowId](#)
[UseFooter](#)
[UseHeader](#)
[UsePrinterSettings](#)
[UseWhen](#) AS [UseWhen class](#)
[VersionID](#)
[VertAlign](#)
[WasDeletedInRevMarkMode](#)
[WasInsertedInRevMarkMode](#)
[Where](#)
[Width](#)
[WPDataSets](#) AS [WPDataSetCollection class](#)
[WrapType](#)
[XOffset](#)
[XPosition](#)
[YOffset](#)

YPosition

Methods

AddChildToLayout

Backward

CreateLayer

DeleteContents

DeleteLayout

DoesMarkerNameMatch

FindClass

Forward

GetMarkerName

GetNamedProperty

GoToLayout

HasNamedProperty

ImportWatermarkGraphic

Mark

MirrorPage

MoveToBack

MoveToFront

RegisterWPDataSet

RemoveChildFromLayout

RemoveNamedProperty

RevisionAcceptLayoutChange

RevisionCancelLayoutChange

SetAllMargins

SetMinimumOrigin

SetNamedProperty

SetPaperNameAndUpdateSize

SetupForCropping

UnregisterWPDataSet

Update

Events

EnterLayout

KeyStroke

MouseDown

MouseUp

Word Pro: FilterHelper class members

Properties

Application AS WPAApplication class

Description

GetFilterExtension

GetFilterExtForDialogBox

GetFilterId

IsValid

Name

Parent AS BaseObject class

VersionID

Methods

None

Events

None

Word Pro: Filter class members

Properties

Application AS WPAApplication class

AsciiCodePage

Description

FilterHelper AS FilterHelper class

GraphicExports AS StringCollection class

GraphicImports AS StringCollection class

IsAsciiCRLF

IsAsciiKeepStyle

IsValid

Name

Parent AS BaseObject class

TableExports AS StringCollection class

TableImports AS StringCollection class

TextandTableExports AS StringCollection class

TextandTableImports AS StringCollection class

VersionID

Methods

GetDisplayableFilterName

GetFileDescription

GetLastUsedFilter

IsFilterTypePresent

SetLastUsedFilter

Events

None

Word Pro: FindAndReplace class members

Properties

[Application](#) AS [WPAApplication class](#)

[CharacterSet](#) AS [CharacterSet class](#)

[Description](#)

[FindExactCase](#)

[FindFont](#) AS [Font class](#)

[FindForwardDirection](#)

[FindString](#)

[FindStyleName](#)

[FindWithProperties](#)

[IncludeList](#)

[IsValid](#)

[MatchType](#)

[Name](#)

[NumberFound](#)

[NumberOfReplacements](#)

[Parent](#) AS [BaseObject class](#)

[ReplaceAttributes](#) AS [Attributes class](#)

[ReplaceExactCase](#)

[ReplaceFont](#) AS [Font class](#)

[ReplaceLanguage](#) AS [Language class](#)

[ReplaceString](#)

[ReplaceStyleName](#)

[ReplaceWithProperties](#)

[SearchAttributes](#) AS [Attributes class](#)

[SearchLanguage](#) AS [Language class](#)

[UseFindStyle](#)

[UseReplaceStyle](#)

[VersionID](#)

[Where](#)

Methods

[Reset](#)

Events

None

Word Pro: FontMetrics class members

Properties

Application AS WPAApplication class

Description

FontName

IsSymbolic

IsTrueType

IsValid

Italic

Name

Oblique

Outline

Parent AS BaseObject class

PitchAndFamily

Shadow

SmallCaps

VersionID

Weight

Methods

None

Events

None

Word Pro: Font class members

Properties

[ActualEnumName](#)

[ActualName](#)

[Align](#)

[AlternateName](#)

[Application](#) AS [WPAApplication class](#)

[Ascent](#)

[BackColor](#) AS [Color class](#)

[BackColorIndex](#)

[Bold](#)

[Case](#)

[DefaultPitch](#)

[Descent](#)

[Description](#)

[DoubleUnderline](#)

[FaceName](#)

[FontColor](#) AS [Color class](#)

[FontMetrics](#) AS [FontMetrics class](#)

[FontName](#)

[ForeColorIndex](#)

[Height](#)

[IsTrueType](#)

[IsValid](#)

[Italic](#)

[LowerCase](#)

[LowerCaseAscent](#)

[Name](#)

[Normal](#)

[Overstrike](#)

[OverstrikeCharacter](#)

[Parent](#) AS [BaseObject class](#)

[Plain](#)

[Size](#)

[SmallCaps](#)

[StrikeThrough](#)

[Subscript](#)

[Superscript](#)

[TextTightness](#)

[Underline](#)

[UpperCase](#)

[VersionID](#)

[Width](#)

[WindowsName](#)

[WordDoubleUnderline](#)

[WordUnderline](#)

MethodsClearRevertToStyle**Events**

None

Word Pro: FooterLayoutCollection class members

Properties

Application AS WPAApplication class

Count

Description

IsValid

Name

Parent AS BaseObject class

VersionID

Methods

IsEmpty

Item

Events

None

Word Pro: FooterLayout class members

Properties

[AbsoluteOn](#)
[AbsoluteXPos](#)
[AbsoluteYPos](#)
[Accelerator](#)
[AllowResizeWhenCrop](#)
[AmtTether](#)
[AmtToRotateContent](#)
[AmtToTetherFrom](#)
[Application](#) AS [WPAApplication class](#)
[Background](#) AS [Background class](#)
[BaseLineOffset](#)
[BinName](#)
[BorderLines](#) AS [BorderLines class](#)
[BorderOffset](#)
[BottomExternalMargin](#)
[Center](#)
[ChildLayouts](#) AS [LayoutCollection class](#)
[ClassName](#)
[ColumnBalance](#)
[ColumnGap](#)
[ConditionType](#)
[Content](#)
[ContentName](#)
[ContentStyleName](#)
[Definition](#)
[Description](#)
[DirectionDown](#)
[DirectionLeft](#)
[DirectionRight](#)
[DirectionUp](#)
[DropCapPosition](#)
[EditorName](#)
[Footer](#) AS [Layout class](#)
[GridDistance](#)
[GridType](#)
[Gutter](#) AS [Gutter class](#)
[Header](#) AS [Layout class](#)
[Height](#)
[IsBreakable](#)
[IsChildSpannable](#)
[IsCollapsed](#)
[IsCollapsible](#)
[IsColumnBreakable](#)
[IsComplex](#)

IsConnected
IsErrorChecking
IsExpandDown
IsExpandLeft
IsExpandRight
IsExpandUp
IsIndexMarkAble
IsLocal
IsLocked
IsMarginSameAsParent
IsNotCopyable
IsNotGroupable
IsNoUICommAllowed
IsOverridden
IsOverride
IsPageBreak
IsPartOfGroup
IsPrintable
IsProtected
IsScrollable
IsSingleClickEntry
IsSizable
IsSnapTo
IsStyle
IsTableHeading
IsTOC
IsTOCMarkAble
IsValid
Join AS Join class
Justifiable
Layer AS Layout class
LayerName
LeaderDotType
LeftExternalMargin
LeftPage AS Layout class
LeftTopCellId
LineLocation
LinkFrame
MaintainAspectRatio
MarginBottom
MarginLeft
MarginRight
MarginTop
MasterName
MinBottomMargin
MinHeight
MinLeftMargin

[MinRightMargin](#)
[MinTopMargin](#)
[Name](#)
[NameBasedOnStyle](#)
[NumberOfLines](#)
[NumCols](#)
[NumColsSpannedOneCell](#)
[NumericFormat](#) AS [NumericFormat class](#)
[NumRowsSpannedOneCell](#)
[PageToUseLayoutOn](#)
[PaperName](#)
[Parent](#) AS [BaseObject class](#)
[RelativeType](#)
[RelativeXDistance](#)
[RelativeYDistance](#)
[RevisionType](#)
[RightExternalMargin](#)
[RightPage](#) AS [Layout class](#)
[ScaleHeight](#)
[ScaleMode](#)
[ScalePercentage](#)
[ScaleWidth](#)
[SelectType](#)
[Shadow](#) AS [Shadow class](#)
[Span](#)
[Style](#) AS [Layout class](#)
[StyleExceptions](#)
[TabRack](#) AS [TabRack class](#)
[TextOrient](#)
[Tile](#)
[TopExternalMargin](#)
[TopLeftCellRowId](#)
[UseFooter](#)
[UseHeader](#)
[UsePrinterSettings](#)
[UseWhen](#) AS [UseWhen class](#)
[VersionID](#)
[VertAlign](#)
[WasDeletedInRevMarkMode](#)
[WasInsertedInRevMarkMode](#)
[Where](#)
[Width](#)
[WPDataSets](#) AS [WPDataSetCollection class](#)
[WrapType](#)
[XOffset](#)
[XPosition](#)
[YOffset](#)

YPosition

Methods

AddChildToLayout

Backward

CreateLayer

DeleteContents

DeleteLayout

DoesMarkerNameMatch

FindClass

Forward

GetMarkerName

GetNamedProperty

GoToLayout

HasNamedProperty

ImportWatermarkGraphic

Mark

MirrorPage

MoveToBack

MoveToFront

RegisterWPDataSet

RemoveChildFromLayout

RemoveNamedProperty

RevisionAcceptLayoutChange

RevisionCancelLayoutChange

SetAllMargins

SetMinimumOrigin

SetNamedProperty

SetPaperNameAndUpdateSize

SetupForCropping

UnregisterWPDataSet

Update

Events

EnterLayout

KeyStroke

MouseDown

MouseUp

Word Pro: FootnoteCollection class members

Properties

Application AS WPAApplication class

Count

Description

IsValid

Name

Parent AS BaseObject class

VersionID

Methods

IsEmpty

Item

Events

None

Word Pro: FootnoteContSep class members

Properties

Application AS WPAApplication class

BorderLines AS BorderLines class

CustomLength

Description

IndentFromLeft

IsFixedLength

IsValid

Name

Parent AS BaseObject class

SpaceAbove

SpaceBelow

UseSeparatorLine

VersionID

Methods

None

Events

None

Word Pro: FootnoteLayoutCollection class members

Properties

Application AS WPAApplication class

Count

Description

IsValid

Name

Parent AS BaseObject class

VersionID

Methods

IsEmpty

Item

Events

None

Word Pro: FootnoteLayout class members

Properties

[AbsoluteOn](#)

[AbsoluteXPos](#)

[AbsoluteYPos](#)

[Accelerator](#)

[AllowResizeWhenCrop](#)

[AmtTether](#)

[AmtToRotateContent](#)

[AmtToTetherFrom](#)

[Application](#) AS [WPAApplication class](#)

[Background](#) AS [Background class](#)

[BaseLineOffset](#)

[BinName](#)

[BorderLines](#) AS [BorderLines class](#)

[BorderOffset](#)

[BottomExternalMargin](#)

[Center](#)

[ChildLayouts](#) AS [LayoutCollection class](#)

[ClassName](#)

[ColumnBalance](#)

[ColumnGap](#)

[ConditionType](#)

[Content](#)

[ContentName](#)

[ContentStyleName](#)

[Definition](#)

[Description](#)

[DirectionDown](#)

[DirectionLeft](#)

[DirectionRight](#)

[DirectionUp](#)

[DropCapPosition](#)

[EditorName](#)

[Footer](#) AS [Layout class](#)

[GridDistance](#)

[GridType](#)

[Gutter](#) AS [Gutter class](#)

[Header](#) AS [Layout class](#)

[Height](#)

[IsBreakable](#)

[IsChildSpannable](#)

[IsCollapsed](#)

[IsCollapsible](#)

[IsColumnBreakable](#)

[IsComplex](#)

IsConnected
IsErrorChecking
IsExpandDown
IsExpandLeft
IsExpandRight
IsExpandUp
IsIndexMarkAble
IsLocal
IsLocked
IsMarginSameAsParent
IsNotCopyable
IsNotGroupable
IsNoUICommAllowed
IsOverridden
IsOverride
IsPageBreak
IsPartOfGroup
IsPrintable
IsProtected
IsScrollable
IsSingleClickEntry
IsSizable
IsSnapTo
IsStyle
IsTableHeading
IsTOC
IsTOCMarkAble
IsValid
Join AS Join class
Justifiable
Layer AS Layout class
LayerName
LeaderDotType
LeftExternalMargin
LeftPage AS Layout class
LeftTopCellId
LineLocation
LinkFrame
MaintainAspectRatio
MarginBottom
MarginLeft
MarginRight
MarginTop
MasterName
MinBottomMargin
MinHeight
MinLeftMargin

[MinRightMargin](#)
[MinTopMargin](#)
[Name](#)
[NameBasedOnStyle](#)
[NumberOfLines](#)
[NumCols](#)
[NumColsSpannedOneCell](#)
[NumericFormat](#) AS [NumericFormat class](#)
[NumRowsSpannedOneCell](#)
[PageToUseLayoutOn](#)
[PaperName](#)
[Parent](#) AS [BaseObject class](#)
[RelativeType](#)
[RelativeXDistance](#)
[RelativeYDistance](#)
[RevisionType](#)
[RightExternalMargin](#)
[RightPage](#) AS [Layout class](#)
[ScaleHeight](#)
[ScaleMode](#)
[ScalePercentage](#)
[ScaleWidth](#)
[SelectType](#)
[Shadow](#) AS [Shadow class](#)
[Span](#)
[Style](#) AS [Layout class](#)
[StyleExceptions](#)
[TabRack](#) AS [TabRack class](#)
[TextOrient](#)
[Tile](#)
[TopExternalMargin](#)
[TopLeftCellRowId](#)
[UseFooter](#)
[UseHeader](#)
[UsePrinterSettings](#)
[UseWhen](#) AS [UseWhen class](#)
[VersionID](#)
[VertAlign](#)
[WasDeletedInRevMarkMode](#)
[WasInsertedInRevMarkMode](#)
[Where](#)
[Width](#)
[WPDataSets](#) AS [WPDataSetCollection class](#)
[WrapType](#)
[XOffset](#)
[XPosition](#)
[YOffset](#)

YPosition

Methods

AddChildToLayout

Backward

CreateLayer

DeleteContents

DeleteLayout

DoesMarkerNameMatch

FindClass

Forward

GetMarkerName

GetNamedProperty

GoToLayout

HasNamedProperty

ImportWatermarkGraphic

Mark

MirrorPage

MoveToBack

MoveToFront

RegisterWPDataSet

RemoveChildFromLayout

RemoveNamedProperty

RevisionAcceptLayoutChange

RevisionCancelLayoutChange

SetAllMargins

SetMinimumOrigin

SetNamedProperty

SetPaperNameAndUpdateSize

SetupForCropping

UnregisterWPDataSet

Update

Events

EnterLayout

KeyStroke

MouseDown

MouseUp

Word Pro: FootnoteNumbering class members

Properties

Application AS WPAApplication class

Description

IsValid

LeadingText

Name

Parent AS BaseObject class

ResetWhen

StartingNumber

TrailingText

UseSuperscriptReferenceNum

VersionID

Methods

None

Events

None

Word Pro: FootnoteNumOpt class members

Properties

Application AS WPAApplication class

Description

IsValid

LeadingText

Name

Parent AS BaseObject class

ResetWhen

StartingNumber

TrailingText

UseSuperscriptReferenceNum

VersionID

Methods

None

Events

None

Word Pro: FootnoteOptions class members

Properties

Application AS WPAApplication class

ContinuedFromAlignment

ContinuedFromMessage

ContinuedOnAlignment

ContinuedOnMessage

Description

EndnoteDivisionGroupNum AS EndnoteDivisionGroupNum class

EndnoteDivisionNum AS EndnoteDivisionNum class

EndnoteDocNum AS EndnoteDocNum class

FootnoteContSep AS FootnoteContSep class

FootnoteNumbering AS FootnoteNumbering class

FootnoteSeparator AS FootnoteSeparator class

IsContinuedFrom

IsContinuedOn

IsRepeat

IsValid

Name

Parent AS BaseObject class

VersionID

Methods

None

Events

None

Word Pro: FootnoteSeparator class members

Properties

Application AS WPAApplication class

BorderLines AS BorderLines class

CustomLength

Description

IndentFromLeft

IsFixedLength

IsValid

Name

Parent AS BaseObject class

SpaceAbove

SpaceBelow

UseSeparatorLine

VersionID

Methods

None

Events

None

Word Pro: FootnoteSepOpt class members

Properties

Application AS WPAApplication class

BorderLines AS BorderLines class

CustomLength

Description

IndentFromLeft

IsFixedLength

IsValid

Name

Parent AS BaseObject class

SpaceAbove

SpaceBelow

UseSeparatorLine

VersionID

Methods

None

Events

None

Word Pro: FootnoteTable class members

Properties

Application AS WPAApplication class

CanEmbed

CellLayouts AS StringCollection class

ColumnLayouts AS StringCollection class

ContentType

CurrentCell AS CellLayout class

CurrentColumn AS Layout class

CurrentRow AS RowLayout class

DefCellStyleName

DefColWidth

DefRowHeight

Description

EndingColOfSelection

EndingRowOfSelection

IsAutoGrow

IsEmpty

IsParagraphNumberingDown

IsReplaceable

IsResetParagraphNumber

IsSizingViaMouse

IsValid

Layout AS Layout class

MaxBottomBorder

MaxBottomGutter

MaxLeftBorder

MaxLeftGutter

MaxNumColsAllowed

MaxNumRowsAllowed

MaxRightBorder

MaxRightGutter

MaxSplitCols

MaxSplitRows

MaxTopBorder

MaxTopGutter

Name

NumCols

NumRows

Parent AS BaseObject class

RowLayouts AS StringCollection class

SelectionType

SingleCellSelected

StartingColOfSelection

StartingColStringOfSelection

StartingRowOfSelection

TableFill AS TableFill class

TableLine AS TableLine class

VersionID

Methods

CellLayout

Connect

Copy

DeleteTable

DisconnectCells

DoesMarkerNameMatch

FindCellLayout

GetMarkerName

GoToTableCell

InsertRowOrColumn

Mark

SelectTableItem

Split

Events

None

Word Pro: Footnote class members

Properties

Application AS WPAApplication class

Content

Description

IsValid

Name

Number

Parent AS BaseObject class

Type

VersionID

Methods

GoTo

Events

None

Word Pro: FormatCheckPref class members

Properties

Application AS WPAApplication class

Description

FixAcronymns

FixBullets

IsValid

Name

Options

Parent AS BaseObject class

Replacements

Typos

UseConsistentSpaceBetweenSentences

UseTwoSpacesBetweenSentences

VersionID

Methods

None

Events

None

Word Pro: Foundry class members

Properties

Application AS WPAApplication class

Bags AS BagCollection class

BaseTables AS TableCollection class

BookmarkManager AS BookmarkManager class

CellEngines AS CellCollection class

CellLayouts AS CellLayoutCollection class

CellLayoutStyles AS CellLayoutCollection class

CharacterStyles AS CharacterStyleCollection class

ClickHeres AS ClickHereCollection class

ConnectedLayouts AS ConnectedLayoutCollection class

Contents AS ContentCollection class

ContinuedFromStory

ContinuedOnStory

Description

DropCaps AS DropCapLayoutCollection class

DropCapStyles AS DropCapLayoutCollection class

Endnotes AS EndnoteLayoutCollection class

Enumeration

Footers AS FooterLayoutCollection class

FooterStyles AS FooterLayoutCollection class

FootnoteLayouts AS FootnoteLayoutCollection class

Footnotes AS FootnoteCollection class

Frames AS FrameLayoutCollection class

FrameStyles AS FrameLayoutCollection class

Glossarys AS GlossaryCollection class

Graphics AS GraphicCollection class

Groups AS GroupLayoutCollection class

Headers AS HeaderLayoutCollection class

HeaderStyles AS HeaderLayoutCollection class

IsUndoOn

IsValid

Layouts AS LayoutCollection class

Markers AS MarkerCollection class

Name

NoteLayouts AS NoteLayoutCollection class

OleObjects AS OleObjectCollection class

OutlineStyleSequences AS OutlineSeqCollection class

Pages AS PageLayoutCollection class

PageStyles AS PageLayoutCollection class

ParagraphStyles AS ParagraphStyleCollection class

ParallelColumns AS ParallelColsCollection class

Parent AS BaseObject class

PowerFields AS PowerFieldCollection class

Rows AS RowLayoutCollection class

[RubyLayouts](#) AS [RubyLayoutCollection](#) class
[Sections](#) AS [SectionCollection](#) class
[SilverBullets](#) AS [SilverBulletCollection](#) class
[SuperTableLayouts](#) AS [SuperTableLayoutCollection](#) class
[SuperTables](#) AS [SuperTableCollection](#) class
[TableHeadingLayouts](#) AS [TableHeadingLayoutCollection](#) class
[TableHeadings](#) AS [TableHeadingCollection](#) class
[TableLayouts](#) AS [TableLayoutCollection](#) class
[TableMarkers](#) AS [TableMarkerCollection](#) class
[Tables](#) AS [TableOnlyCollection](#) class
[TableStyles](#) AS [TableLayoutCollection](#) class
[TextMarkers](#) AS [TextMarkerCollection](#) class
[Texts](#) AS [TextCollection](#) class
[TextStyles](#) AS [TextStyleCollection](#) class
[VersionID](#)

Methods

[Clear](#)
[Copy](#)
[Create](#)
[DestroyPowerFields](#)
[Paste](#)
[Purge](#)
[Release](#)

Events

None

Word Pro: FrameCaptionOptions class members

Properties

Alignment

Application AS WPAApplication class

Description

InitialParaStyle

IsValid

Lines

Name

Parent AS BaseObject class

Position

SequenceName

SequenceNumber

Shadow

TextAfter

TextBefore

VersionID

Methods

None

Events

None

Word Pro: FrameContainer class members

Properties

[AbsoluteTextOrientation](#)
[Application](#) AS [WPAApplication class](#)
[ClientHeight](#)
[ClientWidth](#)
[ContentHeight](#)
[ContentName](#)
[ContentWidth](#)
[Description](#)
[DisplayablePageNum](#)
[DivisionInfo](#) AS [DivisionInfo class](#)
[DivisionName](#)
[Height](#)
[IsInOleDivision](#)
[IsValid](#)
[Layout](#) AS [Layout class](#)
[MaxContentHeight](#)
[MaxContentWidth](#)
[Name](#)
[NumContainers](#)
[PageNum](#)
[Parent](#) AS [BaseObject class](#)
[PositionXOnPage](#)
[PositionYOnPage](#)
[Presentation](#) AS [Presentation class](#)
[RelativePageNum](#)
[SectionName](#)
[TextOrientation](#)
[VersionID](#)
[Width](#)

Methods

[Abandon](#)
[AddContainer](#)
[Adopt](#)
[Anchor](#)
[Backward](#)
[ConnectContainer](#)
[DeleteContainer](#)
[Disconnect](#)
[DoesMultiCellPaste](#)
[EditWrap](#)
[Ending](#)
[Forward](#)
[GetObjectList](#)
[GetPasteFormatCategories](#)

GoToContainer

Hide

IsPointWithin

linkcontainers

RevertToStyle

SetStyle

ShowContainers

Start

unlinkcontainers

Events

None

Word Pro: FrameGroupLayout class members

Properties

[AbsoluteOn](#)

[AbsoluteXPos](#)

[AbsoluteYPos](#)

[Accelerator](#)

[AllowResizeWhenCrop](#)

[AmtTether](#)

[AmtToRotateContent](#)

[AmtToTetherFrom](#)

[Application](#) AS [WPAApplication class](#)

[Background](#) AS [Background class](#)

[BaseLineOffset](#)

[BinName](#)

[BorderLines](#) AS [BorderLines class](#)

[BorderOffset](#)

[BottomExternalMargin](#)

[Center](#)

[ChildLayouts](#) AS [LayoutCollection class](#)

[ClassName](#)

[ColumnBalance](#)

[ColumnGap](#)

[ConditionType](#)

[Content](#)

[ContentName](#)

[ContentStyleName](#)

[Definition](#)

[Description](#)

[DirectionDown](#)

[DirectionLeft](#)

[DirectionRight](#)

[DirectionUp](#)

[DropCapPosition](#)

[EditorName](#)

[Footer](#) AS [Layout class](#)

[GridDistance](#)

[GridType](#)

[Gutter](#) AS [Gutter class](#)

[Header](#) AS [Layout class](#)

[Height](#)

[IsBreakable](#)

[IsChildSpannable](#)

[IsCollapsed](#)

[IsCollapsible](#)

[IsColumnBreakable](#)

[IsComplex](#)

IsConnected
IsErrorChecking
IsExpandDown
IsExpandLeft
IsExpandRight
IsExpandUp
IsIndexMarkAble
IsLocal
IsLocked
IsMarginSameAsParent
IsNotCopyable
IsNotGroupable
IsNoUICommAllowed
IsOverridden
IsOverride
IsPageBreak
IsPartOfGroup
IsPrintable
IsProtected
IsScrollable
IsSingleClickEntry
IsSizable
IsSnapTo
IsStyle
IsTableHeading
IsTOC
IsTOCMarkAble
IsValid
Join AS Join class
Justifiable
Layer AS Layout class
LayerName
LeaderDotType
LeftExternalMargin
LeftPage AS Layout class
LeftTopCellId
LineLocation
LinkFrame
MaintainAspectRatio
MarginBottom
MarginLeft
MarginRight
MarginTop
MasterName
MinBottomMargin
MinHeight
MinLeftMargin

[MinRightMargin](#)
[MinTopMargin](#)
[Name](#)
[NameBasedOnStyle](#)
[NumberOfLines](#)
[NumCols](#)
[NumColsSpannedOneCell](#)
[NumericFormat](#) AS [NumericFormat class](#)
[NumRowsSpannedOneCell](#)
[PageToUseLayoutOn](#)
[PaperName](#)
[Parent](#) AS [BaseObject class](#)
[RelativeType](#)
[RelativeXDistance](#)
[RelativeYDistance](#)
[RevisionType](#)
[RightExternalMargin](#)
[RightPage](#) AS [Layout class](#)
[ScaleHeight](#)
[ScaleMode](#)
[ScalePercentage](#)
[ScaleWidth](#)
[SelectType](#)
[Shadow](#) AS [Shadow class](#)
[Span](#)
[Style](#) AS [Layout class](#)
[StyleExceptions](#)
[TabRack](#) AS [TabRack class](#)
[TextOrient](#)
[Tile](#)
[TopExternalMargin](#)
[TopLeftCellRowId](#)
[UseFooter](#)
[UseHeader](#)
[UsePrinterSettings](#)
[UseWhen](#) AS [UseWhen class](#)
[VersionID](#)
[VertAlign](#)
[WasDeletedInRevMarkMode](#)
[WasInsertedInRevMarkMode](#)
[Where](#)
[Width](#)
[WPDataSets](#) AS [WPDataSetCollection class](#)
[WrapType](#)
[XOffset](#)
[XPosition](#)
[YOffset](#)

YPosition

Methods

AddChildToLayout

Backward

CreateLayer

DeleteContents

DeleteLayout

DoesMarkerNameMatch

FindClass

Forward

GetMarkerName

GetNamedProperty

GoToLayout

HasNamedProperty

ImportWatermarkGraphic

Mark

MirrorPage

MoveToBack

MoveToFront

RegisterWPDataSet

RemoveChildFromLayout

RemoveNamedProperty

RevisionAcceptLayoutChange

RevisionCancelLayoutChange

SetAllMargins

SetMinimumOrigin

SetNamedProperty

SetPaperNameAndUpdateSize

SetupForCropping

UnregisterWPDataSet

Update

Events

EnterLayout

KeyStroke

MouseDown

MouseUp

Word Pro: FrameLayoutCollection class members

Properties

Application AS WPAApplication class

Count

Description

IsValid

Name

Parent AS BaseObject class

VersionID

Methods

IsEmpty

Item

Events

None

Word Pro: FrameLayout class members

Properties

[AbsoluteOn](#)

[AbsoluteXPos](#)

[AbsoluteYPos](#)

[Accelerator](#)

[AllowResizeWhenCrop](#)

[AmtTether](#)

[AmtToRotateContent](#)

[AmtToTetherFrom](#)

[Application](#) AS [WPAApplication class](#)

[Background](#) AS [Background class](#)

[BaseLineOffset](#)

[BinName](#)

[BorderLines](#) AS [BorderLines class](#)

[BorderOffset](#)

[BottomExternalMargin](#)

[Center](#)

[ChildLayouts](#) AS [LayoutCollection class](#)

[ClassName](#)

[ColumnBalance](#)

[ColumnGap](#)

[ConditionType](#)

[Content](#)

[ContentName](#)

[ContentStyleName](#)

[Definition](#)

[Description](#)

[DirectionDown](#)

[DirectionLeft](#)

[DirectionRight](#)

[DirectionUp](#)

[DropCapPosition](#)

[EditorName](#)

[Footer](#) AS [Layout class](#)

[GridDistance](#)

[GridType](#)

[Gutter](#) AS [Gutter class](#)

[Header](#) AS [Layout class](#)

[Height](#)

[IsBreakable](#)

[IsChildSpannable](#)

[IsCollapsed](#)

[IsCollapsible](#)

[IsColumnBreakable](#)

[IsComplex](#)

IsConnected
IsErrorChecking
IsExpandDown
IsExpandLeft
IsExpandRight
IsExpandUp
IsIndexMarkAble
IsLocal
IsLocked
IsMarginSameAsParent
IsNotCopyable
IsNotGroupable
IsNoUICommAllowed
IsOverridden
IsOverride
IsPageBreak
IsPartOfGroup
IsPrintable
IsProtected
IsScrollable
IsSingleClickEntry
IsSizable
IsSnapTo
IsStyle
IsTableHeading
IsTOC
IsTOCMarkAble
IsValid
Join AS Join class
Justifiable
Layer AS Layout class
LayerName
LeaderDotType
LeftExternalMargin
LeftPage AS Layout class
LeftTopCellId
LineLocation
LinkFrame
MaintainAspectRatio
MarginBottom
MarginLeft
MarginRight
MarginTop
MasterName
MinBottomMargin
MinHeight
MinLeftMargin

[MinRightMargin](#)
[MinTopMargin](#)
[Name](#)
[NameBasedOnStyle](#)
[NumberOfLines](#)
[NumCols](#)
[NumColsSpannedOneCell](#)
[NumericFormat](#) AS [NumericFormat class](#)
[NumRowsSpannedOneCell](#)
[PageToUseLayoutOn](#)
[PaperName](#)
[Parent](#) AS [BaseObject class](#)
[RelativeType](#)
[RelativeXDistance](#)
[RelativeYDistance](#)
[RevisionType](#)
[RightExternalMargin](#)
[RightPage](#) AS [Layout class](#)
[ScaleHeight](#)
[ScaleMode](#)
[ScalePercentage](#)
[ScaleWidth](#)
[SelectType](#)
[Shadow](#) AS [Shadow class](#)
[Span](#)
[Style](#) AS [Layout class](#)
[StyleExceptions](#)
[TabRack](#) AS [TabRack class](#)
[TextOrient](#)
[Tile](#)
[TopExternalMargin](#)
[TopLeftCellRowId](#)
[UseFooter](#)
[UseHeader](#)
[UsePrinterSettings](#)
[UseWhen](#) AS [UseWhen class](#)
[VersionID](#)
[VertAlign](#)
[WasDeletedInRevMarkMode](#)
[WasInsertedInRevMarkMode](#)
[Where](#)
[Width](#)
[WPDataSets](#) AS [WPDataSetCollection class](#)
[WrapType](#)
[XOffset](#)
[XPosition](#)
[YOffset](#)

YPosition

Methods

AddChildToLayout

Backward

CreateLayer

DeleteContents

DeleteLayout

DoesMarkerNameMatch

FindClass

Forward

GetMarkerName

GetNamedProperty

GoToLayout

HasNamedProperty

ImportWatermarkGraphic

Mark

MirrorPage

MoveToBack

MoveToFront

RegisterWPDataSet

RemoveChildFromLayout

RemoveNamedProperty

RevisionAcceptLayoutChange

RevisionCancelLayoutChange

SetAllMargins

SetMinimumOrigin

SetNamedProperty

SetPaperNameAndUpdateSize

SetupForCropping

UnregisterWPDataSet

Update

Events

EnterLayout

KeyStroke

MouseDown

MouseUp

Word Pro: GlossaryCollection class members

Properties

Application AS WPAApplication class

Count

Description

IsValid

Name

Parent AS BaseObject class

VersionID

Methods

IsEmpty

Item

Events

None

Word Pro: Glossary class members

Properties

[Application](#) AS [WPAApplication class](#)
[CanEmbed](#)
[CellLayouts](#) AS [StringCollection class](#)
[ColumnLayouts](#) AS [StringCollection class](#)
[ContentType](#)
[CurrentCell](#) AS [CellLayout class](#)
[CurrentColumn](#) AS [Layout class](#)
[CurrentRow](#) AS [RowLayout class](#)
[DefaultLeftColumnStyleName](#)
[DefaultRightColumnStyleName](#)
[DefCellStyleName](#)
[DefColWidth](#)
[DefRowHeight](#)
[Description](#)
[EndingColOfSelection](#)
[EndingRowOfSelection](#)
[IsAutoGrow](#)
[IsEmpty](#)
[IsParagraphNumberingDown](#)
[IsReplaceable](#)
[IsResetParagraphNumber](#)
[IsSizingViaMouse](#)
[IsValid](#)
[Layout](#) AS [Layout class](#)
[MaxBottomBorder](#)
[MaxBottomGutter](#)
[MaxLeftBorder](#)
[MaxLeftGutter](#)
[MaxNumColsAllowed](#)
[MaxNumRowsAllowed](#)
[MaxRightBorder](#)
[MaxRightGutter](#)
[MaxSplitCols](#)
[MaxSplitRows](#)
[MaxTopBorder](#)
[MaxTopGutter](#)
[Name](#)
[NumCols](#)
[NumRows](#)
[Parent](#) AS [BaseObject class](#)
[RowLayouts](#) AS [StringCollection class](#)
[SelectionType](#)
[SingleCellSelected](#)
[StartingColOfSelection](#)

[StartingColStringOfSelection](#)

[StartingRowOfSelection](#)

[TableFill](#) AS [TableFill class](#)

[TableLine](#) AS [TableLine class](#)

[VersionID](#)

Methods

[CellLayout](#)

[Connect](#)

[Copy](#)

[CopyMeaning](#)

[DeleteTable](#)

[DisconnectCells](#)

[DoesMarkerNameMatch](#)

[EnumerateTerm](#)

[ExtractText](#)

[FindCellLayout](#)

[FindTerm](#)

[GetMarkerName](#)

[Glossarize](#)

[GoToTableCell](#)

[InsertRowOrColumn](#)

[Mark](#)

[SelectTableItem](#)

[Split](#)

Events

None

Word Pro: Grammar class members

Properties

Application AS WPAApplication class

ApplyAdjectivePos

ApplyAdjectNounPart

ApplyAgreementWithHereThere

ApplyAnglicisms

ApplyArchaicExpressions

ApplyArticleAgreement

ApplyBadComparatives

ApplyBadInflection

ApplyBadNoun

ApplyBadNounGender

ApplyBadPlural

ApplyBadPrepositions

ApplyBelgianExpression

ApplyBorrowedForeign

ApplyBureuaJargon

ApplyCalque

ApplyCapitalizationCheck

ApplyClauseErrors

ApplyCliches

ApplyColloquialExpression

ApplyCommonlyConfusedWords

ApplyCommonMisspell

ApplyComplexWords

ApplyConfusedEasy

ApplyConfusedEnglish

ApplyConfusedHard

ApplyConfusedMedium

ApplyConfusedVerb

ApplyConsecutiveNouns

ApplyContractions

ApplyDerogatory

ApplyDifferentPrep

ApplyDoubleNegative

ApplyDoublePlural

ApplyDoubleWordCheck

ApplyElision

ApplyEnglishDerived

ApplyEnglishWords

ApplyExotic

ApplyExtraPrepositionCheck

ApplyFalseFriend

ApplyFemaleOccupation

ApplyFixedExpression

[ApplyForeignWord](#)
[ApplyFormalTerms](#)
[ApplyFormatErrors](#)
[ApplyGallicisms](#)
[ApplyGenderExpressions](#)
[ApplyGermanisms](#)
[ApplyHomoGraphs](#)
[ApplyHomonyms](#)
[ApplyHomonymsEasy](#)
[ApplyHomonymsHard](#)
[ApplyHomoPhone1](#)
[ApplyHomoPhone2](#)
[ApplyHomoPhones](#)
[ApplyIncorrectPlural](#)
[ApplyInformalExpressions](#)
[ApplyJargonWords](#)
[ApplykSplitInfinitives](#)
[ApplyLowercaseAdjective](#)
[ApplyLowercaseColor](#)
[ApplyLowercaseNumbers](#)
[ApplyLowercasePhrases](#)
[ApplyLowercasePronouns](#)
[ApplyMassVsCount](#)
[ApplyMisspelledExpressions](#)
[ApplyMisspelledForeignExpressions](#)
[ApplyMisspelledItalian](#)
[ApplyMisspelledWords](#)
[ApplyMisusedWords](#)
[ApplyNonStandardExpression](#)
[ApplyNonStandardModifiers](#)
[ApplyNoudModifierOrderCheck](#)
[ApplyNounConsistency](#)
[ApplyNounPhraseAgree](#)
[ApplyNSAdjective](#)
[ApplyNSClause](#)
[ApplyNSCompare](#)
[ApplyNSContract](#)
[ApplyNSGeography](#)
[ApplyNSInflection](#)
[ApplyNSNegation](#)
[ApplyNSPrep](#)
[ApplyNSPronoun](#)
[ApplyNSSpell](#)
[ApplyNSUsage](#)
[ApplyNSVerbForm](#)
[ApplyOpenClosedSpelling](#)
[ApplyOpenUsage](#)

[ApplyOverusedPhrases](#)
[ApplyPassiveVerbErrors](#)
[ApplyPostClitAgree](#)
[ApplyPrepExpression](#)
[ApplyPrepositionalPhrases](#)
[ApplyPretentiousWords](#)
[ApplyPronounErrors](#)
[ApplyPunctuationErrors](#)
[ApplyRedundantExpressions](#)
[ApplyRegionalExpression](#)
[ApplyRelatedWord](#)
[ApplySensitiveExp](#)
[ApplySexistExpressions](#)
[ApplySpellStandard](#)
[ApplyStockPhrase](#)
[ApplyStyleParameters](#)
[ApplySubjectVerbAgreement](#)
[ApplySwedishGender](#)
[ApplySwedishNegation](#)
[ApplySwedishUsage](#)
[ApplyTrite](#)
[ApplyTwoGender](#)
[ApplyTypicalMisspell](#)
[ApplyUnGrammaticalExpressions](#)
[ApplyVagueQuantifiers](#)
[ApplyVerbGroupConsistency](#)
[ApplyWeakModifiers](#)
[ApplyWordChoice](#)
[ApplyWordCompoundingCheck](#)
[ApplyWordConfusion](#)
[ApplyWordForm](#)
[ApplyWordGender](#)
[ApplyWordParts](#)
[ApplyWordyPhraseCheck](#)
[Description](#)
[GrammarFormalityLevel](#)
[GrammarProofLevel](#)
[IsValid](#)
[MaxIdenticalConsecSentOpens](#)
[MaxIdenticalSentOpensWithin10](#)
[MaximumWordsinaSentence](#)
[Name](#)
[Parent AS BaseObject class](#)
[ShowStatistics](#)
[SpacesBetweenSentences](#)
[VersionID](#)

Methods

None

Events

None

Word Pro: GraphicCollection class members

Properties

Application AS WPAApplication class

Count

Description

IsValid

Name

Parent AS BaseObject class

VersionID

Methods

IsEmpty

Item

Events

None

Word Pro: GraphicOleObjectCollection class members

Properties

Application AS WPAApplication class

Count

Description

IsValid

Name

Parent AS BaseObject class

VersionID

Methods

IsEmpty

Item

Events

None

Word Pro: GraphicOleObject class members

Properties

[Alignment](#) AS [Alignment class](#)
[Application](#) AS [WPAApplication class](#)
[CanEmbed](#)
[ContentType](#)
[DataFormat](#)
[Description](#)
[ExternallyControlledUndo](#)
[Font](#) AS [Font class](#)
[GetAfidHelpInfo](#)
[GetUndoWhatDesc](#)
[Height](#)
[IsActive](#)
[IsChartLink](#)
[IsDraw](#)
[IsEmpty](#)
[IsEquation](#)
[IsExternalFile](#)
[IsGraphicalObject](#)
[IsLotusChart](#)
[IsReplaceable](#)
[IsScalable](#)
[IsValid](#)
[LayoutName](#)
[LinkDisplayName](#)
[LinkedFileName](#)
[MetafilePict](#)
[Name](#)
[OrigHeight](#)
[OrigWidth](#)
[Parent](#) AS [BaseObject class](#)
[PositionXOnPage](#)
[PositionYOnPage](#)
[RightMousePropId](#)
[RightMousePropText](#)
[Section](#)
[ServerFormat](#)
[Text](#) AS [Text class](#)
[VersionID](#)
[WatermarkName](#)
[Width](#)

Methods

[CreateFromBitmap](#)
[CreateFromMetafile](#)
[CreateNew](#)

DeleteContent

DoesMultiCellPaste

DoneWithRightMouseMenu

GetCopyFormatCategories

GetRightMouseMenu

ImageProcess

ImportPicture

Mark

SaveData

SaveSnapshot

Select

Undo

Events

None

Word Pro: Graphic class members

Properties

Afid AS [Chart class](#)

AfidClassName

Alignment AS [Alignment class](#)

Application AS [WPAApplication class](#)

AtBeginningOfStream

AtEndOfStream

AutomaticLink

CanEmbed

ContentType

DataFormat

Description

EqnFontHeight

ExternallyControlledUndo

font AS [Font class](#)

GetAfidHelpInfo

GetUndoWhatDesc

Height

InfoBoxSelectionText

IsActive

IsChartLink

IsDraw

IsEmpty

IsEquation

IsExternalFile

IsGraphicalObject

IsLotusChart

IsNotCopyImage

IsReplaceable

IsScalable

IsValid

IsWordProChart

LayoutName

LinkAvailable

LinkDisplayName

LinkedFileName

MetafilePict

Name

OrigHeight

OrigWidth

Parent AS [BaseObject class](#)

PositionXOnPage

PositionYOnPage

Prompt

RightMousePropId

[RightMousePropText](#)

[Section](#)

[ServerFormat](#)

[Text AS Text class](#)

[VersionID](#)

[WatermarkName](#)

[Width](#)

Methods

[BreakLink](#)

[ConvertToClass](#)

[CreateFromBitmap](#)

[CreateFromClipBrd](#)

[CreateFromDataObject](#)

[CreateFromMetafile](#)

[CreateNew](#)

[DataObjectGetData](#)

[DataObjectGetDataHere](#)

[DeleteContent](#)

[DoesMultiCellPaste](#)

[DoneWithRightContextMenu](#)

[EnumerateChartLinks](#)

[GetCopyFormatCategories](#)

[GetLinkDisplayNameFileLength](#)

[GetLinkName](#)

[GetLinkSourceName](#)

[GetPasteFormatCategories](#)

[GetRightContextMenu](#)

[GetSource](#)

[GetUserClassNameFull](#)

[GetUserClassNameShort](#)

[ImageProcess](#)

[ImportPicture](#)

[Mark](#)

[ProcessAccelKey](#)

[SaveData](#)

[SaveSnapshot](#)

[Select](#)

[SetLinkSource](#)

[Undo](#)

[UpdateLink](#)

Events

None

Word Pro: GroupLayoutCollection class members

Properties

Application AS WPAApplication class

Count

Description

IsValid

Name

Parent AS BaseObject class

VersionID

Methods

IsEmpty

Item

Events

None

Word Pro: GroupLayout class members

Properties

[AbsoluteOn](#)

[AbsoluteXPos](#)

[AbsoluteYPos](#)

[Accelerator](#)

[AllowResizeWhenCrop](#)

[AmtTether](#)

[AmtToRotateContent](#)

[AmtToTetherFrom](#)

[Application](#) AS [WPAApplication class](#)

[Background](#) AS [Background class](#)

[BaseLineOffset](#)

[BinName](#)

[BorderLines](#) AS [BorderLines class](#)

[BorderOffset](#)

[BottomExternalMargin](#)

[Center](#)

[ChildLayouts](#) AS [LayoutCollection class](#)

[ClassName](#)

[ColumnBalance](#)

[ColumnGap](#)

[ConditionType](#)

[Content](#)

[ContentName](#)

[ContentStyleName](#)

[Definition](#)

[Description](#)

[DirectionDown](#)

[DirectionLeft](#)

[DirectionRight](#)

[DirectionUp](#)

[DropCapPosition](#)

[EditorName](#)

[Footer](#) AS [Layout class](#)

[GridDistance](#)

[GridType](#)

[Gutter](#) AS [Gutter class](#)

[Header](#) AS [Layout class](#)

[Height](#)

[IsBreakable](#)

[IsChildSpannable](#)

[IsCollapsed](#)

[IsCollapsible](#)

[IsColumnBreakable](#)

[IsComplex](#)

IsConnected
IsErrorChecking
IsExpandDown
IsExpandLeft
IsExpandRight
IsExpandUp
IsIndexMarkAble
IsLocal
IsLocked
IsMarginSameAsParent
IsNotCopyable
IsNotGroupable
IsNoUICommAllowed
IsOverridden
IsOverride
IsPageBreak
IsPartOfGroup
IsPrintable
IsProtected
IsScrollable
IsSingleClickEntry
IsSizable
IsSnapTo
IsStyle
IsTableHeading
IsTOC
IsTOCMarkAble
IsValid
Join AS Join class
Justifiable
Layer AS Layout class
LayerName
LeaderDotType
LeftExternalMargin
LeftPage AS Layout class
LeftTopCellId
LineLocation
LinkFrame
MaintainAspectRatio
MarginBottom
MarginLeft
MarginRight
MarginTop
MasterName
MinBottomMargin
MinHeight
MinLeftMargin

[MinRightMargin](#)
[MinTopMargin](#)
[Name](#)
[NameBasedOnStyle](#)
[NumberOfLines](#)
[NumCols](#)
[NumColsSpannedOneCell](#)
[NumericFormat](#) AS [NumericFormat class](#)
[NumRowsSpannedOneCell](#)
[PageToUseLayoutOn](#)
[PaperName](#)
[Parent](#) AS [BaseObject class](#)
[RelativeType](#)
[RelativeXDistance](#)
[RelativeYDistance](#)
[RevisionType](#)
[RightExternalMargin](#)
[RightPage](#) AS [Layout class](#)
[ScaleHeight](#)
[ScaleMode](#)
[ScalePercentage](#)
[ScaleWidth](#)
[SelectType](#)
[Shadow](#) AS [Shadow class](#)
[Span](#)
[Style](#) AS [Layout class](#)
[StyleExceptions](#)
[TabRack](#) AS [TabRack class](#)
[TextOrient](#)
[Tile](#)
[TopExternalMargin](#)
[TopLeftCellRowId](#)
[UseFooter](#)
[UseHeader](#)
[UsePrinterSettings](#)
[UseWhen](#) AS [UseWhen class](#)
[VersionID](#)
[VertAlign](#)
[WasDeletedInRevMarkMode](#)
[WasInsertedInRevMarkMode](#)
[Where](#)
[Width](#)
[WPDataSets](#) AS [WPDataSetCollection class](#)
[WrapType](#)
[XOffset](#)
[XPosition](#)
[YOffset](#)

YPosition

Methods

AddChildToLayout

Backward

CreateLayer

DeleteContents

DeleteLayout

DoesMarkerNameMatch

FindClass

Forward

GetMarkerName

GetNamedProperty

GoToLayout

HasNamedProperty

ImportWatermarkGraphic

Mark

MirrorPage

MoveToBack

MoveToFront

RegisterWPDataSet

RemoveChildFromLayout

RemoveNamedProperty

RevisionAcceptLayoutChange

RevisionCancelLayoutChange

SetAllMargins

SetMinimumOrigin

SetNamedProperty

SetPaperNameAndUpdateSize

SetupForCropping

UnregisterWPDataSet

Update

Events

EnterLayout

KeyStroke

MouseDown

MouseUp

Word Pro: Gutter class members

Properties

AllBorders AS Border class
Application AS WPAApplication class
BottomBorder AS Border class
Description
IsValid
LeftBorder AS Border class
LinePlacement
LineValid
Name
Parent AS BaseObject class
RightBorder AS Border class
TopBorder AS Border class
VersionID

Methods

None

Events

None

Word Pro: HeaderLayoutCollection class members

Properties

Application AS WPAApplication class

Count

Description

IsValid

Name

Parent AS BaseObject class

VersionID

Methods

IsEmpty

Item

Events

None

Word Pro: HeaderLayout class members

Properties

[AbsoluteOn](#)

[AbsoluteXPos](#)

[AbsoluteYPos](#)

[Accelerator](#)

[AllowResizeWhenCrop](#)

[AmtTether](#)

[AmtToRotateContent](#)

[AmtToTetherFrom](#)

[Application](#) AS [WPAApplication class](#)

[Background](#) AS [Background class](#)

[BaseLineOffset](#)

[BinName](#)

[BorderLines](#) AS [BorderLines class](#)

[BorderOffset](#)

[BottomExternalMargin](#)

[Center](#)

[ChildLayouts](#) AS [LayoutCollection class](#)

[ClassName](#)

[ColumnBalance](#)

[ColumnGap](#)

[ConditionType](#)

[Content](#)

[ContentName](#)

[ContentStyleName](#)

[Definition](#)

[Description](#)

[DirectionDown](#)

[DirectionLeft](#)

[DirectionRight](#)

[DirectionUp](#)

[DropCapPosition](#)

[EditorName](#)

[Footer](#) AS [Layout class](#)

[GridDistance](#)

[GridType](#)

[Gutter](#) AS [Gutter class](#)

[Header](#) AS [Layout class](#)

[Height](#)

[IsBreakable](#)

[IsChildSpannable](#)

[IsCollapsed](#)

[IsCollapsible](#)

[IsColumnBreakable](#)

[IsComplex](#)

IsConnected
IsErrorChecking
IsExpandDown
IsExpandLeft
IsExpandRight
IsExpandUp
IsIndexMarkAble
IsLocal
IsLocked
IsMarginSameAsParent
IsNotCopyable
IsNotGroupable
IsNoUICommAllowed
IsOverridden
IsOverride
IsPageBreak
IsPartOfGroup
IsPrintable
IsProtected
IsScrollable
IsSingleClickEntry
IsSizable
IsSnapTo
IsStyle
IsTableHeading
IsTOC
IsTOCMarkAble
IsValid
Join AS Join class
Justifiable
Layer AS Layout class
LayerName
LeaderDotType
LeftExternalMargin
LeftPage AS Layout class
LeftTopCellId
LineLocation
LinkFrame
MaintainAspectRatio
MarginBottom
MarginLeft
MarginRight
MarginTop
MasterName
MinBottomMargin
MinHeight
MinLeftMargin

[MinRightMargin](#)
[MinTopMargin](#)
[Name](#)
[NameBasedOnStyle](#)
[NumberOfLines](#)
[NumCols](#)
[NumColsSpannedOneCell](#)
[NumericFormat](#) AS [NumericFormat class](#)
[NumRowsSpannedOneCell](#)
[PageToUseLayoutOn](#)
[PaperName](#)
[Parent](#) AS [BaseObject class](#)
[RelativeType](#)
[RelativeXDistance](#)
[RelativeYDistance](#)
[RevisionType](#)
[RightExternalMargin](#)
[RightPage](#) AS [Layout class](#)
[ScaleHeight](#)
[ScaleMode](#)
[ScalePercentage](#)
[ScaleWidth](#)
[SelectType](#)
[Shadow](#) AS [Shadow class](#)
[Span](#)
[Style](#) AS [Layout class](#)
[StyleExceptions](#)
[TabRack](#) AS [TabRack class](#)
[TextOrient](#)
[Tile](#)
[TopExternalMargin](#)
[TopLeftCellRowId](#)
[UseFooter](#)
[UseHeader](#)
[UsePrinterSettings](#)
[UseWhen](#) AS [UseWhen class](#)
[VersionID](#)
[VertAlign](#)
[WasDeletedInRevMarkMode](#)
[WasInsertedInRevMarkMode](#)
[Where](#)
[Width](#)
[WPDataSets](#) AS [WPDataSetCollection class](#)
[WrapType](#)
[XOffset](#)
[XPosition](#)
[YOffset](#)

YPosition

Methods

AddChildToLayout

Backward

CreateLayer

DeleteContents

DeleteLayout

DoesMarkerNameMatch

FindClass

Forward

GetMarkerName

GetNamedProperty

GoToLayout

HasNamedProperty

ImportWatermarkGraphic

Mark

MirrorPage

MoveToBack

MoveToFront

RegisterWPDataSet

RemoveChildFromLayout

RemoveNamedProperty

RevisionAcceptLayoutChange

RevisionCancelLayoutChange

SetAllMargins

SetMinimumOrigin

SetNamedProperty

SetPaperNameAndUpdateSize

SetupForCropping

UnregisterWPDataSet

Update

Events

EnterLayout

KeyStroke

MouseDown

MouseUp

Word Pro: HTMLOptions class members

Properties

Application AS WPAApplication class

Description

DownLoadGraphics

EntireDocument

FontFaceSupport

GraphicType

ImportAsSource

IncludeFooter

IncludeHeader

IncludeURL

IsValid

Name

NavigationLink

Parent AS BaseObject class

Recursive

SaveDivisions

SavePath

SaveSections

TableOfContents

UnknownTagsInCommentNotes

URLAddress

URLDescription

VersionID

Methods

None

Events

None

Word Pro: HyphenationOptions class members

Properties

Application AS WPAApplication class

AutoHyphenate

Description

HyphenateLastWordInColumnOrPage

HyphenateLastWordInPara

HyphZoneAfter

HyphZoneBefore

IgnoreSoftHyphens

IsValid

MaxHyphLines

Name

Parent AS BaseObject class

VersionID

Methods

None

Events

None

Word Pro: IconBarCollection class members

Properties

Application AS WPAApplication class

Count

Description

IsValid

Name

Parent AS BaseObject class

VersionID

Methods

IsEmpty

Item

Events

None

Word Pro: IconBarManager class members

Properties

Application AS WPAApplication class

AreDisabledIconsGrayed

Description

IconBars AS IconBarCollection class

IconBarSets AS StringCollection class

IconHelpText

IconScript

IconSize

IsActionOnButtonDown

IsBubbleHelp

IsIconDepressible

IsShowing

IsValid

Name

Parent AS BaseObject class

VersionID

Methods

Configure

HidelconBars

SelectCustomIcon

SelectStandardIcon

ShowIconBars

Events

None

Word Pro: IconBar class members

Properties

Application AS WPAApplication class

Description

Height

IconBarPositionState

IconSetName

ID

IsEnabled

IsGrabBar

IsSame

IsSave

IsScalableBorder

IsShowing

IsValid

Name

Parent AS BaseObject class

PositionType

ScreenPositionX

ScreenPositionY

SetContextOfBar

ShowInContext

VersionID

Width

Methods

AddIcon

Configure

HideIconBar

Show

Events

None

Word Pro: Indent class members

Properties

All

Application AS WPAApplication class

BodyOnly

Description

First

Hang

IsBothSidesEqual

IsValid

Name

Parent AS BaseObject class

Relative

Rest

Right

UseRelative

VersionID

Methods

RevertToStyle

Events

None

Word Pro: IndexSection class members

Properties

Application AS WPAApplication class

Color AS Color class

Description

IndexAlphabeticSeparator

IndexDivision

IndexIndentType

IndexParent

IndexRange

IndexSection

IndexSource

IsIndex

IsValid

LayoutName

Name

NextSection

Parent AS BaseObject class

ShowTabs

UserName

VersionID

Methods

DeleteSection

GoToSection

Events

None

Word Pro: Join class members

Properties

Application AS WPAApplication class

Description

IsValid

JoinCorners

JoinHeight

JoinName

JoinType

JoinWidth

Name

Parent AS BaseObject class

Percentage

ScaleMode

VersionID

Methods

Clear

Events

None

Word Pro: Kinsoku class members

Properties

Application AS WPAApplication class

Description

IsEnabled

IsHangover

IsSqueeze

IsValid

Level

Name

Parent AS BaseObject class

VersionID

Methods

RevertToStyle

Events

None

Word Pro: Language class members

Properties

Application AS WPAApplication class

Description

IsValid

Language

Name

Parent AS BaseObject class

VersionID

Methods

RevertToStyle

Events

None

Word Pro: LayoutCollection class members

Properties

Application AS WPAApplication class

Count

Description

IsValid

Name

Parent AS BaseObject class

VersionID

Methods

IsEmpty

Item

Events

None

Word Pro: LayoutOverride class members

Properties

Application AS WPAApplication class

Description

IsValid

Name

Parent AS BaseObject class

VersionID

Methods

AddLayoutOverride

Events

None

Word Pro: Layout class members

Properties

[AbsoluteOn](#)

[AbsoluteXPos](#)

[AbsoluteYPos](#)

[Accelerator](#)

[AllowResizeWhenCrop](#)

[AmtTether](#)

[AmtToRotateContent](#)

[AmtToTetherFrom](#)

[Application](#) AS [WPAApplication class](#)

[Background](#) AS [Background class](#)

[BaseLineOffset](#)

[BinName](#)

[BorderLines](#) AS [BorderLines class](#)

[BorderOffset](#)

[BottomExternalMargin](#)

[Center](#)

[ChildLayouts](#) AS [LayoutCollection class](#)

[ClassName](#)

[ColumnBalance](#)

[ColumnGap](#)

[ConditionType](#)

[Content](#)

[ContentName](#)

[ContentStyleName](#)

[Definition](#)

[Description](#)

[DirectionDown](#)

[DirectionLeft](#)

[DirectionRight](#)

[DirectionUp](#)

[DropCapPosition](#)

[EditorName](#)

[Footer](#) AS [Layout class](#)

[GridDistance](#)

[GridType](#)

[Gutter](#) AS [Gutter class](#)

[Header](#) AS [Layout class](#)

[Height](#)

[IsBreakable](#)

[IsChildSpannable](#)

[IsCollapsed](#)

[IsCollapsible](#)

[IsColumnBreakable](#)

[IsComplex](#)

IsConnected
IsErrorChecking
IsExpandDown
IsExpandLeft
IsExpandRight
IsExpandUp
IsIndexMarkAble
IsLocal
IsLocked
IsMarginSameAsParent
IsNotCopyable
IsNotGroupable
IsNoUICommAllowed
IsOverridden
IsOverride
IsPageBreak
IsPartOfGroup
IsPrintable
IsProtected
IsScrollable
IsSingleClickEntry
IsSizable
IsSnapTo
IsStyle
IsTableHeading
IsTOC
IsTOCMarkAble
IsValid
Join AS Join class
Justifiable
Layer AS Layout class
LayerName
LeaderDotType
LeftExternalMargin
LeftPage AS Layout class
LeftTopCellId
LineLocation
LinkFrame
MaintainAspectRatio
MarginBottom
MarginLeft
MarginRight
MarginTop
MasterName
MinBottomMargin
MinHeight
MinLeftMargin

[MinRightMargin](#)
[MinTopMargin](#)
[Name](#)
[NameBasedOnStyle](#)
[NumberOfLines](#)
[NumCols](#)
[NumColsSpannedOneCell](#)
[NumericFormat](#) AS [NumericFormat class](#)
[NumRowsSpannedOneCell](#)
[PageToUseLayoutOn](#)
[PaperName](#)
[Parent](#) AS [BaseObject class](#)
[RelativeType](#)
[RelativeXDistance](#)
[RelativeYDistance](#)
[RevisionType](#)
[RightExternalMargin](#)
[RightPage](#) AS [Layout class](#)
[ScaleHeight](#)
[ScaleMode](#)
[ScalePercentage](#)
[ScaleWidth](#)
[SelectType](#)
[Shadow](#) AS [Shadow class](#)
[Span](#)
[Style](#) AS [Layout class](#)
[StyleExceptions](#)
[TabRack](#) AS [TabRack class](#)
[TextOrient](#)
[Tile](#)
[TopExternalMargin](#)
[TopLeftCellRowId](#)
[UseFooter](#)
[UseHeader](#)
[UsePrinterSettings](#)
[UseWhen](#) AS [UseWhen class](#)
[VersionID](#)
[VertAlign](#)
[WasDeletedInRevMarkMode](#)
[WasInsertedInRevMarkMode](#)
[Where](#)
[Width](#)
[WPDataSets](#) AS [WPDataSetCollection class](#)
[WrapType](#)
[XOffset](#)
[XPosition](#)
[YOffset](#)

YPosition

Methods

AddChildToLayout

Backward

CreateLayer

DeleteContents

DeleteLayout

DoesMarkerNameMatch

FindClass

Forward

GetMarkerName

GetNamedProperty

GoToLayout

HasNamedProperty

ImportWatermarkGraphic

Mark

MirrorPage

MoveToBack

MoveToFront

RegisterWPDataSet

RemoveChildFromLayout

RemoveNamedProperty

RevisionAcceptLayoutChange

RevisionCancelLayoutChange

SetAllMargins

SetMinimumOrigin

SetNamedProperty

SetPaperNameAndUpdateSize

SetupForCropping

UnregisterWPDataSet

Update

Events

EnterLayout

KeyStroke

MouseDown

MouseUp

Word Pro: LineNumberOptions class members

Properties

Application AS WPAApplication class

CountBlankLines

Description

DistanceFromLeftMargin

IsValid

LinesSpacedEveryNthUnit

Name

NumberEveryNthLine

NumberWhichLines

Parent AS BaseObject class

ResetOnEachPage

VersionID

Methods

None

Events

None

Word Pro: LWPTimer class members

Properties

Application AS WPAApplication class

Description

Enabled

Interval

IsValid

Name

Parent AS BaseObject class

VersionID

Methods

None

Events

TimerTick

Word Pro: Macro class members

Properties

Application AS WPAApplication class

DebugVariable

Description

IsDebug

IsValid

MacroName

MacroStatus

Name

Parent AS BaseObject class

ResumePausedMacro

VersionID

Methods

AppendMacro

Run

SaveMacro

Events

None

Word Pro: MailRouting class members

Properties

AboutToReturnToOriginator
Application AS WPAApplication class
CanEditProperty
Description
InitalizeRoute
Initialize
IsDocumentInRoute
IsLastStop
IsValid
MailRoutingPtr
Name
Parent AS BaseObject class
VersionID

Methods

None

Events

None

Word Pro: MarkerCollection class members

Properties

Application AS WPAApplication class

Count

Description

IsValid

Name

Parent AS BaseObject class

VersionID

Methods

IsEmpty

Item

Events

None

Word Pro: Marker class members

Properties

[Application](#) AS [WPAApplication class](#)
[Description](#)
[DivisionInfo](#) AS [DivisionInfo class](#)
[IsChanged](#)
[IsMarkerValid](#)
[IsRegistered](#)
[IsValid](#)
[Layout](#) AS [Layout class](#)
[MarkerClass](#)
[Name](#)
[NumCols](#)
[NumRows](#)
[PageNumber](#)
[PageNumberAsText](#)
[Parent](#) AS [BaseObject class](#)
[StartColumns](#)
[StartRow](#)
[StateID](#)
[VersionID](#)
[WPDataSets](#) AS [WPDataSetCollection class](#)

Methods

[DeleteContents](#)
[DeleteMarker](#)
[GetContents](#)
[GetMarkedText](#)
[GetNamedProperty](#)
[GetParagraphNumber](#)
[GetParagraphNumberString](#)
[GoTo](#)
[HasNamedProperty](#)
[InsertMarker](#)
[RegisterWPDataSet](#)
[RemoveNamedProperty](#)
[SetNamedProperty](#)
[UnregisterWPDataSet](#)

Events

None

Word Pro: MenuItemCollection class members

Properties

Application AS WPAApplication class

Count

Description

IsValid

Name

Parent AS BaseObject class

VersionID

Methods

IsEmpty

Item

Events

None

Word Pro: MenuItem class members

Properties

Action

Application AS WPAApplication class

Caption

Checked

Description

Enabled

ID

IsValid

Items AS MenuItemCollection class

Name

OverrideCheckAndEnable

Parent AS BaseObject class

ParentMenuHWND

VersionID

Methods

CopyItem

DeleteItem

DeleteItemByPosition

ExchangeItem

MoveItem

NewItem

NewItemByPosition

Events

None

Word Pro: MergeOptions class members

Properties

Application AS WPAApplication class

DataFileFieldNames

DataFileName

Description

DescriptionFileName

FilterName

IsAmiProTableImport

IsValid

MergeFileType

MergeInfoPtr

MergeStepNumber

MergeToFile

Name

NumberOfDataFields

NumberOfMergeConditions

Options

Parent AS BaseObject class

VersionID

Methods

AddACondition

Clear

RemoveDataFile

SetNoFields

Events

None

Word Pro: NoteContainer class members

Properties

AbsoluteTextOrientation
Application AS WPAApplication class
ClientHeight
ClientWidth
ContentHeight
ContentName
ContentWidth
Description
DisplayablePageNum
DivisionInfo AS DivisionInfo class
DivisionName
Height
IsInOleDivision
IsValid
Layout AS Layout class
MaxContentHeight
MaxContentWidth
Name
NumContainers
PageNum
Parent AS BaseObject class
PositionXOnPage
PositionYOnPage
Presentation AS Presentation class
RelativePageNum
SectionName
TextOrientation
VersionID
Width

Methods

Abandon
AddContainer
Adopt
Anchor
Backward
ConnectContainer
DeleteContainer
Disconnect
DoesMultiCellPaste
EditWrap
Ending
Forward
GetObjectList
GetPasteFormatCategories

GoToContainer

Hide

IsPointWithin

linkcontainers

RevertToStyle

SetStyle

ShowContainers

Start

unlinkcontainers

Events

None

Word Pro: NoteLayoutCollection class members

Properties

Application AS WPAApplication class

Count

Description

IsValid

Name

Parent AS BaseObject class

VersionID

Methods

IsEmpty

Item

Events

None

Word Pro: NoteLayout class members

Properties

[AbsoluteOn](#)

[AbsoluteXPos](#)

[AbsoluteYPos](#)

[Accelerator](#)

[AllowResizeWhenCrop](#)

[AmtTether](#)

[AmtToRotateContent](#)

[AmtToTetherFrom](#)

[Application](#) AS [WPAApplication class](#)

[Background](#) AS [Background class](#)

[BaseLineOffset](#)

[BinName](#)

[BorderLines](#) AS [BorderLines class](#)

[BorderOffset](#)

[BottomExternalMargin](#)

[Center](#)

[ChildLayouts](#) AS [LayoutCollection class](#)

[ClassName](#)

[Color](#) AS [Color class](#)

[ColumnBalance](#)

[ColumnGap](#)

[ConditionType](#)

[content](#)

[ContentName](#)

[ContentStyleName](#)

[Definition](#)

[Description](#)

[DirectionDown](#)

[DirectionLeft](#)

[DirectionRight](#)

[DirectionUp](#)

[DropCapPosition](#)

[EditorName](#)

[Footer](#) AS [Layout class](#)

[GridDistance](#)

[GridType](#)

[Gutter](#) AS [Gutter class](#)

[Header](#) AS [Layout class](#)

[Height](#)

[InitialsForFilters](#)

[IsBreakable](#)

[IsChildSpannable](#)

[IsCollapsed](#)

[IsCollapsible](#)

IsColumnBreakable
IsComplex
IsConnected
IsErrorChecking
IsExpandDown
IsExpandLeft
IsExpandRight
IsExpandUp
IsIndexMarkAble
IsLocal
IsLocked
IsMarginSameAsParent
IsNotCopyable
IsNotGroupable
IsNoUICommAllowed
IsOverridden
IsOverride
IsPageBreak
IsPartOfGroup
IsPrintable
IsProtected
IsScrollable
IsSingleClickEntry
IsSizable
IsSnapTo
IsStyle
IsTableHeading
IsTOC
IsTOCMarkAble
IsValid
Join AS Join class
Justifiable
Layer AS Layout class
LayerName
LeaderDotType
LeftExternalMargin
LeftPage AS Layout class
LeftTopCellId
LineLocation
LinkFrame
MaintainAspectRatio
MarginBottom
MarginLeft
MarginRight
MarginTop
MasterName
MinBottomMargin

[MinHeight](#)
[MinLeftMargin](#)
[MinRightMargin](#)
[MinTopMargin](#)
[Name](#)
[NameBasedOnStyle](#)
[NameForFilters](#)
[NumberOfLines](#)
[NumCols](#)
[NumColsSpannedOneCell](#)
[NumericFormat](#) AS [NumericFormat class](#)
[NumRowsSpannedOneCell](#)
[PageToUseLayoutOn](#)
[PaperName](#)
[Parent](#) AS [BaseObject class](#)
[RelativeType](#)
[RelativeXDistance](#)
[RelativeYDistance](#)
[RevisionType](#)
[RightExternalMargin](#)
[RightPage](#) AS [Layout class](#)
[ScaleHeight](#)
[ScaleMode](#)
[ScalePercentage](#)
[ScaleWidth](#)
[SelectType](#)
[Shadow](#) AS [Shadow class](#)
[Span](#)
[Style](#) AS [Layout class](#)
[StyleExceptions](#)
[TabRack](#) AS [TabRack class](#)
[Text](#) AS [Text class](#)
[TextOrient](#)
[Tile](#)
[Time](#)
[TopExternalMargin](#)
[TopLeftCellRowId](#)
[UseFooter](#)
[UseHeader](#)
[UsePrinterSettings](#)
[UseWhen](#) AS [UseWhen class](#)
[VersionID](#)
[VertAlign](#)
[WasDeletedInRevMarkMode](#)
[WasInsertedInRevMarkMode](#)
[Where](#)
[Width](#)

WPDataSets AS WPDataSetCollection class

WrapType

XOffset

XPosition

YOffset

YPosition

Methods

AddChildToLayout

Backward

createlayer

DeleteContents

DeleteLayout

DoesMarkerNameMatch

FindClass

Forward

GetMarkerName

GetNamedProperty

GoToLayout

HasNamedProperty

Hide

ImportWatermarkGraphic

Mark

MirrorPage

MoveToBack

MoveToFront

Open

registerwpdataset

RemoveChildFromLayout

RemoveNamedProperty

RevisionAcceptLayoutChange

RevisionCancelLayoutChange

setallmargins

SetMinimumOrigin

SetNamedProperty

SetPaperNameAndUpdateSize

SetupForCropping

UnregisterWPDataSet

Update

Events

EnterLayout

KeyStroke

MouseDown

MouseUp

Word Pro: NumberFormat class members

Properties

ListSelection

Parentheses

Precision

String

Type

Methods

SetNumericFormat

Events

None

Word Pro: Numbering class members

Properties

Application AS WPAApplication class

Description

Heading

IsValid

Level

Name

Parent AS BaseObject class

Position

SmartLevel

VersionID

Methods

RevertToStyle

Events

None

Word Pro: NumericFormatSubset class members

Properties

Application AS WPAApplication class

Color AS Color class

ColorOverride

Description

IsValid

Name

Parent AS BaseObject class

Prefix

Suffix

VersionID

Methods

None

Events

None

Word Pro: NumericFormat class members

Properties

AnyNumber AS NumericFormatSubset class

Application AS WPAApplication class

Description

FormatType AS Variant

IsValid

Name

Negative AS NumericFormatSubset class

NumDecimalPlaces

Parent AS BaseObject class

VersionID

Zero AS NumericFormatSubset class

Methods

Reset

Events

None

Word Pro: OleControl class members

Properties

WordProObjectName

Methods

None

Events

None

Word Pro: OleObjectCollection class members

Properties

Application AS WPAApplication class

Count

Description

IsValid

Name

Parent AS BaseObject class

VersionID

Methods

IsEmpty

Item

Events

None

Word Pro: OleObject class members

Properties

[Alignment](#) AS [Alignment class](#)
[Animated](#)
[Application](#) AS [WPAApplication class](#)
[AutomaticLink](#)
[CanEmbed](#)
[ClassId](#)
[ContentType](#)
[DataFormat](#)
[Description](#)
[DisplayAsIcon](#)
[EmitsEvents](#)
[ExternallyControlledUndo](#)
[FileFormat](#)
[font](#) AS [Font class](#)
[GetAfidHelpInfo](#)
[GetUndoWhatDesc](#)
[Height](#)
[IDispatch](#)
[InDocument](#)
[IsActive](#)
[IsChartLink](#)
[IsDraw](#)
[IsEmpty](#)
[IsEquation](#)
[IsExternalFile](#)
[IsGraphicalObject](#)
[IsLotusChart](#)
[IsReplaceable](#)
[IsScalable](#)
[IsValid](#)
[LayoutName](#)
[LinkAvailable](#)
[LinkDisplayName](#)
[LinkDisplayNameFileLength](#)
[Linked](#)
[LinkedFileName](#)
[LotusMiscStatus](#)
[MetafilePict](#)
[Name](#)
[Object](#)
[Ole1Object](#)
[OleObjectSize](#)
[OrigHeight](#)
[OrigWidth](#)

Parent AS BaseObject class

PositionXOnPage

PositionYOnPage

ProgID

RightMousePropId

RightMousePropText

Section

ServerFormat

Text AS Text class

TopLevelMenuText

UIActive

URLDownloadComplete

UserClassNameApp

UserClassNameFull

UserClassNameShort

VersionID

WatermarkName

Width

Methods

AbortURLDownload

ActivateAs

AddVerbMenu

BreakLink

ConvertTo

CreateFromBitmap

CreateFromMetafile

CreateNew

DeleteContent

DoesMultiCellPaste

DoneWithRightMouseMenu

GetAspectMetafilePict

GetCopyFormatCategories

GetRightMouseMenu

ImageProcess

ImportPicture

Mark

Open

SaveData

SaveSnapshot

Select

SetLinkSource

Undo

UpdateLink

Events

None

Word Pro: Options class members

Properties

Application AS WPAApplication class

CharacterSetName

ClickHerePrompts

Description

EmbedFonts

GrammarSetName

IncludeInitialsInNotes

IsValid

MarkCharacter

MarkPosition

MarkType

Name

PairKerning

Parent AS BaseObject class

PowerField

VersionID

WidowOrphan

Methods

None

Events

None

Word Pro: OutlineSeqCollection class members

Properties

Application AS WPAApplication class

Count

Description

IsValid

Name

Parent AS BaseObject class

VersionID

Methods

IsEmpty

Item

Events

None

Word Pro: OutlineSeqItemCollection class members

Properties

Application AS WPAApplication class

Count

Description

IsValid

Name

Parent AS BaseObject class

VersionID

Methods

IsEmpty

Item

Events

None

Word Pro: OutlineStyleSequence class members

Properties

Application AS WPAApplication class

Description

Heading

IsValid

Name

OutlineSeqItems AS OutlineSeqItemCollection class

Parent AS BaseObject class

VersionID

Methods

AddOutlineSequenceItem

Clear

Events

None

Word Pro: OutSeqItem class members

Properties

Application AS WPAApplication class

Description

IsValid

Name

Parent AS BaseObject class

Position

VersionID

Methods

None

Events

None

Word Pro: PageContainer class members

Properties

[AbsoluteTextOrientation](#)
[Application](#) AS [WPAApplication class](#)
[ClientHeight](#)
[ClientWidth](#)
[ContentHeight](#)
[ContentName](#)
[ContentWidth](#)
[Description](#)
[DisplayablePageNum](#)
[DivisionInfo](#) AS [DivisionInfo class](#)
[DivisionName](#)
[Height](#)
[IsFooter](#)
[IsHeader](#)
[IsInOleDivision](#)
[IsValid](#)
[Layout](#) AS [Layout class](#)
[MaxContentHeight](#)
[MaxContentWidth](#)
[Name](#)
[NumContainers](#)
[PageNum](#)
[Parent](#) AS [BaseObject class](#)
[PositionXOnPage](#)
[PositionYOnPage](#)
[Presentation](#) AS [Presentation class](#)
[RelativePageNum](#)
[SectionName](#)
[TextOrientation](#)
[VersionID](#)
[Width](#)

Methods

[Abandon](#)
[AddContainer](#)
[Adopt](#)
[Anchor](#)
[Backward](#)
[ConnectContainer](#)
[DeleteContainer](#)
[Disconnect](#)
[DoesMultiCellPaste](#)
[EditWrap](#)
[Ending](#)
[Forward](#)

GetObjectList

GetPasteFormatCategories

GoToContainer

Hide

IsPointWithin

linkcontainers

RevertToStyle

SetStyle

ShowContainers

Start

unlinkcontainers

Events

None

Word Pro: PageLayoutCollection class members

Properties

Application AS WPAApplication class

Count

Description

IsValid

Name

Parent AS BaseObject class

VersionID

Methods

IsEmpty

Item

Events

None

Word Pro: PageLayout class members

Properties

[AbsoluteOn](#)

[AbsoluteXPos](#)

[AbsoluteYPos](#)

[Accelerator](#)

[AllowResizeWhenCrop](#)

[AmtTether](#)

[AmtToRotateContent](#)

[AmtToTetherFrom](#)

[Application](#) AS [WPAApplication class](#)

[Background](#) AS [Background class](#)

[BaseLineOffset](#)

[BinName](#)

[BorderLines](#) AS [BorderLines class](#)

[BorderOffset](#)

[BottomExternalMargin](#)

[Center](#)

[ChildLayouts](#) AS [LayoutCollection class](#)

[ClassName](#)

[ColumnBalance](#)

[ColumnGap](#)

[ConditionType](#)

[content](#)

[ContentName](#)

[ContentStyleName](#)

[Definition](#)

[Description](#)

[DirectionDown](#)

[DirectionLeft](#)

[DirectionRight](#)

[DirectionUp](#)

[DropCapPosition](#)

[EditorName](#)

[Footer](#) AS [Layout class](#)

[GridDistance](#)

[GridType](#)

[Gutter](#) AS [Gutter class](#)

[Header](#) AS [Layout class](#)

[Height](#)

[IsBreakable](#)

[IsChildSpannable](#)

[IsCollapsed](#)

[IsCollapsible](#)

[IsColumnBreakable](#)

[IsComplex](#)

[IsConnected](#)
[IsErrorChecking](#)
[IsExpandDown](#)
[IsExpandLeft](#)
[IsExpandRight](#)
[IsExpandUp](#)
[IsIndexMarkAble](#)
[IsLocal](#)
[IsLocked](#)
[IsMarginSameAsParent](#)
[IsNotCopyable](#)
[IsNotGroupable](#)
[IsNoUICommAllowed](#)
[IsOverridden](#)
[IsOverride](#)
[IsPageBreak](#)
[IsPartOfGroup](#)
[IsPrintable](#)
[IsProtected](#)
[IsScrollable](#)
[IsSingleClickEntry](#)
[IsSizable](#)
[IsSnapTo](#)
[IsStyle](#)
[IsTableHeading](#)
[IsTOC](#)
[IsTOCMarkAble](#)
[IsValid](#)
[Join AS Join class](#)
[Justifiable](#)
[LandscapeMode](#)
[Layer AS Layout class](#)
[LayerName](#)
[LeaderDotType](#)
[LeftExternalMargin](#)
[LeftPage AS Layout class](#)
[LeftTopCellId](#)
[LineLocation](#)
[LinkFrame](#)
[MaintainAspectRatio](#)
[MarginBottom](#)
[MarginLeft](#)
[MarginRight](#)
[MarginTop](#)
[MasterName](#)
[MinBottomMargin](#)
[MinHeight](#)

[MinLeftMargin](#)
[MinRightMargin](#)
[MinTopMargin](#)
[Name](#)
[NameBasedOnStyle](#)
[NumberOfLines](#)
[NumCols](#)
[NumColsSpannedOneCell](#)
[NumericFormat](#) AS [NumericFormat class](#)
[NumRowsSpannedOneCell](#)
[PageToUseLayoutOn](#)
[PaperName](#)
[Parent](#) AS [BaseObject class](#)
[RelativeType](#)
[RelativeXDistance](#)
[RelativeYDistance](#)
[RevisionType](#)
[RightExternalMargin](#)
[RightPage](#) AS [Layout class](#)
[ScaleHeight](#)
[ScaleMode](#)
[ScalePercentage](#)
[ScaleWidth](#)
[SelectType](#)
[Shadow](#) AS [Shadow class](#)
[Span](#)
[Style](#) AS [Layout class](#)
[StyleExceptions](#)
[TabRack](#) AS [TabRack class](#)
[TextOrient](#)
[Tile](#)
[TopExternalMargin](#)
[TopLeftCellRowId](#)
[UseFooter](#)
[UseHeader](#)
[UsePrinterSettings](#)
[UseWhen](#) AS [UseWhen class](#)
[VersionID](#)
[VertAlign](#)
[WasDeletedInRevMarkMode](#)
[WasInsertedInRevMarkMode](#)
[Where](#)
[Width](#)
[WPDataSets](#) AS [WPDataSetCollection class](#)
[WrapType](#)
[XOffset](#)
[XPosition](#)

YOffset

YPosition

Methods

AddChildToLayout

Backward

createlayer

DeleteContents

DeleteLayout

DoesMarkerNameMatch

FindClass

Forward

GetMarkerName

GetNamedProperty

GoToLayout

HasNamedProperty

ImportWatermarkGraphic

Mark

MirrorPage

MoveToBack

MoveToFront

registerwpdataset

RemoveChildFromLayout

RemoveNamedProperty

RevisionAcceptLayoutChange

RevisionCancelLayoutChange

setallmargins

SetMinimumOrigin

SetNamedProperty

SetPaperNameAndUpdateSize

SetupForCropping

UnregisterWPDataSet

Update

Events

EnterLayout

KeyStroke

MouseDown

MouseUp

Word Pro: ParagraphBackground class members

Properties

Application AS WPAApplication class

Background AS Background class

Description

IsValid

Name

Parent AS BaseObject class

VersionID

Methods

RevertToStyle

Events

None

Word Pro: ParagraphBorder class members

Properties

[Application](#) AS [WPAApplication class](#)
[BetweenLines](#) AS [BetweenLines class](#)
[BorderLines](#) AS [BorderLines class](#)
[Description](#)
[IsBorder](#)
[IsValid](#)
[MarginBetween](#)
[MarginBottom](#)
[MarginLeft](#)
[MarginRight](#)
[MarginTop](#)
[Name](#)
[Parent](#) AS [BaseObject class](#)
[Shadow](#) AS [Shadow class](#)
[TypeAbove](#)
[TypeBelow](#)
[TypeBetween](#)
[TypeRight](#)
[VersionID](#)
[WidthAbove](#)
[WidthBelow](#)
[WidthBetween](#)

Methods

[RevertToStyle](#)

Events

None

Word Pro: ParagraphStyleCollection class members

Properties

Application AS WPAApplication class

Count

Description

IsValid

Name

Parent AS BaseObject class

VersionID

Methods

IsEmpty

Item

Events

None

Word Pro: ParagraphStyle class members

Properties

[Accelerator](#)

[Alignment](#) AS [Alignment class](#)

[AlignStyleName](#)

[Amikake](#) AS [Amikake class](#)

[AmikakeName](#)

[Application](#) AS [WPAApplication class](#)

[AttrStyleName](#)

[BorderStyleName](#)

[Breaks](#) AS [Breaks class](#)

[BreaksStyleName](#)

[Bullet](#) AS [Bullet class](#)

[BulletStyleName](#)

[CharacterBorder](#) AS [CharacterBorder class](#)

[CharacterBorderName](#)

[Definition](#)

[Description](#)

[DocumentLevel](#)

[FaceStyleName](#)

[Font](#) AS [Font class](#)

[FontStyleName](#)

[HasTabs](#)

[Heading](#)

[Indent](#) AS [Indent class](#)

[IndentStyleName](#)

[IsCumulative](#)

[IsLesser](#)

[IsLocal](#)

[IsPrivate](#)

[IsTemp](#)

[IsValid](#)

[Kinsoku](#) AS [Kinsoku class](#)

[Language](#) AS [Language class](#)

[Name](#)

[Numbering](#) AS [Numbering class](#)

[NumberingPosition](#)

[NumberingStyleName](#)

[ParagraphBackground](#) AS [ParagraphBackground class](#)

[ParagraphBorder](#) AS [ParagraphBorder class](#)

[Parent](#) AS [BaseObject class](#)

[RelativeIndent](#) AS [RelativeIndent class](#)

[SizeStyleName](#)

[Spacing](#) AS [Spacing class](#)

[SpacingStyleName](#)

[TabRack](#) AS [TabRack class](#)

TextAttributes AS Attributes class

Type

VersionID

WPDataSets AS WPDataSetCollection class

Methods

Clear

OutlineStyleSequence

RegisterWPDataSet

UnregisterWPDataSet

Update

Events

None

Word Pro: ParallelColsCollection class members

Properties

Application AS WPAApplication class

Count

Description

IsValid

Name

Parent AS BaseObject class

VersionID

Methods

IsEmpty

Item

Events

None

Word Pro: ParallelColsContainer class members

Properties

Application AS WPAApplication class

Description

IsValid

Name

Parent AS BaseObject class

VersionID

Methods

None

Events

None

Word Pro: ParallelColumns class members

Properties

Application AS WPAApplication class

CanEmbed

CellLayouts AS StringCollection class

ColumnLayouts AS StringCollection class

ContentType

CurrentCell AS CellLayout class

CurrentColumn AS Layout class

CurrentRow AS RowLayout class

DefaultLeftColumnStyleName

DefaultRightColumnStyleName

DefCellStyleName

DefColWidth

DefRowHeight

Description

EndingColOfSelection

EndingRowOfSelection

IsAutoGrow

IsEmpty

IsParagraphNumberingDown

IsReplaceable

IsResetParagraphNumber

IsSizingViaMouse

IsValid

Layout AS Layout class

MaxBottomBorder

MaxBottomGutter

MaxLeftBorder

MaxLeftGutter

MaxNumColsAllowed

MaxNumRowsAllowed

MaxRightBorder

MaxRightGutter

MaxSplitCols

MaxSplitRows

MaxTopBorder

MaxTopGutter

Name

NumCols

NumRows

Parent AS BaseObject class

RowLayouts AS StringCollection class

SelectionType

SingleCellSelected

StartingColOfSelection

StartingColStringOfSelection

StartingRowOfSelection

TableFill AS TableFill class

TableLine AS TableLine class

VersionID

Methods

CellLayout

Connect

Copy

DeleteTable

DisconnectCells

DoesMarkerNameMatch

FindCellLayout

GetMarkerName

GoToTableCell

InsertRowOrColumn

Mark

SelectTableItem

Split

Events

None

Word Pro: PowerFieldCollection class members

Properties

Application AS WPAApplication class

Count

Description

IsValid

Name

Parent AS BaseObject class

VersionID

Methods

IsEmpty

Item

Events

None

Word Pro: PowerField class members

Properties

[Application](#) AS [WPAApplication class](#)
[ConvertOnNew](#)
[Description](#)
[DivisionInfo](#) AS [DivisionInfo class](#)
[Formula](#)
[HideFormula](#)
[IsChanged](#)
[IsMarkerValid](#)
[IsRegistered](#)
[IsValid](#)
[Layout](#) AS [Layout class](#)
[LockResult](#)
[MarkerClass](#)
[Name](#)
[NumCols](#)
[NumRows](#)
[PageNumber](#)
[PageNumberAsText](#)
[Parent](#) AS [BaseObject class](#)
[Private](#)
[Result](#)
[StartColumns](#)
[StartRow](#)
[StateID](#)
[Type](#)
[UpdateOnLoad](#)
[UpdateOnLoadImmediate](#)
[VersionID](#)
[WPDataSets](#) AS [WPDataSetCollection class](#)

Methods

[DeleteContents](#)
[DeleteMarker](#)
[GetContents](#)
[GetMarkedText](#)
[GetNamedProperty](#)
[getparagraphnumber](#)
[GetParagraphNumberString](#)
[GoTo](#)
[HasNamedProperty](#)
[InsertMarker](#)
[registerwpdataset](#)
[RemoveNamedProperty](#)
[SetFieldFormula](#)
[SetNamedProperty](#)

UnregisterWPDataSet

Update

Events

None

Word Pro: Preferences class members

Properties

[AccessibilityOn](#)

[Address1](#)

[Address2](#)

[Application](#) AS [WPAApplication class](#)

[BackgroundPrintingOn](#)

[BackgroundSpellingOn](#)

[CellStyleName](#)

[ChangeKeyboardToLanguage](#)

[ChangeTextToMatchkeyboard](#)

[CharacterSet](#) AS [CharacterSet class](#)

[CityState](#)

[Company](#)

[DdeEnabled](#)

[DefaultCellStyleDescription](#)

[DefaultColumnName](#)

[DefaultDropCapStyleDescription](#)

[DefaultFrameStyleDescription](#)

[DefaultLatinFont](#)

[DefaultLeftColumnName](#)

[DefaultMenu](#)

[DefaultNonLatinFont](#)

[DefaultPageStyleDescription](#)

[DefaultRightColumnName](#)

[DefaultTableStyleDescription](#)

[DefaultTextStyleDescription](#)

[Description](#)

[DragDropOn](#)

[DropCapStyleName](#)

[EMail](#)

[FaxNumber](#)

[FontMatching](#)

[FontName](#)

[FrameStyleName](#)

[GIFAnimationOn](#)

[HiLiteColor](#) AS [Color class](#)

[IndexPrimaryStyleName](#)

[IndexSecondaryStyleName](#)

[IndexSeparatorStyleName](#)

[InsertionMode](#)

[IsSmartCorrectEnabled](#)

[IsSmartEditEnabled](#)

[IsSmartShadeEnabled](#)

[IsValid](#)

[LinkGraphic](#)

MultiCellPasteOn
Name
NumberSequenceName
NumUndoLevels
OLEEnabled
PageStyleName
Parent AS BaseObject class
PersonalData1
PersonalData2
PersonalData3
PersonalData4
PhoneNumber
PointSize
RevisionDisplay AS Revisiondisplay class
SaveSnapShot
SmallFileFormat
SmartLabelsEnabled
SnapshotSaveOptions AS StringCollection class
TableStyleName
TextStyleName
Title
UserInitials
UserName
VersionID
ZipCode

Methods

None

Events

None

Word Pro: Presentation class members

Properties

Application AS WPAApplication class

Description

IsValid

Name

OutlineLevel

Parent AS BaseObject class

VersionID

Methods

None

Events

None

Word Pro: PrintManager class members

Properties

Application AS WPAApplication class

BinNames AS StringCollection class

CanWePrint

CurrentlyPrintingInBackground

DefaultBinName

DefaultPageHeight

DefaultPageWidth

Description

DriverName

FaceNames AS StringCollection class

IsValid

Name

PaperNames AS StringCollection class

Parent AS BaseObject class

PrintDestination

PrinterName

QueueName

UseDefaultPrinter

VersionID

Methods

BinNameFromNumber

EndPrinting

GetEnvelopeDefaults

GetPrinterInfo

Print

ResetPrinting

UpdatePageSizeChange

UpdatePrinterBins

UpdatePrinterChanges

Events

None

Word Pro: PrintSettings class members

Properties

Application AS WPAApplication class

BookletPrinting

Collate

Copies

Crop

Description

Divisions AS StringCollection class

IsPrePrintedForm

IsPrintClickHereBlocks

IsUpdateIndex

IsUpdateTOC

IsValid

Name

OutputToFile

PageOrder

Parent AS BaseObject class

PrintGraphics

PrintPagesFrom

PrintPagesTo

PrintPageType

PrintRange

SelectedPages

UpdateFields

VersionID

ViewLevel

ViewType

WithComments

Methods

AddDivisionToPrint

ClearDivisionList

Events

None

Word Pro: ExpandOutlineLevel method

{button ,AL('H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL('H_EXPANDOUTLINELEVEL_METHOD_EXSCRIPT',1)} [See example](#)

Expands the highest level contracted heading(s) which are subordinate to the paragraph for which you are calling the method. For example, when you call this method for a Level 1 heading, it expands the highest level contracted heading(s) which are subordinate to that Level 1 heading.

Syntax

[objectreference].ExpandOutlineLevel([IsExpandAllLevels])

Parameters

IsExpandAllLevels

Allows you to expand all the subordinate headings under the heading from which you call this method. Data type is Integer, but the legal values for this parameter are -1 and 0. You may use the LotusScript constants True (-1) and False (0). A value of True causes all subordinate headings to be expanded, regardless of their level. When called from WPAplication, this parameter is optional. Default is True.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Equivalent to choosing Text - Outline - Expand. If the IsExpandAllLevels parameter is used, it is equivalent to Text - Outline - Expand All.

Word Pro: ExportAllAsNotesFX method

{button ,AL('H_DOCINFO_CLASS',0)} [See list of classes](#)

{button ,AL('H_EXPORTALLASNOTESFX_METHOD_EXSCRIPT',1)} [See example](#)

Exports all DocInfo fields in a document as Notes/FX.

Syntax

[objectreference].ExportAllAsNotesFX(Export)

Parameters

Export

An Integer value indicating whether or not to export all DocInfo fields in a document to Notes. You can use the LotusScript constants of True (-1) and False (0) for this parameter.

Return value

The return value for this method will always be -1 or 0. When testing the return value, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Calling this method with an Export parameter value of True is equivalent to choosing File - Document Properties - Document, clicking the Fields panel, and enabling "Export as Notes/FX field data" for all document fields.

Calling this method with an Export parameter value of False is equivalent to choosing File - Document Properties - Document, clicking the Fields panel, and disabling "Export as Notes/FX field data" for all document fields.

Word Pro: ExportAsNotesFX method

{button ,AL('H_DOCINFO_CLASS',0)} [See list of classes](#)

{button ,AL('H_EXPORTASNOTESFX_METHOD_EXSCRIPT',1)} [See example](#)

Allows you to determine whether a document field will be exported as Notes/FX

Syntax

[objectreference].ExportAsNotesFX(Type, Export)

Parameters

prexType

You can select and set any one of the DocInfo field data types below to export as Notes/FX. The value of this Variant parameter must be one of the strings below or its numeric equivalent.

<u>Value</u>	<u>Effect</u>
\$LwpDocVarAllversionnames (210)	The names of all versions of a document.
\$LwpDocVarCreatedby (2046)	The Word Pro user name of the person who first saved the document.
\$LwpDocVarDatecreated (196)	The date and time the document was first created.
\$LwpDocVarDatelastrevision (197)	The last time the document was opened, edited, and saved.
\$LwpDocVarDescription (195)	A description of the document.
\$LwpDocVarDivisionname (203)	The name of a division within the document.
\$LwpDocVarDoccoategory (212)	The assigned category of the division or document.
\$LwpDocVarDocsize (202)	The size of the document in kilobytes.
\$LwpDocVarField (191)	Specifies the DocInfo field you want to export as Notes/FX.
\$LwpDocVarFilename (192)	The file name you specified in the Save As dialog box when you saved the document.
\$LwpDocVarKeywords (215)	Displays the assigned keywords for the document.
\$LwpDocVarLasteditor (2047)	The initials of the editor who last saved the document.
\$LwpDocVarNone (190)	No DocInfo fields are set or reset as Notes/FX.
\$LwpDocVarNumchars (201)	The number of characters in the document.
\$LwpDocVarNumpages (199)	The number of pages in the document.
\$LwpDocVarNumversions (209)	The number of versions in the document.
\$LwpDocVarNumwords (200)	The number of words in the document.
\$LwpDocVarOthereditors (2048)	The Word Pro user names of any other people who saved the document.
\$LwpDocVarOtherversioneditors (207)	The Word Pro user names of any other people who saved a version of the document.
\$LwpDocVarPath (193)	The drive and folder where the document is located.
\$LwpDocVarSectionname (204)	The name of a section in the document.
\$LwpDocVarStylesheet (194)	The style sheet used in the document.
\$LwpDocVarTotaledittime (198)	The total number of minutes the document was open.
\$LwpDocVarVersioncreatedate (206)	The date and time the version was first created.
\$LwpDocVarVersioncreatedby (205)	The Word Pro user name of the person who first saved the version of the document.
\$LwpDocVarVersionlasteditdate (213)	The date and time that the version was last saved.

\$LwpDocVarVersionlasteditedby (214)	The initials of the editor who last saved the version.
\$LwpDocVarVersionname (208)	The name of the current version.
\$LwpDocVarVersionnumrevisions (2049)	The number of times the version was opened, edited, and saved.
\$LwpDocVarVersionremarks (211)	Any remarks written by an editor to be reviewed by any other assigned editors.

Export

The data type for this parameter is Integer. The legal values for this parameter are -1 and 0 but you may use the LotusScript constants True (-1) and False (0) instead of the integer values.

This parameter indicates whether or not to export a DocInfo field to Notes.

Return value

The return value for this method will always be -1 or 0. When testing the return value, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Calling this method with an Export parameter value of True is equivalent to choosing File - Document Properties - Document, clicking the Fields panel, and enabling "Export as Notes/FX field data" for the field specified in the Type parameter.

Calling this method with an Export parameter value of False is equivalent to choosing File - Document Properties - Document, clicking the Fields panel, and disabling "Export as Notes/FX field data" for the field specified in the Type parameter.

{button ,AL('H_EXPORTTONOTESFX_PROPERTY_MEMDEF',0)} [See related topics](#)

Word Pro: ExtractText method

{button ,AL('H_GLOSSARY_CLASS',0)} [See list of classes](#)

{button ,AL('H_EXTRACTTEXT_METHOD_EXSCRIPT',1)} [See example](#)

Extracts the text in the meaning of a term and changes that text to a null terminated string.

Syntax

[objectreference].Glossary.ExtractText(Term)

Parameters

Term

Specifies the string term for which you extract the meaning.

Return value

A String value that represents the meaning of the term specified in the Term parameter.

Usage

Word Pro: FindCellLayout method

{button ,AL(^H_BASETABLE_CLASS;H_FOOTNOTETABLE_CLASS;H_PARALLELCOLUMNS_CLASS;H_GLOSSARY_CLASS;H_TABLEHEADING_CLASS;H_TABLE_CLASS',0)} [See list of classes](#)

{button ,AL(^H_FINDCELLLAYOUT_METHOD_EXSCRIPT',1)} [See example](#)

Allows you to determine whether or not there is a layout object within the specified cell.

Syntax

[objectreference].FindCellLayout(Row, Column)

Parameters

Row

Data type is Integer. The row ID is zero-based, which means that the first row in a table has a zero ID value.

Column

Data type is Integer. The column ID is zero-based, which means that the first column in a table has a zero ID value.

Return value

The return value for this method will always be -1 or 0. When testing the return value, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values. A True return value indicates that there is a layout object within the specified cell. A False return value indicates that there is not a layout object within the specified cell.

Usage

Cells within a table do not always contain cell layout objects. In order to conserve system memory, cell layout objects are created only when they are required by Word Pro.

Word Pro: FindClass method

```
{button ,AL('H_LAYOUT_CLASS;H_CELLGROUPLAYOUT_CLASS;H_CELLLAYOUT_CLASS;H_CONNECTEDLAY  
OUT_CLASS;H_COLUMNGROUPLAYOUT_CLASS;H_FOOTERLAYOUT_CLASS;H_FRAMELAYOUT_CLASS;H  
_FRAMEGROUPLAYOUT_CLASS;H_DROPcapLAYOUT_CLASS;H_GROUPLAYOUT_CLASS;H_HEADERLAY  
OUT_CLASS;H_NOTELAYOUT_CLASS;H_PAGELAYOUT_CLASS;H_ROWGROUPLAYOUT_CLASS;H_ROWLA  
YOUT_CLASS;H_RUBYLAYOUT_CLASS;H_SUPERTABLEGROUPLAYOUT_CLASS;H_SUPERTABLELAYOUT  
_CLASS;H_TABLELAYOUT_CLASS;H_ENDNOTELAYOUT_CLASS;H_FOOTNOTELAYOUT_CLASS;H_TABLE  
HEADINGLAYOUT_CLASS;H_TOCSUPERTABLELAYOUT_CLASS',0)} See list of classes
```

```
{button ,AL('H_FINDCLASS_METHOD_EXSCRIPT',1)} See example
```

Finds the first child of a layout object of the class that you specify.

Syntax

```
[objectreference].FindClass(LayoutClassName)
```

Parameters

LayoutClassName

The name of the class that contains the layout you want to find. The name of the class must be one of the String values listed below:

- Frame
- DropCap
- SuperTable
- Viewport
- Footer
- TableHeading
- Footnote
- Endnote
- NoteFrame
- NoteHeader
- NoteText
- Cell
- FnCell
- FnContinueOn
- FnContinueFrom
- Page
- Table
- Connected
- Hidden
- Row
- Header
- GroupFrame

Return value

Returns a string value representing the name of the layout that matches the class you want to find.

Usage

The LayoutClass name parameter is case sensitive.

Word Pro: FindTerm method

{button ,AL('H_GLOSSARY_CLASS',0)} [See list of classes](#)

{button ,AL('H_FINDTERM_METHOD_EXSCRIPT',1)} [See example](#)

Determines whether or not a specified term exists in a Glossary object.

Syntax

[objectreference].FindTerm(Term)

Parameters

Term

Specifies the String term to be found and used as a glossary entry.

Return value

If the term specified in the Term parameter is found in the Glossary object, this method returns a value of 1. If the term is not found, this method returns a value of 0.

Usage

Word Pro: Find method

{button ,AL('H_BOOKMARKMANAGER_CLASS;H_CLICKHERE_CLASS;H_TEXTMARKER_CLASS;H_TEXT_CLA
SS',0)} [See list of classes](#)

{button ,AL('H_FIND_METHOD_EXSCRIPT',1)} [See example](#)

Executes a search for text, styles, or bookmarks. See Usage for details.

Syntax

When called from the WPAApplication object:

```
.Find()
```

When called from a Text, TextMarker, or ClickHere object:

```
[objectreference].Find([MarkerName,] [ReplaceAll,] [ UseTempOptions])
```

When called from a BookmarkManager object:

```
[objectreference].Find(FindName, Name)
```

Parameters

MarkerName

Data type is String. Required if no Name parameter. A bookmark parameter must have a Name or MarkerName, depending on the first parameter set.

ReplaceAll

Data type is Integer. The legal values for this parameter are -1 and 0 but you may use the LotusScript constants True (-1) and False (0). Optional parameter. Default is False (0).

UseTempOptions

Data type is Integer. The legal values for this parameter are -1 and 0 but you may use the LotusScript constants True (-1) and False (0). Optional parameter. Default is False (0).

FindName

Indicates whether you want to find a bookmark by its user-defined name or by the internal (hexadecimal) marker name. The value of this parameter must be one of the strings or codes listed below:

<u>Value</u>	<u>Effect</u>
\$LwpBookmarkFindNameBookmark (2054)	The internal (hexadecimal) name of the bookmark is provided in the Name parameter.
\$LwpBookmarkFindNameMarker (2055)	The user-defined name of the bookmark is provided in the Name parameter.

Name

Data type is String; contains the name. A bookmark parameter must have a Name or MarkerName, depending on the first parameter set. If the FindName parameter is \$LwpBookmarkFindNameBookmark, then the string should be the user-defined name. If the FindName parameter is \$LwpBookmarkFindNameMarker, then the string should be the internal (hexadecimal) name.

Return value

When called from WPAApplication, the return values are:

<u>Value</u>	<u>Effect</u>
0	Found a match and will show to user.
1	Found a match.
2	Match not found.
3	Match not found; reached end of all streams or shaded region.
4	Match not found; finished searching shaded range, will complete search of whole document.

When called from Text, TextMarker, or ClickHere, a successful find returns a value of 0 (True). If unsuccessful, returns a value of 2.

When called from a BookmarkManager object, the return value is a String representing the name of the bookmark found.

Usage

[WPAApplication]

Executes a Find based on the values set in the FindAndReplace object. No parameters are required when calling this method from WPAApplication.

[BookmarkManager]

Finds a specific bookmark by looking for the user-defined bookmark name or the internal (hexadecimal) marker name. If you supply the internal bookmark name, the method returns the user defined name. If you supply the user defined bookmark name, the method returns the internal name.

[Text, TextMarker, ClickHere]

Conducts a Find based on the current Find & Replace options. Equivalent to clicking the Find button in the Find and Replace bar.

[Text]

The Find method in the Text class can be used to initiate a Find, based on the current parameters in the FindAndReplace class off Application. You can generally use this method directly off the Application class, where the FindAndReplace class is located.

Note Do not execute this method off a text object without parameters and without first setting the parameters in the FindAndReplace class for the current Application object. If you do, it can cause unreliable results.

Word Pro: ForceDocToLoad method

{button ,AL('H_DIVISION_CLASS;H_TEXTDOCUMENT_CLASS',0)} [See list of classes](#)

Syntax

[objectreference].ForceDocToLoad()

Parameters

Return value

Usage

Word Pro: Forward method

```
{button ,AL('H_BASECONTAINER_CLASS;H_CELLCONTAINER_CLASS;H_FRAMECONTAINER_CLASS;H_NOTECONTAINER_CLASS;H_RUBYCONTAINER_CLASS;H_DROPCAPCONTAINER_CLASS;H_PAGECONTAINER_CLASS;H_SUBPAGECONTAINER_CLASS;H_SUPERPAGECONTAINER_CLASS;H_ROWCONTAINER_CLASS;H_SUPERTABLECONTAINER_CLASS;H_TABLECONTAINER_CLASS;H_PARALLELCOLSCONTAINER_CLASS;H_TABLEONLYCONT_CLASS;H_LAYOUT_CLASS;H_CELLGROUPLAYOUT_CLASS;H_CELLLAYOUT_CLASS;H_CONNECTEDLAYOUT_CLASS;H_COLUMNGROUPLAYOUT_CLASS;H_FOOTERLAYOUT_CLASS;H_FRAMELAYOUT_CLASS;H_FRAMEGROUPLAYOUT_CLASS;H_DROPCAPLAYOUT_CLASS;H_GROUPLAYOUT_CLASS;H_HEADERLAYOUT_CLASS;H_NOTELAYOUT_CLASS;H_PAGELAYOUT_CLASS;H_ROWGROUPLAYOUT_CLASS;H_ROWLAYOUT_CLASS;H_RUBYLAYOUT_CLASS;H_SUPERTABLEGROUPLAYOUT_CLASS;H_SUPERTABLELAYOUT_CLASS;H_TABLELAYOUT_CLASS;H_ENDNOTELAYOUT_CLASS;H_FOOTNOTELAYOUT_CLASS;H_TABLEHEADINGLAYOUT_CLASS;H_TOCSUPERTABLELAYOUT_CLASS;H_CLICKHERE_CLASSES;H_TEXTMARKER_CLASS;H_TEXT_CLASS',0)} See list of classes
```

```
{button ,AL('H_FORWARD_METHOD_EXSCRIPT',1)} See example
```

Moves an object or the insertion point forward. A Backward method is also available.

Syntax

When called from a Layout object:

```
[objectreference.]Forward()
```

When called from a container object:

```
[objectreference.]Forward(Direction)
```

When called from a Text, TextMarker, or ClickHere object:

```
[objectreference.]Forward(Unit, N[, Cursoring][, TextOnly])
```

Parameters

Direction

Specifies whether Word Pro should move the insertion point forward by page or by window. Data type is Variant, which allows the value of this parameter to be one of the constants listed below or its numeric equivalent (in parentheses). There is no default value.

<u>Value</u>	<u>Effect</u>
\$LwpDirectionPage (182)	Moves the insertion point forward by one page.
\$LwpDirectionWindow (183)	Moves the insertion point forward by one window.

This parameter is only used when calling this method from a container object. A container object is any object created from a container class. A container class is any class derived from the BaseContainer class including: CellContainer, DropCapContainer, FrameContainer, NoteContainer, PageContainer, ParallelColsContainer, RowContainer, RubyContainer, SubPageContainer, SuperPageContainer, SuperTableContainer, TableContainer, and TableOnlyCont.

Unit

Specifies the unit of measurement you want to use in moving the insertion point. Use this parameter only when calling this method from a Text, TextMarker, or ClickHere object. You must also use the N parameter to indicate how many of these units to move forward. Data type is Variant, which allows the value of this parameter to be one of the constants listed below or its numeric equivalent (in parentheses). There is no default value.

<u>Value</u>	<u>Effect</u>
\$LwpNavigateObjectTypeCharacter (1519)	Moves the insertion point the specified number of characters. Sets the Cursoring parameter to True to mimic the use of the arrow keys.
\$LwpNavigateObjectTypeChunk (1522)	Moves the insertion point the specified number of chunks. A chunk is comprised of a single word (a contiguous group of characters with no spaces) and all the spaces following that word. If the insertion point is at the beginning, the end or anywhere within a word, the chunk is comprised of that word and the spaces which follow it. If the insertion point is between two spaces, the chunk is seen as all the spaces following the insertion point to the beginning of the next word. If there is no

	word between the spaces and the end of the paragraph, the chunk is comprised of all the spaces up to the end of the paragraph.
\$LwpNavigateObjectTypeObject (1520)	Any of the objects defined in this list.
\$LwpNavigateObjectTypePage (1518)	Moves the insertion point the specified number of pages, leaving it at the top of the page.
\$LwpNavigateObjectTypeParagraph (1524)	Moves the insertion point the specified number of paragraphs. A paragraph is comprised of all the text and tables between two paragraph markers as well as any frames whose "Place frame" option is set to "With paragraph above."
\$LwpNavigateObjectTypeSentence (1523)	Moves the insertion point the specified number of sentences. A sentence is comprised of all the text between two periods.
\$LwpNavigateObjectTypeWord (1521)	Moves the insertion point the specified number of words. A word is comprised of a contiguous string of alphanumeric characters. Punctuation or a space is seen as the end of a word. If the insertion point is between two spaces, the word is comprised of all the spaces on both sides of the insertion point as well as the word preceding the spaces.

N

An Integer expression which specifies the number of units you want to move the insertion point. Use this parameter only when calling this method from a Text, TextMarker, or ClickHere object.

Cursoring

This optional Integer parameter is valid only when the Unit parameter has a value of \$LwpNavigateObjectTypeCharacter. This parameter takes an Integer expression which indicates whether or not you want Word Pro to move the insertion point as if you were using the arrow keys to move the cursor through a document. When you use the arrows keys, Word Pro skips over hidden markers such as bookmarks. The default value is False (0), which causes Word Pro to include any hidden markers when it moves the insertion point by characters. The legal values for this parameter are -1 and 0 but you may use the LotusScript constants True (-1) and False (0) instead of the integer values. Use this parameter only when calling this method from a Text, TextMarker, or ClickHere object.

TextOnly

An optional Integer expression which indicates whether you want Word Pro to ignore objects in the stream that are not text-like. Data type is Integer. The legal values for this parameter are -1 and 0 but you may use the LotusScript constants True (-1) and False (0) instead of the integer values. The default value is False (0), which includes objects that are not text-like. A value of True will cause Word Pro to skip over objects that are not text-like. Text-like objects include text, Click Here Blocks, and bookmarks. Use this parameter only when calling this method from a Text, TextMarker, or ClickHere object.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Even though this method has the same name on each object, it will behave differently from one object to the next. For example, when you call this method from a Layout object, it behaves differently than it does when you call it from a Text object.

- From a Layout object
When you call this method from a Layout object, it moves that Layout object forward one level in relation to the other layout objects of the same type. For example, a FrameLayout would be moved forward one step in the hierarchy of FrameLayouts.
- From a container object
When you call this method from a container object, Word Pro places the insertion point at the beginning of the next page.
- From a Text object

When you call this method from a Text, TextMarker, or ClickHere object, Word Pro moves the insertion point forward the specified number of units.

The forward method will not enter embedded streams such as frames or tables. Use the methods provided by those objects to enter their streams or remove them completely.

Word Pro: FrameRevert method

{button ,AL(^H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL(^H_FRAMEREVERT_METHOD_EXSCRIPT',1)} [See example](#)

Reverts the attributes of the currently active frame to the attributes specified in the frame style.

Syntax

[objectreference].FrameRevert()

Parameters

None.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Word Pro: FXGetNotesString method

{button ,AL(^H_DIVISION_CLASS;H_TEXTDOCUMENT_CLASS',0)} [See list of classes](#)

Returns the string expression representing a FX Notes field.

Syntax

[objectreference].FXGetNotesString(FieldName)

Parameters

FieldName

Data type is String.

Return value

Usage

Word Pro: FXGetNotesWriteHandle method

{button ,AL(^H_DIVISION_CLASS;H_TEXTDOCUMENT_CLASS',0)} [See list of classes](#)

Syntax

[objectreference].FXGetNotesWriteHandle()

Parameters

Return value

Usage

Word Pro: FXSetNotesString method

{button ,AL(^H_DIVISION_CLASS;H_TEXTDOCUMENT_CLASS',0)} [See list of classes](#)

Sets the string expression representing the name and value of a FX Notes field.

Syntax

[objectreference].FXNotesString(FieldName, Value)

Parameters

FieldName

Data type is String.

Value

Data type is String.

Return value

Usage

Word Pro: GetActiveList method

{button ,AL(^H_DIVISION_CLASS;H_TEXTDOCUMENT_CLASS',0)} [See list of classes](#)

Syntax

[objectreference].GetActiveList()

Parameters

Return value

Usage

Word Pro: GetArrayProp method

{button ,AL('H_SILVERBULLET_CLASS;H_USERINTERFACEPREFS_CLASS',0)} [See list of classes](#)

Returns the properties of an array. This method is defined in the following classes:

[SilverBullet]

[UserInterfacePrefs]

Syntax

[objectreference].SilverBullet.GetArrayProp(BulletArrayProp, Level)

[objectreference].UserInterfacePrefs.GetArrayProp(PrefPropScope, Index)

Parameters

BulletArrayProp

Data type is Variant. The value of this parameter must be one of the strings below or its numeric equivalent.

<u>Value</u>	<u>Effect</u>
\$LwpBulletArrayPropCumulative (79)	
\$LwpBulletArrayPropDivision (81)	
\$LwpBulletArrayPropLesser (78)	
\$LwpBulletArrayPropLesserspecific (77)	
\$LwpBulletArrayPropSection (80)	

Level

Data type is Integer.

Positions

Data type is Variant. The value of this parameter must be one of the strings below or its numeric equivalent.

<u>Value</u>	<u>Effect</u>
\$LwpPositionsAfter (1639)	
\$LwpPositionsBefore (1637)	
\$LwpPositionsEqual (1638)	
\$LwpPositionsUnknown (1636)	

Index

Data type is Integer. Legal values are 0, 1, 2, and 3.

Return value

String.

Usage

[SilverBullet]

[UserInterfacePrefs]

Use this method to obtain a value indicated in the Find & Replace bar.

Word Pro: GetContents method

{button ,AL(^H_MARKER_CLASS;H_POWERFIELD_CLASS;H_CLICKHERE_CLASS;H_TEXTMARKER_CLASS;H_RUBYMARKER_CLASS;H_TABLEMARKER_CLASS',0)} [See list of classes](#)

Displays the contents of a marker in a document.

Syntax

[objectreference].GetContents(IncludeMarkers)

Parameters

IncludeMarkers

Data type is Integer. The legal values for this parameter are -1 and 0 but you may use the LotusScript constants True (-1) and False (0).

Return value

Usage

Word Pro: GetCopyFormatCategories method

{button ,AL(^H_WPAPPLICATION_CLASS;H_GRAPHICOLEBJECT_CLASS;H_GRAPHIC_CLASS;H_OLEOBJECT_CLASS;H_CLICKHERE_CLASS;H_TEXTMARKER_CLASS;H_TEXT_CLASS',0)} [See list of classes](#)

{button ,AL(^H_GETCOPYFORMATCATEGORIES_METHOD_EXSCRIPT',1)} [See example](#)

Checks the current selection in a Word Pro document and determines which OLE data formats are present within that selection.

Note OLE is not supported in OS/2.

Syntax

[objectreference].GetCopyFormatCategories()

Parameters

None.

Return value

The return value for this method is always an Integer representing one or more of the values below:

- 0 - No file types
- 1 - Text in the selection
- 2 - A table or any part of a table
- 4 - A graphic
- 8 - A linked or embedded OLE object

If more than one of these format categories exists within the selection, the return value is the sum of integers for each format in the selection. For example, if both text and a graphic are in the selection, the return value is 5.

Usage

Used to determine the nature of the contents of a selection before copying that selection to the Clipboard for use in an OLE operation. To make good use of this method's return value, you should have a thorough understanding of OLE data formats, as defined in the OLE2 for Windows specifications.

Word Pro: GetCount method

{button ,AL('H_CLICKHERE_CLASS;H_TEXTMARKER_CLASS;H_TEXT_CLASS',0)} [See list of classes](#)

{button ,AL('H_GETCOUNT_METHOD_EXSCRIPT',1)} [See example](#)

Counts the words or characters in the current selection or object.

Syntax

When called from a *Text*, *TextMarker*, or *ClickHere* object:

[objectreference].GetCount(What, Which)

Parameters

What

Allows you to choose between counting within the current selection and counting within the current object. Data type is Variant, which allows the value of this parameter to be one of the constants listed below or its numeric equivalent (in parentheses). Default is \$LwpGetCountWhatText.

<u>Value</u>	<u>Effect</u>
\$LwpGetCountWhatSelection (2203)	Counts the words or characters in the current selection.
\$LwpGetCountWhatText (2204)	Counts the words or characters in the object from which you call this method.

Which

Allows you to choose between counting characters and counting words. Data type is Variant, which allows the value of this parameter to be one of the constants listed below or its numeric equivalent (in parentheses). Default is \$LwpGetCountWhichCharacter.

<u>Value</u>	<u>Effect</u>
\$LwpGetCountWhichCharacter (2205)	Counts the characters.
\$LwpGetCountWhichWord (2206)	Counts the words.

Return value

Returns the desired count.

Usage

Counts the number of words or characters within an entire object or within a selection. If nothing is selected and you specify \$LwpGetCountWhatSelection as the value of the What parameter, this method counts all the words or characters in the *Text*, *TextMarker*, or *ClickHere* object from which you call this method.

Word Pro: GetCurrentMarkerName method

```
{button ,AL('H_WPAPPLICATION_CLASS;H_CLICKHERE_CLASS;H_TEXTMARKER_CLASS;H_TEXT_CLASS',0)}
```

[See list of classes](#)

```
{button ,AL('H_GETCURRENTMARKERNAME_METHOD_EXSCRIPT',1)} See example
```

Returns the name of the specified type of marker uppermost in the current focus.

Syntax

```
[objectreference].GetCurrentMarkerName(MarkerType)
```

Parameters

MarkerType

Specifies which type of marker Word Pro should look for. If Word Pro find this type of marker in the focus, it returns the name of the marked object. Data type is Variant, which allows the value of this parameter to be one of the constants listed below or its numeric equivalent (in parentheses). There is no default value.

<u>Value</u>	<u>Effect</u>
\$LwpMarkerTypeBookmark (590)	Looks for a Bookmark marker.
\$LwpMarkerTypeClickhere (593)	Looks for a Click Here marker.
\$LwpMarkerTypeField (592)	Looks for a Power Field marker.
\$LwpMarkerTypeDde (591)	Looks for a DDE marker.

Return value

Returns the name of the marker which has the focus when the method is called. Data type is String.

- Markers for Bookmarks return the internal bookmark name in hexadecimal format. Use this internal name to get the external bookmark name as seen in the Bookmarks dialog box.
- Markers for Click Here Blocks return the name you provide in the Click Here Block Options dialog box.
- Markers for Power Fields return the internal power field name in hexadecimal format. The external names of power fields as seen in the Document Fields dialog box are not available within LotusScript.
- Markers for DDE objects return the internal name in hexadecimal format.

For more on internal and external names, see [Overview: Internal and External names in LotusScript](#)

Usage

When you call this method, Word Pro looks for the type of marker you specified in the MarkerType parameter. If no marker of that type is present within the focus, this method returns a null value.

To give a marker the focus, move the insertion point so it is within the bookmarked text, the Click Here Block, the Power Field, or the DDE call.

Word Pro: GetData method

{button ,AL('H_SCRIPTDATASET_CLASS;H_WPDATASET_CLASS',0)} [See list of classes](#)

{button ,AL('H_GETDATA_METHOD_EXSCRIPT',1)} [See example](#)

Returns the value of a variable in a data set.

Syntax

[objectreference].GetData(DataName,Default)

Parameters

DataName

A String value representing the variable name for which you want to return the value.

Default

A variant value representing the default value of the variable that you want to return.

Return value

Variant

Usage

Use this method to return the data associated with a variable. When you return the data, you must specify the variable name and default value. If the variable does not exist, it should return the default value.

Note In the 16-bit Version of Word Pro, you specify the variable name and a null string default value. If the variable does not exist, it should return the null string default value.

Word Pro: GetDisplayableFilterName method

{button ,AL(^H_FILTER_CLASS',0)} [See list of classes](#)

Syntax

[objectreference].GetDisplayableFilterName(FilterId)

Parameters

FilterId

Data type is Integer.

Return value

Usage

Word Pro: GetEnum method

{button ,AL('H_APPLICATION_CLASS;H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL('H_GETENUM_METHOD_EXSCRIPT',1)} [See example](#)

Returns the numeric equivalent of an enumerated string value.

Syntax

[objectreference].GetEnum(Name)

Parameters

Name

The String value for which you need the numeric equivalent.

Return value

Variant

Usage

In many properties and methods, Word Pro provides an enumerated list of legal values. Each value can be expressed as either a string or a number while working within LotusScript. However, if you need to use one of the values through another programming language such as Visual Basic, you must use the numeric value.

This method returns the numeric equivalent of an enumerated string value.

Word Pro: GetEnvelopeDefaults method

{button ,AL(`H_PRINTMANAGER_CLASS',0)} [See list of classes](#)

{button ,AL(`H_GETENVELOPEDEFAULTS_METHOD_EXSCRIPT',1)} [See example](#)

Retrieves the default envelope setting in the current document.

Syntax

[objectreference].GetEnvelopeDefaults()

Parameters

Return value

Returns the sheet feeder bin used as a default for envelopes, as a String.

Usage

Word Pro: GetFileDescription method

{button ,AL(^H_FILTER_CLASS',0)} [See list of classes](#)

Retrieves a description of a file.

Syntax

[objectreference].GetFileDescription(FileName)

Parameters

FileName

Data type is String.

Return value**Usage**

Word Pro: GetFormula method

{button ,AL('H_CELLENGINE_CLASS',0)} [See list of classes](#)

{button ,AL('H_GETFORMULA_METHOD_EXSCRIPT',1)} [See example](#)

Returns a string containing the formula that resides in a table cell.

Syntax

[objectreference].GetFormula(Row, Column)

Parameters

Row

This Integer parameter allows you to specify the row ID of the cell from which you want to return a formula.

Column

This Integer parameter allows you to specify the column ID of the cell from which you want to return a formula.

Return value

This method returns a String containing the formula that is stored within a specified cell.

Usage

Row and column ID values are zero based. The first row of a table has a row ID value of 0, and the first column of a table has a column ID value of 0.

Word Pro: GetInternetFile method

{button ,AL(^H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

Retrieves the text and HTML code of the specified URL and stores them in a temporary file.

Syntax

[objectreference].GetInternetFile(URL)

Parameters

URL

A String expression which specifies the URL for the Internet file you want to retrieve.

Return value

Returns the name of the temporary file in which the URL contents are stored. Data type is String.

Usage

Graphics and other external files referenced in the HTML code are not retrieved by this method.

To display the Internet file, the user can use the Web browsing features in Word Pro or the [RetrieveInternetFileAndOpen](#) method in LotusScript.

Word Pro: GetLastUsedFilter method

{button ,AL('H_FILTER_CLASS',0)} [See list of classes](#)

{button ,AL('H_GETLASTUSEDFILTER_METHOD_EXSCRIPT',1)} [See example](#)

Retrieves the last filter used to import or export the current document.

Syntax

[objectreference].GetLastUsedFilter(FilterType)

Parameters

FilterType

Data type is Variant. The value of this parameter must be one of the strings below or its code equivalent.

\$LwpFilterTypeGraphic (280)

\$LwpFilterTypeTable (281)

\$LwpFilterTypeText (279)

Return value

Usage

Word Pro: GetLineMix method

{button ,AL(^H_TABLELINE_CLASS',0)} [See list of classes](#)

Returns whether or not all sides of a table have the same type of line style.

Syntax

[objectreference].GetLineMix()

Parameters**Return value****Usage**

Word Pro: GetLineStyle method

{button ,AL('H_TABLELINE_CLASS',0)} [See list of classes](#)

{button ,AL('H_GETLINESTYLE_METHOD_EXSCRIPT',1)} [See example](#)

Returns the borderline, outline, or diagonal line style used in a table in the current document.

Syntax

[objectreference].GetLineStyle()

Parameters**Return value**

Returns the legal values found in the [ChgLineStyle method](#). For information, see the ChgLineStyle method.

Usage

Word Pro: GetLinkDisplayNameFileLength method

{button ,AL(^H_GRAPHIC_CLASS',0)} [See list of classes](#)

Syntax

[objectreference].GetLinkDisplayNameFileLength(LinkCookie)

Parameters

LinkCookie

Data type is Long.

Return value

Long.

Usage

Word Pro: GetLinkName method

{button ,AL(^H_GRAPHIC_CLASS',0)} [See list of classes](#)

Syntax

[objectreference].GetLinkName(LinkCookie)

Parameters

LinkCookie

Data type is Long.

Return value

String.

Usage

Word Pro: GetLinkSourceName method

{button ,AL(^H_GRAPHIC_CLASS',0)} [See list of classes](#)

Syntax

[objectreference].GetLinkSourceName(LinkCookie)

Parameters

LinkCookie

Data type is Long.

Return value

String.

Usage

Word Pro: GetMarkedText method

{button ,AL(^H_MARKER_CLASS;H_POWERFIELD_CLASS;H_CLICKHERE_CLASS;H_TEXTMARKER_CLASS;H_RUBYMARKER_CLASS;H_TABLEMARKER_CLASS',0)} [See list of classes](#)

{button ,AL(^H_GETMARKEDTEXT_METHOD_EXSCRIPT',1)} [See example](#)

Retrieves marked text in the current document.

Syntax

[objectreference].GetMarkedText()

Parameters

Return value

Usage

Word Pro: GetMarkerName method

{button ,AL(^H_DDELINKMANAGER_CLASS;H_LAYOUT_CLASS;H_CELLGROUPLAYOUT_CLASS;H_CELLLAYO
UT_CLASS;H_CONNECTEDLAYOUT_CLASS;H_COLUMNGROUPLAYOUT_CLASS;H_FOOTERLAYOUT_CLA
SS;H_FRAMELAYOUT_CLASS;H_FRAMEGROUPLAYOUT_CLASS;H_DROPCAPLAYO_U_T_CLASS;H_GROUPL
AYOUT_CLASS;H_HEADERLAYOUT_CLASS;H_NOTELAYOUT_CLASS;H_PAGELAYOUT_CLASS;H_ROWGR
OUPLAYOUT_CLASS;H_ROWLAYOUT_CLASS;H_RUBYLAYOUT_CLASS;H_SUPERTABLEGROUPLAYOUT_
CLASS;H_SUPERTABLELAYOUT_CLASS;H_TABLELAYOUT_CLASS;H_ENDNOTELAYOUT_CLASS;H_FOOT
NOTELAYOUT_CLASS;H_TABLEHEADINGLAYOUT_CLASS;H_TOCSUPERTABLELAYOUT_CLASS;H_BASET
ABLE_CLASS;H_FOOTNOTETABLE_CLASS;H_PARALLELOLUMNS_CLASS;H_GLOSSARY_CLASS;H_TABL
EHEADING_CLASS;H_TABLE_CLASS',0)} [See list of classes](#)

{button ,AL(^H_GETMARKERNAME_METHOD_EXSCRIPT',1)} [See example](#)

This method is defined in the following classes:

[BaseTable]

Retrieves a list of all the marker names of the specified type associated with a table object layout.

[DdeLinkManager]

Retrieves the name of the Dde marker which you requested.

[Layout]

Retrieves a list of all the marker names of the specified type associated with a layout.

Syntax

[objectreference].DdeLinkManager.GetMarkerName(DdeFind,p2)

[objectreference].Layout.GetMarkerName(MarkerType)

[objectreference].BaseTable.GetMarkerName(MarkerType)

Parameters

DdeFind

Data type is Variant. The value of this parameter must be one of the strings below or its numeric equivalent.

<u>Value</u>	<u>Effect</u>
\$LwpDdeFindLinknameUsingMarker (173)	Returns the user name (you have the internal [hexidecimal] name).
\$LwpDdeFindMarkerNameUsingHandle (172)	Returns the internal name (you have the conversation handle).
\$LwpDdeFindMarkerUsingLinkInfo (174)	Returns the internal name (you have the user name).

p2

Data type is Variant.

MarkerType

Data type is Variant, which allows it to be one of the string constants below or its numeric equivalent.

<u>Value</u>	<u>Effect</u>
\$LwpMarkerTypeDefault (589)	Returns a positional marker; it is not of any other type and is only used to mark a spot or position.
\$LwpMarkerTypeBookmark (590)	Returns all the specified bookmark markers.
\$LwpMarkerTypeDde (591)	Returns all the specified Dde markers.
\$LwpMarkerTypeField (592)	Returns all the specified field markers.
\$LwpMarkerTypeClickhere (593)	Returns all the specified Click Here markers.
\$LwpMarkerTypeRuby (594)	Returns all the specified Ruby markers.

Return value

[DdeLinkManager]

String.

[Layout]

Returns a list of names of all marker objects that are associated with the layout object.

Usage

Word Pro: GetMisspelledWord method

{button ,AL('H_CLICKHERE_CLASS;H_TEXTMARKER_CLASS;H_TEXT_CLASS',0)} [See list of classes](#)

{button ,AL('H_GETMISPelledWORD_METHOD_EXSCRIPT',1)} [See example](#)

Finds the next misspelled word in the current text stream.

Syntax

[objectreference].GetMisspelledWord([EndMarkerName])

Parameters

EndMarkerName

A String expression which specifies the name of the marked range within which you want to find a misspelled word.

Optional parameter which should only be used if the following statements are true:

- You have a marked range within your document.
- You want Word Pro to restrict its search to that marked range.

If you provide a range name and the insertion point is located within that named range, Word Pro searches the range for the next misspelled word. If the insertion point is not within the named range, Word Pro does nothing.

Return value

Returns a String representing the next misspelled word in the text stream.

Usage

Equivalent to choosing Edit - Check Spelling, with the exception that this method does not highlight all misspelled words and it does not open the Spell Check bar.

If the insertion point is within or at the end of a misspelled word when you call this method, Word Pro returns that misspelled word. Otherwise, Word Pro moves to, selects, and returns the next misspelled word in the current text stream.

For more on the nature of text streams, see [Text Class](#).

Word Pro: GetNamedProperty method

```
{button ,AL('H_CHARACTERSTYLE_CLASS;H_DIVISION_CLASS;H_LAYOUT_CLASS;H_CELLGROUPLAYOUT_CLASS;H_CELLLAYOUT_CLASS;H_CONNECTEDLAYOUT_CLASS;H_COLUMNGROUPLAYOUT_CLASS;H_FOOTERLAYOUT_CLASS;H_FRAMELAYOUT_CLASS;H_FRAMEGROUPLAYOUT_CLASS;H_DROPCAPLAYO  
UT_CLASS;H_GROUPLAYOUT_CLASS;H_HEADERLAYOUT_CLASS;H_NOTELAYOUT_CLASS;H_PAGELAYO  
UT_CLASS;H_ROWGROUPLAYOUT_CLASS;H_ROWLAYOUT_CLASS;H_RUBYLAYOUT_CLASS;H_SUPERTAB  
LEGROUPLAYOUT_CLASS;H_SUPERTABLELAYOUT_CLASS;H_TABLELAYOUT_CLASS;H_ENDNOTELAYO  
UT_CLASS;H_FOOTNOTELAYOUT_CLASS;H_TABLEHEADINGLAYOUT_CLASS;H_TOCSUPERTABLELAYOUT  
_CLASS;H_MARKER_CLASS;H_POWERFIELD_CLASS;H_CLICKHERE_CLASS;H_TEXTMARKER_CLASS;H_  
RUBYMARKER_CLASS;H_TABLEMARKER_CLASS;H_TEXTDOCUMENT_CLASS',0)} See list of classes
```

```
{button ,AL('H_GETNAMEDPROPERTY_METHOD_EXSCRIPT',1)} See example
```

Retrieves the value of the named property in the current object.

Syntax

```
[objectreference].GetNamedProperty(PropertyName)
```

Parameters

PropertyName

A String expression representing the name of the property you want to retrieve.

Return value

String.

Usage

A named property is a user-defined property assigned to an object. Unlike variables, named properties are persistent. They continue to exist when a script stops executing, and when a document is closed and reopened.

This method returns the String value stored in a specific named property. If you refer to a named property which doesn't exist on an object, this method returns an empty string. A run-time error does not occur. Use the HasNamedProperty method to determine whether or not a particular named property exists for an object.

Word Pro: GetNameFromPage method

{button ,AL('H_DIVISION_CLASS;H_TEXTDOCUMENT_CLASS',0)} [See list of classes](#)

{button ,AL('H_GETNAMEFROMPAGE_METHOD_EXSCRIPT',1)} [See example](#)

Returns the name of the page of the current division.

Syntax

[objectreference].GetNameFromPage(PageNumber,[IsReturnInternalName])

Parameters

PageNumber

Data type is Integer.

IsReturnInternalName

Data type is Boolean. Optional parameter. Default is False.

Return value

Usage

Word Pro: GetObjectList method

```
{button ,AL(^H_BASECONTAINER_CLASS;H_CELLCONTAINER_CLASS;H_FRAMECONTAINER_CLASS;H_NOTECONTAINER_CLASS;H_RUBYCONTAINER_CLASS;H_DROPDOWNCONTAINER_CLASS;H_PAGECONTAINER_CLASS;H_SUBPAGECONTAINER_CLASS;H_SUPERPAGECONTAINER_CLASS;H_ROWCONTAINER_CLASS;H_SUPERTABLECONTAINER_CLASS;H_TABLECONTAINER_CLASS;H_PARALLELCOLSCONTAINER_CLASS;H_TABLEONLYCONT_CLASS',0)} See list of classes
```

```
{button ,AL(^H_GETOBJECTLIST_METHOD_EXSCRIPT',1)} See example
```

Returns a value indicating the type of a specific container.

Syntax

[objectreference].GetObjectList()

Parameters

Return value

<u>Value</u>	<u>Effect</u>
7	The container is a PageContainer object.
8	The container is a FrameContainer object.
23	The container is a TableContainer object.
24	The container is a RowContainer object.
25	The container is a CellContainer object.
550	The container is a SuperTableContainer object.
107	The container is a HeaderContainer object.
108	The container is a FooterContainer object.

Usage

Word Pro: GetParaNumber method

{button ,AL('H_CLICKHERE_CLASS;H_TEXTMARKER_CLASS;H_TEXT_CLASS',0)} [See list of classes](#)

{button ,AL('H_GETPARAMNUMBER_METHOD_EXSCRIPT',1)} [See example](#)

Returns an integer which indicates the paragraph number of the current paragraph.

See Usage for details.

Syntax

[objectreference].GetParaNumber(Position)

Parameters

Position

Data type is Integer.

Return value

The number of the paragraph.

Usage

To understand the meaning of the paragraph number, you must first understand outline sequence positions.

Outline sequence positions

There are nine positions available for an outline sequence. These positions tell Word Pro how to organize and indent paragraphs within an outline. For example, if you assign the HEAD and the SUBHEAD styles to Position 1, and the BODY and INDENT styles to Position 2, all paragraphs using the HEAD or SUBHEAD style are assigned to Position 1 in an outline, and all paragraphs using the BODY or INDENT styles are assigned to Position 2. In this way, paragraphs using different styles can be grouped by their common outline sequence position.

You can see these positions displayed in the Set Outline Style Sequence dialog box. Choose Text - Outline - Outline Styles.

Note You can assign one or more paragraph styles to each of these positions.

The paragraph number

This method returns a number that indicates the number of the current paragraph among all other paragraphs that share the same outline sequence position.

For example, if you have 30 paragraphs in your document and three of those paragraphs are at Position 4 in your outline sequence, when you call this method from one of the Position 4 paragraphs, it tells you whether the paragraph is the first, second, or third paragraph at Position 4. The paragraph number is not affected by paragraphs that are not at Position 4.

In short, the paragraph number tells you the number of the current paragraph in relation to other paragraphs at the same outline sequence position.

Word Pro: GetPasteFormatCategories method

```
{button ,AL('H_WPAPPLICATION_CLASS;H_BASECONTAINER_CLASS;H_CELLCONTAINER_CLASS;H_FRAMECONTAINER_CLASS;H_NOTECONTAINER_CLASS;H_RUBYCONTAINER_CLASS;H_DROPDOWNCONTAINER_CLASS;H_PAGECONTAINER_CLASS;H_SUBPAGECONTAINER_CLASS;H_SUPERPAGECONTAINER_CLASS;H_ROWCONTAINER_CLASS;H_SUPERTABLECONTAINER_CLASS;H_TABLECONTAINER_CLASS;H_PARALLELCOLSCONTAINER_CLASS;H_TABLEONLYCONT_CLASS;H_GRAPHIC_CLASS;H_CLICKHERE_CLASS;H_TEXTMARKER_CLASS;H_TEXT_CLASS',0)} See list of classes
```

```
{button ,AL('H_GETPASTEFORMATCATEGORIES_METHOD_EXSCRIPT',1)} See example
```

Looks at the focus in a Word Pro document and determines which OLE data formats can be pasted at the insertion point.

Note OLE is not supported under OS/2.

Syntax

```
[objectreference].GetPasteFormatCategories()
```

Parameters

None.

Return value

The return value for this method is always an Integer representing one or more of the values below:

<u>Value</u>	<u>Effect</u>
0	No data types can be pasted at the insertion point.
1	Text can be pasted at the insertion point.
2	A table or any part of a table can be pasted at the insertion point.
4	A graphic can be pasted at the insertion point.
8	A linked or embedded OLE object can be pasted at the insertion point.
15	All formats listed above can be pasted at the insertion point.

If more than one of these format categories can be pasted in the current context, the return value is the sum of integers for each format that can be pasted. For example, if it is possible to paste all the formats, the return value is 15 (the sum of 1, 2, 4 and 8).

Usage

To make good use of this method's return value, you should have a thorough understanding of OLE data formats, as defined in the OLE2 for Windows specifications.

Word Pro: GetPosition method

{button ,AL(^H_CLICKHERE_CLASS;H_TEXTMARKER_CLASS;H_TEXT_CLASS',0)} [See list of classes](#)

{button ,AL(^H_GETPOSITION_METHOD_EXSCRIPT',1)} [See example](#)

Retrieves the position of the insertion point relative to the specified Marker object.

Syntax

[objectreference].GetPosition(MarkerName)

Parameters

MarkerName

A String expression which specifies the name of the Marker object for which you want the position.

Return value

Returns one of the four integers listed below:

<u>Return Value</u>	<u>Definition</u>
-2	The insertion point is not located in the same text stream as the specified Marker object.
-1	The insertion point is located before the specified Marker object.
0	The insertion point is located within the specified Marker object.
1	The insertion point is located after the specified Marker object.

Usage

Word Pro: GetPrinterInfo method

{button ,AL('H_PRINTMANAGER_CLASS',0)} [See list of classes](#)

{button ,AL('H_GETPRINTERINFO_METHOD_EXSCRIPT',1)} [See example](#)

Displays information about the selected printer for the current document.

Syntax

[objectreference].GetPrinterInfo()

Parameters

Return value

Usage

Word Pro: GetProfileString method

{button ,AL('H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL('H_GETPROFILESTRING_METHOD_EXSCRIPT',1)} [See example](#)

Returns the specified profile string from the specified .INI file. You can get a profile string from any of the standard Word Pro .INI files or you can specify another .INI file.

Syntax

[objectreference].GetProfileString(Section, Key[, DefString][, IniFileType][, WhichIniLocation][, IniName])

Parameters

Section

A String expression which specifies a name of a section in the .INI. Word Pro searches only the section you name in this parameter. If the named section does not match a section in the specified INI, this method fails. If you use an empty string ("") Word Pro assumes you are searching the LWPUSER.INI file (IniFileType parameter = "\$LwpIniUserPrefs") and looks for the "WordProUser" section. Most .INIs have more than one section. The section name you provide in this parameter must match the section name in the INI exactly.

Key

A String expression which specifies the key name in the section you are searching.

DefString

An optional String expression which allows you to return a default string, if Word Pro fails to find the specified key.

IniFileType

Specifies the .INI from which you want to get the profile string. You can choose one of the standard Word Pro .INI files or choose \$LwpIniCustomFile to search another .INI file. Data type is Variant which allows the value of this parameter to be one of the constants listed below or its numeric equivalent (in parentheses). You do not have to provide a value for this parameter. Default is \$LwpIniUserPrefs.

<u>Value</u>	<u>Effect</u>
\$LwpIniUserPrefs (2101)	Default; the .INI file used to store Word Pro's user preference information (lwpuser.ini).
\$LwpIniConfigPrefs (2102)	The .INI file used to store Word Pro's configuration preference information.
\$LwpIniEnvelopeAndMerge (2105)	The .INI file used to store Word Pro's envelope and merge information.
\$LwpIniLanguages (2107)	The .INI file used to store some of Word Pro's language information.
\$LwpIniSharedLotusInfo (2103)	The .INI file used to store shared information between Word Pro and other Lotus products.
\$LwpIniSmartcorrect (2106)	The .INI file used to store Word Pro's SmartCorrect information.
\$LwpIniSmartfill (2104)	The .INI file used to store Word Pro's SmartFill lists.
\$LwpIniCustomfile (2100)	Allows you to get a profile string from an .INI file that is not one of the standard Word Pro .INI files. If you use this value, you must use the IniName parameter to specify the name of the .INI file

(Windows 3.1 or OS/2) or .INI entry (Windows 95)
in which the profile string is located.

WhichIniLocation

Tells Word Pro whether to look on the network or the local machine for the specified .INI file. Data type is Variant which allows the value of this parameter to be one of the constants listed below or its numeric equivalent (in parentheses). You do not have to provide a value for this parameter. Default is \$LwpUserIniLocation.

<u>Value</u>	<u>Effect</u>
\$LwpNetworkIniLocation (2171)	Searches directory for network .INI files.
\$LwpUserIniLocation (2172)	Searches directory for user .INI files.

Note For Windows 95, in the registry, the user location is HKEY_CURRENT_USER or HKEY_USERS. The network location is HKEY_LOCAL_MACHINE. Within either of these locations, the path below this would be: Software\Lotus\WordPro\96.0.

IniName

An optional String expression that identifies which .INI you want to search. Use this parameter only if you used \$LwpIniCustomFile as the value of the IniFileType parameter. This .INI must be stored in the same directory as the standard Word Pro .INIs.

Note If you are using Windows 3.1 or OS/2, this value is an .INI file name. If you are using Windows 95, this value is an .INI entry as seen in the Windows Registry application (REGEDIT.EXE).

Return value

Returns the profile string. Data type is String.

Usage

Word Pro: GetRGB method

{button ,AL('H_COLOR_CLASS',0)} [See list of classes](#)

{button ,AL('H_GETRGB_METHOD_EXSCRIPT',1)} [See example](#)

Returns a hexadecimal value representing the red, green, and blue components of an object's color.

Syntax

[objectreference].GetRGB()

Parameters

Return value

Returns the RGB values for a specific object. If the color of a specific object is a predefined Word Pro color, then Word Pro returns the RGB value of that predefined color.

Usage

Colors are usually represented by a combination of three separate components. The three components include a red value which can range from 0-255, a green value which can range from 0-255, and a blue value which can range from 0-255. You can combine different amounts of these three component colors to produce any other color. For example, in order to produce yellow, you can set a color object's red value to 255, green value to 255, and blue value to 0. The combination of all three of the component colors appears yellow.

Colors can also be represented by a single numeric value. The GetRGB method returns a single hexadecimal value which represents the values for red, green, and blue components of a color. The table below lists the bit-field values that make up this hexadecimal value.

Value	Effect
Bits 0 - 7	Blue
Bits 8 - 15	Green
Bits 16 - 23	Red

If you use this method on an object that is red, it returns a hexadecimal value of FF0000. The first two digits represent the value of the red component. The hexadecimal value FF equates to a decimal value of 255. The second two digits represent the green component value. The last two digits represent the red component value. Since the object is red, the green and blue component values are 0.

The following table shows some common colors, their hexadecimal values, and their RGB values.

Color	Hexadecimal Value	Red	Green	Blue
White	FFFFFF	255	255	255
25% Gray	C0C0C0	192	192	192
Red	FF0000	255	0	0
Yellow	FFFF00	255	255	0
Neon Green	FF00	0	255	0
Turquoise	FFFF	0	255	255
Blue	FF	0	0	255
Hot Pink	FF00FF	255	0	255
Black	0	0	0	0
50% Gray	808080	128	128	128
Scarlet	800000	128	0	0
Olive	808000	128	128	0
Dark Green	8000	0	128	0
Aztec Blue	8080	0	128	128
Dark Grape	61	0	0	97
Plum Red	800080	128	0	128

Word Pro: GetRightContextMenu method

{button ,AL(^H_GRAPHICOLEBJECT_CLASS;H_GRAPHIC_CLASS;H_OLEOBJECT_CLASS',0)} [See list of classes](#)

Syntax

[objectreference].GetRightContextMenu()

Parameters

None.

Return value

Long.

Usage

Word Pro: GetSource method

{button ,AL(^H_GRAPHIC_CLASS',0)} [See list of classes](#)

Syntax

[objectreference].GetSource(LinkCookie)

Parameters

LinkCookie

Data type is Long.

Return value

Integer.

Usage

Word Pro: GetSpellStatus method

{button ,AL(^H_CLICKHERE_CLASS;H_TEXTMARKER_CLASS;H_TEXT_CLASS',0)} [See list of classes](#)

Checks the status of the Spell Check utility and forces it to load, if not already loaded.

Syntax

[objectreference].GetSpellStatus([Language])

Parameters

Language

Do not use this parameter. Word Pro sets the value automatically.

Return value

Should always return a value of -1 (True). If the return value is 0, there is a problem loading the Spell Check utility.

Usage

Word Pro: GetSpellUserDictStatus method

{button ,AL(^H_CLICKHERE_CLASS;H_TEXTMARKER_CLASS;H_TEXT_CLASS',0)} [See list of classes](#)

Checks to see whether or not there is a user dictionary available for the specified language.

Syntax

[objectreference].GetSpellUserDictStatus([Language])

Parameters

Language

Specifies which dictionary you want to check. Data type is Variant, which allows the value of this parameter to be one of the constants listed below or its numeric equivalent (in parentheses). Default is \$LwpLanguagesSystem (434).

\$LwpLanguagesAfrikaans (474)	\$LwpLanguagesGerman (457)
\$LwpLanguagesAlbanian (475)	\$LwpLanguagesGermanAustrian (459)
\$LwpLanguagesAmerican (442)	\$LwpLanguagesGermanLiechtenstein (2140)
\$LwpLanguagesArabic (476)	\$LwpLanguagesGermanLuxembourg (2139)
\$LwpLanguagesArabicAlgeria (2111)	\$LwpLanguagesGermanSwiss (458)
\$LwpLanguagesArabicBahrain (2121)	\$LwpLanguagesGreek (460)
\$LwpLanguagesArabicEgypt (2109)	\$LwpLanguagesHebrew (483)
\$LwpLanguagesArabicIraq (2108)	\$LwpLanguagesHungarian (461)
\$LwpLanguagesArabicJordan (2117)	\$LwpLanguagesIcelandic (484)
\$LwpLanguagesArabicKuwait (2119)	\$LwpLanguagesIndonesian (2141)
\$LwpLanguagesArabicLebanon (2118)	\$LwpLanguagesItalian (462)
\$LwpLanguagesArabicLibya (2110)	\$LwpLanguagesItalianSwiss (463)
\$LwpLanguagesArabicMorocco (2112)	\$LwpLanguagesJapanese (485)
\$LwpLanguagesArabicOman (2114)	\$LwpLanguagesKorean (486)
\$LwpLanguagesArabicQatar (2122)	\$LwpLanguagesKoreanJohab (2142)
\$LwpLanguagesArabicSyria (2116)	\$LwpLanguagesLatvian (2143)
\$LwpLanguagesArabicTunisia (2113)	\$LwpLanguagesLithuanian (2144)
\$LwpLanguagesArabicUae (2120)	\$LwpLanguagesMedical (448)
\$LwpLanguagesArabicYemen (2115)	\$LwpLanguagesNorwegian (464)
\$LwpLanguagesAustralian (444)	\$LwpLanguagesNynorsk (465)
\$LwpLanguagesBasque (2123)	\$LwpLanguagesPolish (466)
\$LwpLanguagesBrazilian (468)	\$LwpLanguagesPortuguese (467)
\$LwpLanguagesBritish (443)	\$LwpLanguagesRhaetoRoman (2145)
\$LwpLanguagesBritishmedize (451)	\$LwpLanguagesRomanian (488)
\$LwpLanguagesBrmedical (449)	\$LwpLanguagesRussian (469)
\$LwpLanguagesBulgarian (478)	\$LwpLanguagesRussianio (470)
\$LwpLanguagesByelorussian (2124)	\$LwpLanguagesSlovak (492)
\$LwpLanguagesCatalan (436)	\$LwpLanguagesSlovene (493)
\$LwpLanguagesChineseHongkong (2126)	\$LwpLanguagesSorbian (2146)
\$LwpLanguagesChinesePrchina (2125)	\$LwpLanguagesSpanish (471)
\$LwpLanguagesChineseSingapore (2127)	\$LwpLanguagesSpanishArgentina (2155)
\$LwpLanguagesChineseTraditional (479)	\$LwpLanguagesSpanishBolivia (2160)
\$LwpLanguagesCroatian (2128)	\$LwpLanguagesSpanishChile (2157)
\$LwpLanguagesCroatianCyrillic (2130)	\$LwpLanguagesSpanishColombia (2153)
\$LwpLanguagesCroatianSerbian (2131)	\$LwpLanguagesSpanishCostarica (2149)
\$LwpLanguagesCzech (437)	\$LwpLanguagesSpanishDominican (2151)

\$LwpLanguagesDanish (438)	\$LwpLanguagesSpanishEcuador (2156)
\$LwpLanguagesDutch (439)	\$LwpLanguagesSpanishGuatemala (2148)
\$LwpLanguagesDutchBelgian (440)	\$LwpLanguagesSpanishMexican (472)
\$LwpLanguagesEnglishCanadian (445)	\$LwpLanguagesSpanishModern (2147)
\$LwpLanguagesEnglishCaribbean (2134)	\$LwpLanguagesSpanishPanama (2150)
\$LwpLanguagesEnglishIreland (447)	\$LwpLanguagesSpanishParaguay (2159)
\$LwpLanguagesEnglishJamaica (2133)	\$LwpLanguagesSpanishPeru (2154)
\$LwpLanguagesEnglishNewzealand (446)	\$LwpLanguagesSpanishUruguay (2158)
\$LwpLanguagesEnglishSafrica (2132)	\$LwpLanguagesSpanishVenezuela (2152)
\$LwpLanguagesEstonian (2135)	\$LwpLanguagesSwedish (473)
\$LwpLanguagesFaeroese (2136)	\$LwpLanguagesSystem (434)
\$LwpLanguagesFarsi (2137)	\$LwpLanguagesThai (494)
\$LwpLanguagesFinnish (452)	\$LwpLanguagesTurkish (495)
\$LwpLanguagesFrench (453)	\$LwpLanguagesUkrainian (496)
\$LwpLanguagesFrenchBelgian (454)	\$LwpLanguagesUniversal (435)
\$LwpLanguagesFrenchCanadian (455)	\$LwpLanguagesUrdu (497)
\$LwpLanguagesFrenchLuxembourg (2138)	\$LwpLanguagesVoorkeur (441)
\$LwpLanguagesFrenchSwiss (456)	

Return value

If there is a user dictionary assigned to the language, this method returns a value of -1 (True); if not, returns 0 (False), indicating that the method succeeded or failed respectively.

Usage

Word Pro: GetStandardButtonId method

{button ,AL('H_STATUSBAR_CLASS',0)} [See list of classes](#)

{button ,AL('H_GETSTANDARDBUTTONID_METHOD_EXSCRIPT',1)} [See example](#)

Obtains the ID of a standard Word Pro status bar button.

Syntax

[objectreference].GetStandardButtonId(ButtonType)

Parameters

Button Type

Data type is Variant which allows the value of this parameter to be one of the constants listed below or its numeric equivalent (in parentheses). There is no default.

<u>Value</u>	<u>Effect</u>
\$LwpStandButtBoldButton (1812)	Specifies the ID of a bold button.
\$LwpStandButtCollapsibleButton (1816)	Specifies the ID of a collapsible button.
\$LwpStandButtColorButton (1811)	Specifies the ID of a color button.
\$LwpStandButtCustomButton (1823)	Specifies the ID of a custom button.
\$LwpStandButtFontButton (1809)	Specifies the ID of a font button.
\$LwpStandButtItalicButton (1813)	Specifies the ID of an italic button.
\$LwpStandButtPagenumberButton (1819)	Specifies the ID of a page down button.
\$LwpStandButtPagedownButton (1818)	Specifies the ID of a page number button.
\$LwpStandButtPageupButton (1817)	Specifies the ID of a page up button.
\$LwpStandButtPointSizeButton (1810)	Specifies the ID of a point size button.
\$LwpStandButtSpacer1Button (1820)	Specifies the ID of the first spacer button.
\$LwpStandButtSpacer2Button (1821)	Specifies the ID of the second spacer button.
\$LwpStandButtSpacer3Button (1822)	Specifies the ID of the third spacer button.
\$LwpStandButtStyleButton (1815)	Specifies the ID of a style button.
\$LwpStandButtUnderlineButton (1814)	Specifies the ID of an underline button.

Return value

Long.

Usage

Use this method to obtain the ID of a standard Word Pro status bar button. For example, when you are creating a new button, you can use this method to identify the standard button, after which the new button should be inserted.

Word Pro: GetText method

{button ,AL('H_CLICKHERE_CLASS;H_TEXTMARKER_CLASS;H_TEXT_CLASS',0)} [See list of classes](#)

{button ,AL('H_GETTEXT_METHOD_EXSCRIPT',1)} [See example](#)

Returns text or the name of the type of text subobject from the specified part of a Text, ClickHere, or TextMarker object.

Syntax

[objectreference].GetText(GetObjectType, Advance, [R1], [R2], [EndMarkerName], [AcrossParagraphs])

Parameters

GetObjectType

Specifies the type of object you want Word Pro to find. You can choose between a chunk, a subobject, a paragraph, a selection, a review marker, a sentence, or a word. Each is described below. Use one of the constants described below to specify the value of this parameter. Data type is Variant which means you can use a constant or its numeric equivalent (in parentheses).

\$LwpGetObjectTypeChunk (367)

Returns the text chunk at the insertion point. A chunk is comprised of a single word (a group of characters with no spaces) and all the contiguous spaces following that word. If the insertion point is at the beginning, the end, or anywhere within a word, the chunk is comprised of that word and the spaces which follow it. If the insertion point is between two spaces, the chunk is seen as all the spaces following the insertion point to the beginning of the next word. If there is no word between the spaces and the end of the paragraph, the chunk is comprised of all the spaces up to the end of the paragraph.

\$LwpGetObjectTypeObject (365)

Returns the name of the type of text subobject found at the insertion point. Note that this option does not return the actual text. It only returns the name of the type of subobject. For more on subobject types, see [Overview: Word Pro Text Subobjects](#)

\$LwpGetObjectTypeParagraph (369)

Returns all the text in the current paragraph.

\$LwpGetObjectTypeParatag (371)

Returns the text used for the TeamConsolidate/TeamReview marker. This marker only appears on paragraphs that are marked as revisions. This option returns the marker text for the paragraph at the insertion point.

\$LwpGetObjectTypeSelection (370)

Returns the selected text. Use this option in conjunction with the [Select](#) method when you want to get the text used as a bullet on a bulleted paragraph.

\$LwpGetObjectTypeSentence (368)

Returns the text of the sentence at the insertion point.

\$LwpGetObjectTypeWord (366)

Returns the text of the word at the insertion point.

Advance

Specifies whether or not Word Pro should move the cursor to the next instance of the type of object specified in the GetObjectType parameter. For example, if you set the value of this parameter to True, Word Pro finds the first instance of the object type you specify, returns the appropriate value, and then moves the cursor to the next instance of that object type. This parameter is ignored if you use \$LwpGetObjectTypeSelection (370) or \$LwpGetObjectTypeParatag (371) as the value for the GetObjectType parameter. Data type is Integer. The legal values for this parameter are -1 and 0 but you may use the LotusScript constants True (-1) and False (0). This parameter must have a value. There is no default value.

R1

Not used. Default value is 0.

R2

Not used. Default value is 0.

EndMarkerName

A String expression which specifies the name of the marker at which you want to end the GetText operation. For

example, if you have a marked range of text and the marker name is "MyRange", you can use "MyRange" as the value for this parameter and Word Pro stops the execution of this method if it encountered the end of the marked range. Optional parameter.

AcrossParagraphs

Allows you to get all the text in a selection instead of stopping at the first paragraph. Data type is Integer. The legal values for this parameter are -1 and 0 but you may use the LotusScript constants True (-1) and False (0). Optional parameter. Default is False (0).

Return value

String.

Usage

On bulleted text, you can use this method in conjunction with the Select method to retrieve the text used as a bullet.

Word Pro: GetUniqueName method

{button ,AL('H_BOOKMARKMANAGER_CLASS',0)} [See list of classes](#)

{button ,AL('H_GETUNIQUENAME_METHOD_EXSCRIPT',1)} [See example](#)

Returns a unique name for a bookmark.

Syntax

[objectreference].GetUniqueName()

Parameters

None.

Return value

A String value representing a bookmark name that is not being used in the BookmarkManager object.

Usage

Returns a name which is not currently in use for a bookmark in the specified division. This method returns a case sensitive bookmark name, so "Bookmark1" is not the same as "BookMark1."

Word Pro: GetUserClassNameFull method

{button ,AL(^H_GRAPHIC_CLASS',0)} [See list of classes](#)

Syntax

[objectreference].GetUserClassNameFull(LinkCookie)

Parameters

LinkCookie

Data type is Long.

Return value

String.

Usage

Word Pro: GetUserClassNameShort method

{button ,AL(^H_GRAPHIC_CLASS',0)} [See list of classes](#)

Syntax

[objectreference].GetUserClassNameShort(LinkCookie)

Parameters

LinkCookie

Data type is Long.

Return value

String.

Usage

Word Pro: GetWordMisspelled method

{button ,AL('H_CLICKHERE_CLASS;H_TEXTMARKER_CLASS;H_TEXT_CLASS',0)} [See list of classes](#)

{button ,AL('H_GETWORDMISPELLED_METHOD_EXSCRIPT',1)} [See example](#)

Checks the spelling on the current word.

Syntax

[objectreference].GetWordMisspelled()

Parameters

None.

Return value

This method returns a value of -1 (True) if the current word is misspelled or 0 (False) indicating that the current word is spelled correctly.

Usage

Word Pro: Get method

{button ,AL(^H_SMARTCORRECT_CLASS',0)} [See list of classes](#)

Returns a string expression representing the replacement text for the specified SmartCorrect entry.

Syntax

[objectreference].Get(Entry)

Parameters

Entry

This String parameter represents the SmartCorrect entry for which you want to retrieve replacement text.

Return value

String.

Usage

If the entry provided does not appear in the specified SmartCorrect object, an empty string is returned.

Word Pro: Glossarize method

{button ,AL('H_GLOSSARY_CLASS',0)} [See list of classes](#)

{button ,AL('H_GLOSSARIZE_METHOD_EXSCRIPT',1)} [See example](#)

Adds a term to a Glossary object.

Syntax

[objectreference].Glossarize(Term)

Parameters

Term

Specifies the String term you want to add to the glossary.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Creates a glossary entry for the term specified by the Term parameter.

Before an entry can be added to the glossary, it must be copied to the Temporary Foundry using the InternalCopy method.

Word Pro: GlossaryInsert method

{button ,AL('H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL('H_GLOSSARYINSERT_METHOD_EXSCRIPT',1)} [See example](#)

Inserts a glossary entry at the insertion point in the currently active document. Equivalent to choosing Edit - Glossary and clicking Insert in the Glossary dialog box.

Syntax

[objectreference].GlossaryInsert(GlossFilePath, Key)

Parameters

GlossFilePath

A String expression specifying the path and name of the Glossary file in which the glossary entry is located.

Key

A String expression specifying the abbreviation for the glossary entry you want to insert into the document.

Return value

None.

Usage

Word Pro: GlossaryOpen method

{button ,AL('H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL('H_GLOSSARYOPEN_METHOD_EXSCRIPT',1)} [See example](#)

Opens a Glossary file for use with the currently active Word Pro document. You may use only one Glossary file for a document at a time.

Syntax

[objectreference].GlossaryOpen([FilePath,] [FileType,] [Password,] [AddToLastFileOpenList,] [Restore])

Parameters

FilePath

A String expression specifying the path and name of the Glossary file you want to open. Optional parameter.

FileType

A String expression indicating the file type of the file you want to use as the glossary. Word Pro automatically recognizes and imports many file types. Optional parameter. Use this parameter only if the file specified in the Path parameter is not one of these file types:

DCA/RFT	Lotus Manuscript 2.x	MS Word for Windows 1.0
DIF	Lotus Organizer 1.x	MS Word for Windows 2.0
DisplayWrite	Lotus Word Pro	MS Word for Windows 6.0
HTML	Lotus Word Pro SmartMaster	MS Word for Windows95 7.0
Lotus 1-2-3	MS Excel	MS WordPad 1.0
Lotus 1-2-3 for OS/2	MS Excel 3.0	OfficeWriter 4,5,6
Lotus 1-2-3 R3	MS Excel 4.0	Rich Text Format(RTF)
Lotus 1-2-3 R4,5	MS Excel 5.0	SAMNA Word
Lotus 1-2-3 R6	MS Excel 7.0	WordPerfect 5.0
Lotus Ami Pro	MS Windows Write 3.x	WordPerfect 5.1
Lotus Ami Pro 3.x Macro	MS Word for DOS 3,4,5,6	WordPerfect 6.x
Lotus Ami Pro 3.x Styles	MS Word for OS/2	WordStar 2000 R3

Password

Use this parameter to provide a password if the Glossary file is password protected. Data type is String. Optional parameter.

AddToLastFileOpenList

This parameter allows you to show or hide the Glossary file from the last file opened list. Data type is Integer. The legal values for this parameter are -1 and 0 but you may use the LotusScript constants True (-1) and False (0). Optional parameter. Default is False which hides the Glossary file from the list.

Restore

This parameter is not implemented.

Return value

None.

Usage

Word Pro: GoToBookmark method

{button ,AL('H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL('H_GOTOBOOKMARK_METHOD_EXSCRIPT',1)} [See example](#)

Moves the insertion point to a specified bookmark. If more than one bookmark exists with the name you provide, Word Pro moves the insertion point to the first bookmark it encounters with that name.

Syntax

[objectreference].GoToBookmark(Name)

Parameters

Name

A String expression specifying the name of the bookmark. You can indicate the name of the division that contains the bookmark by using the following statement: *divisionname!bookmarkname*.

Return value

A Boolean value of type Integer which indicates success (True) or failure (False). The constants True and False are returned as -1 and 0 respectively.

Usage

Word Pro: GoToContainer method

```
{button ,AL('H_BASECONTAINER_CLASS;H_CELLCONTAINER_CLASS;H_FRAMECONTAINER_CLASS;H_NOTECONTAINER_CLASS;H_RUBYCONTAINER_CLASS;H_DROPCAPCONTAINER_CLASS;H_PAGECONTAINER_CLASS;H_SUBPAGECONTAINER_CLASS;H_SUPERPAGECONTAINER_CLASS;H_ROWCONTAINER_CLASS;H_SUPERTABLECONTAINER_CLASS;H_TABLECONTAINER_CLASS;H_PARALLELCOLSCONTAINER_CLASS;H_TABLEONLYCONT_CLASS',0)} See list of classes
```

```
{button ,AL('H_GOTOCONTAINER_METHOD_EXSCRIPT',1)} See example
```

Moves the insertion point to the container within the current context of the document.

Syntax

[objectreference].GoToContainer(GoToLocation, [p2])

Parameters

GoToLocation

The value of this Variant parameter moves the insertion point to any one of the values below and must be one of the strings below or its code equivalent.

<u>Value</u>	<u>Effect</u>
\$LwpGoToLocationPage (372)	Moves the insertion point to the next page container within the current context of the document.
\$LwpGoToLocationHeader (373)	Moves the insertion point to the next header container within the current context of the document.
\$LwpGoToLocationFooter (374)	Moves the insertion point to the next footer container within the current context of the document.
\$LwpGoToLocationFrame (375)	Moves the insertion point to the next frame container within the current context of the document.
\$LwpGoToLocationDivision (376)	Moves the insertion point to the next division container within the current context of the document.
\$LwpGoToLocationExit (377)	Moves the insertion point out of the next container within the current context of the document.

p2

This optional Variant parameter specifies the string name of a page container or a division container.

Return value

The return value for this method will always be -1 or 0. When testing the return value, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Word Pro: GoToLayout method

```
{button ,AL('H_WPAPPLICATION_CLASS;H_LAYOUT_CLASS;H_CELLGROUPLAYOUT_CLASS;H_CELLLAYOUT_CLASS;H_CONNECTEDLAYOUT_CLASS;H_COLUMNGROUPLAYOUT_CLASS;H_FOOTERLAYOUT_CLASS;H_FRAMELAYOUT_CLASS;H_FRAMEGROUPLAYOUT_CLASS;H_DROP_CAPLAYOUT_CLASS;H_GROUPLAYOUT_CLASS;H_HEADERLAYOUT_CLASS;H_NOTELAYOUT_CLASS;H_PAGELAYOUT_CLASS;H_ROWGROUPLAYOUT_CLASS;H_ROWLAYOUT_CLASS;H_RUBYLAYOUT_CLASS;H_SUPERTABLEGROUPLAYOUT_CLASS;H_SUPERTABLELAYOUT_CLASS;H_TABLELAYOUT_CLASS;H_ENDNOTELAYOUT_CLASS;H_FOOTNOTELAYOUT_CLASS;H_TABLEHEADINGLAYOUT_CLASS;H_TOCSUPERTABLELAYOUT_CLASS',0)} See list of classes
```

```
{button ,AL('H_GOTOLAYOUT_METHOD_EXSCRIPT',1)} See example
```

Moves the insertion point to a specific layout object.

Syntax

When called from WPAApplication:

```
[objectreference].GoToLayout(Name)
```

When called from any other object:

```
[objectreference].GoToLayout()
```

Parameters

Name

Only used when you call this method from the WPAApplication object. Provides the name of the layout object to which you want to move the insertion point; name must be the same as that found in the Name property of the layout object. Data type is String.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

This method is available on most objects which are created from the Layout class or one of its child classes. When you call this method from one of these objects, Word Pro moves the insertion point into the object from which you called the method.

When you call this method from the WPAApplication object, you must identify the layout you want to go to by providing the layout's name in the Name parameter.

Word Pro: GotoNextParallelColumn method

{button ,AL('H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL('H_GOTONEXTPARALLELCOLUMN_METHOD_EXSCRIPT',1)} [See example](#)

Moves the insertion point to the beginning of the parallel column block to the right of the current parallel column. If the insertion point is already in the last parallel column, nothing happens.

Syntax

[objectreference].GotoNextParallelColumn()

Parameters

None.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Calling this method is equivalent to choosing Goto Next Column Block from the Columns menu.

Word Pro: GoToObject method

{button ,AL('H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL('H_GOTOOBJECT_METHOD_EXSCRIPT',1)} [See example](#)

Moves the insertion point to the next object of the type you specify.

Syntax

[objectreference].GoToObject(ObjectName, Forward)

Parameters

ObjectName

A String expression which specifies the type of the object to which you want to go. The legal values for this parameter are listed below:

Align	HardSpace	PageNumber
AnchoredFrame	Indent	PowerField
BookMark	Index	RulerMark
ColumnBreak	Layout	SoftHyphen
DDE	LineBreak	Spacing
DocVariable	Note	Table
FootnoteMark	PageBreak	TOC
Frame		

Forward

A Numeric expression which allows you to specify whether you want to go to an object behind or in front of the insertion point. Data type is Integer. The legal values for this parameter are -1 and 0 but you may use the LotusScript constants True (-1) and False (0) instead of the integer values. Optional parameter. Default is True.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Word Pro: GoToPage method

{button ,AL('H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL('H_GOTOPAGE_METHOD_EXSCRIPT',1)} [See example](#)

Moves the insertion point to the specified page number.

Syntax

[objectreference].GoToPage(PageNumber)

Parameters

PageNumber

An Integer representing the page number.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Word Pro: GoToSection method

{button ,AL(^H_WPAPPLICATION_CLASS;H_SECTION_CLASS;H_INDEXSECTION_CLASS',0)} [See list of classes](#)

{button ,AL(^H_GOTOSECTION_METHOD_EXSCRIPT',1)} [See example](#)

Moves the insertion point to a specified section of a division.

Syntax

[objectreference].GoToSection([SectionGoTo])

Parameters

SectionGoTo

Data type is Variant. Optional parameter. The value of this parameter must be one of the strings below or its code equivalent.

\$LwpSectionGoToBeforeSectionMarker (1746)

\$LwpSectionGoToEndOfSection (1747)

Return value

Usage

Word Pro: GoToTableCell method

{button ,AL(^H_BASSETABLE_CLASS;H_FOOTNOTETABLE_CLASS;H_PARALLELCOLUMNS_CLASS;H_GLOSSARY_CLASS;H_TABLEHEADING_CLASS;H_TABLE_CLASS',0)} [See list of classes](#)

{button ,AL(^H_GOTOTABLECELL_METHOD_EXSCRIPT',1)} [See example](#)

Moves the insertion point to a specific location within the table.

Syntax

[objectreference].GoToTableCell(CellObjectType, [Next])

Parameters

CellObjectType

Indicates the specific cell to which you want to move the insertion point. Data type is Variant. The value of this parameter must be the string constant below or its numeric equivalent.

<u>Value</u>	<u>Effect</u>
\$LwpCellObjectType (117)	The CellObjectType parameter must be this value.

Next

Indicates whether the focus should be moved to the next location within the table. Data type is Integer. The legal values are -1 (True) or 0 (False). Optional parameter. Default is True. If Next is set to False, the method will do nothing and return a value of 0.

Return value

This method returns an Integer value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Assuming the Next parameter is set to True, the following table describes possible initial insertion point states and what the result of the GoToTableCell method will be in that situation. If the Next parameter is set to False, the GoToTableCell method will do nothing and return a value of 0.

<u>Cursor located in the last cell of a row</u>	<u>Cursor at the end of the text within the cell</u>	<u>Cursor located in last cell of the table.</u>	<u>Result of the GoToTableCell method</u>
No	Doesn't matter	Doesn't matter	Cursor moves to the next cell.
Yes	No	Doesn't matter	Cursor moves to the end of the text within the cell.
Yes	Yes	No	Cursor moves to the next cell.
Yes	Yes	Yes	New row is appended to the table, and the cursor moves to the first cell of the new row.

Word Pro: GoTo method

{button ,AL(`H_FOOTNOTE_CLASS;H_MARKER_CLASS;H_POWERFIELD_CLASS;H_CLICKHERE_CLASS;H_TEXTMARKER_CLASS;H_RUBYMARKER_CLASS;H_TABLEMARKER_CLASS',0)} [See list of classes](#)

{button ,AL(`H_GOTO_METHOD_EXSCRIPT',1)} [See example](#)

Moves the insertion point to the footnote or marker object from which this method is called.

Syntax

[objectreference].GoTo(SelectAll)

[objectreference].GoTo()

[objectreference].GoTo()

Parameters

SelectAll

Data type is Integer. The legal values for this parameter are -1 and 0 but you may use the LotusScript constants True (-1) and False (0).

Return value

The return value for this method will always be -1 or 0. When testing the return value, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Word Pro: GroupDivision method

{button ,AL('H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL('H_GROUPDIVISION_METHOD_EXSCRIPT',1)} [See example](#)

Creates a parent division for the division specified in the DivisionName parameter. Equivalent to clicking the right mouse button on a division tab and selecting Group Tabs.

Syntax

[objectreference].GroupDivision([DivisionName])

Parameters

DivisionName

A String expression which specifies the internal name of the division you want to place in the new parent division. If you do not provide a value for this parameter, the currently active division is placed in the new parent division.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

For more information on division names in LotusScript, see [Overview: Division names in LotusScript](#)

Word Pro: HandsOffStorage method

{button ,AL(^H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

Causes Word Pro to release all references to an embedded Word Pro object's IStorage.

Syntax

[objectreference].HandsOffStorage()

Parameters

None.

Return value

None.

Usage

Typically used internally by Word Pro during an OLE operation. You may need this method if you write a script in which you access a Word Pro document that is stored as an OLE object in another application's document (such as a Word Pro document stored in a 1-2-3 worksheet). In such a script, the HandsOffStorage method instructs Word Pro to release its connection to the IStorage in which the Word Pro OLE object is stored.

Note OLE is not supported under OS/2.

Word Pro: HasNamedProperty method

```
{button ,AL(^H_CHARACTERSTYLE_CLASS;H_DIVISION_CLASS;H_LAYOUT_CLASS;H_CELLGROUPLAYOUT_CLASS;H_CELLLAYOUT_CLASS;H_CONNECTEDLAYOUT_CLASS;H_COLUMNGROUPLAYOUT_CLASS;H_FOOTERLAYOUT_CLASS;H_FRAMELAYOUT_CLASS;H_FRAMEGROUPLAYOUT_CLASS;H_DROPCAPLAYO  
UT_CLASS;H_GROUPLAYOUT_CLASS;H_HEADERLAYOUT_CLASS;H_NOTELAYOUT_CLASS;H_PAGELAYO  
UT_CLASS;H_ROWGROUPLAYOUT_CLASS;H_ROWLAYOUT_CLASS;H_RUBYLAYOUT_CLASS;H_SUPERTAB  
LEGROUPLAYOUT_CLASS;H_SUPERTABLELAYOUT_CLASS;H_TABLELAYOUT_CLASS;H_ENDNOTELAYO  
UT_CLASS;H_FOOTNOTELAYOUT_CLASS;H_TABLEHEADINGLAYOUT_CLASS;H_TOCSUPERTABLELAYOUT  
_CLASS;H_MARKER_CLASS;H_POWERFIELD_CLASS;H_CLICKHERE_CLASS;H_TEXTMARKER_CLASS;H_  
RUBYMARKER_CLASS;H_TABLEMARKER_CLASS;H_TEXTDOCUMENT_CLASS',0)} See list of classes
```

```
{button ,AL(^H_HASNAMEDPROPERTY_METHOD_EXSCRIPT',1)} See example
```

Checks for the presence of a named property within the object from which this method is called.

Syntax

```
[objectreference.]HasNamedProperty(PropertyName)
```

Parameters

PropertyName

The name of the property that you want to check for within an object. Data type is String.

Return value

The return value for this method will always be 1 or 0. A return value of 1 indicates that the specified object does have a named property that matches the *PropertyName* parameter.

Usage

A named property is a user-defined property assigned to an object. Unlike variables, named properties are persistent. They continue to exist when a script stops executing, and when a document is closed and reopened.

Use this method to determine whether or not a specific named property exists on an object. For example, if you call this method from a ClickHere object, Word Pro checks that ClickHere object for the named property specified in the *PropertyName* parameter. Use the *GetNamedProperty* method to retrieve the value stored in a named property.

Word Pro: Help method

{button ,AL('H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL('H_HELP_METHOD_EXSCRIPT',1)} [See example](#)

Launches the specified Help file.

Syntax

[objectreference].Help ([HelpFile][, HelpContextID][, HelpString][, Parent])

Parameters

HelpFile

A String expression which specifies the name of the Help file you want to open. Optional parameter. If you do not provide a value for this parameter, Word Pro opens the appropriate Help file in the current context.

HelpContextID

A Numeric expression that allows you to specify which topic you want to turn to, by its resource code or Help panel ID number.

HelpString

A String expression that allows you to specify which topic you want to turn to by the topic's unique context ID string. For example, the context ID for this topic is "H_HELP_METHOD_MEMDEF."

Parent

A Numeric expression of type Long, which allows you to specify the parent. Required for use with an OS/2 Help file.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Word Pro: HideCaretAndSelection method

{button ,AL('H_CLICKHERE_CLASS;H_TEXTMARKER_CLASS;H_TEXT_CLASS',0)} [See list of classes](#)

Hides the caret (the blinking insertion point) and the shading which marks a selection.

Syntax

[objectreference].HideCaretAndSelection([HideCaret][, HideSelection])

Parameters

HideCaret

Allows you to hide the caret (-1) or leave it visible (0). Data type is Integer. The legal values for this parameter are -1 and 0 but you may use the LotusScript constants True (-1) and False (0) instead of the integer values. Optional parameter. Default is -1 (True).

HideSelection

Allows you to hide the shading which indicates a selection (-1) or leave it visible (0). Data type is Integer. The legal values for this parameter are -1 and 0 but you may use the LotusScript constants True (-1) and False (0) instead of the integer values. Optional parameter. Default is -1 (True).

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Word Pro: HideIconBar method

{button ,AL('H_ICONBAR_CLASS',0)} [See list of classes](#)

{button ,AL('H_HIDEICONBAR_METHOD_EXSCRIPT',1)} [See example](#)

Hides an icon bar set temporarily until the next context change. Equivalent to hiding a bar from the drop down menu that appears when you click on the Close box.

Syntax

[objectreference].HideIconBar()

Parameters

Data type is Integer. The legal values for this parameter will always be -1 or 0 but you may use the LotusScript constants of True (-1) and False (0).

Return value

The return value for this method will always be -1 or 0. When testing the return value, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

To hide an icon bar object permanently, use the ShowInContext property.

Word Pro: Hide method

```
{button ,AL(^H_WPAPPLICATION_CLASS;H_BASECONTAINER_CLASS;H_CELLCONTAINER_CLASS;H_FRAME  
CONTAINER_CLASS;H_NOTECONTAINER_CLASS;H_RUBYCONTAINER_CLASS;H_DROPDOWNCONTAINER_  
CLASS;H_PAGECONTAINER_CLASS;H_SUBPAGECONTAINER_CLASS;H_SUPERPAGECONTAINER_CLASS  
;H_ROWCONTAINER_CLASS;H_SUPERTABLECONTAINER_CLASS;H_TABLECONTAINER_CLASS;H_PARAL  
LELCOLSCONTAINER_CLASS;H_TABLEONLYCONT_CLASS;H_NOTELAYOUT_CLASS;H_DOCWINDOW_CL  
ASS',0)} See list of classes
```

Reduces a specific note layout object to an icon.

Syntax

[objectreference].Hide()

Parameters

Return value

The return value for this method will always be -1 or 0. When testing the return value, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

This method applies only to note container objects.

How do I create a document with divisions?

Creating documents and divisions within new documents is straightforward in Word Pro. The document and division objects support the Create method with numerous options. The following example illustrates how to create a named document and a named division within that new document.

The following sub, named CreateDocAndDiv, calls two worker subs and specifies names for the new document and division.

```
Sub CreateDocAndDiv
    ' Call the CreateDoc sub to create a document named "WEEK6.LWP".
    CreateDoc "WEEK6.LWP"

    ' Call the CreateDiv sub to create a division named "MondaySales"
    ' in the new document.
    CreateDiv "MondaySales"
End Sub
```

The CreateDoc sub takes the name of the new document as a parameter and passes it to the NewDocument method. You can add other parameters to specify a particular file path or SmartMaster.

```
Sub CreateDoc(NewDocName as String)
' * RUNTIME DEPENDENCIES
' *   Files and paths: You must have the SmartMaster LETTER1.MWP in
'   the subdirectory C:\LOTUS\SMARTERS\WORDPRO.

' Create the document with the following options:
'   Document name = the value of parameter NewDocName
'   File path = "C:\DATA\WORDPRO\"
'   SmartMaster = "C:\LOTUS\SMARTERS\WORDPRO\LETTER1.MWP"
.NewDocument NewDocName, _
    "C:\DATA\WORDPRO\", _
    "C:\LOTUS\SMARTERS\WORDPRO\LETTER1.MWP", "", "", ""

' Save the new document.
.Save
End Sub
```

The CreateDiv sub takes the name of the new division as a parameter and passes it to the CreateDivision method. You can add parameters to specify the style of the new division and its relative position in the current sequence of divisions.

```
Sub CreateDiv(NewDivisionName as String)
    ' Declare a path for the Division SmartMaster.
    Dim SMasterFullPath As String
    ' Declare a variable for the new division.
    Dim MyNewDivision As String

    ' Get the file path for the division SmartMaster.
    SMasterFullPath = _
        .ApplicationWindow.UserInterfacePrefs.StylePath _
        & "\DEFAULT.MWP"

    ' Create the division.
    MyNewDivision = _
        .CreateDivision(SMasterFullPath, "", _
            $LWPDIVLocInsertAfterCurrentDiv, "", "")

    ' Name the new division. The division tab reflects the new name.
    .Division.DivisionInfo.Name = "NewDivisionName"

End Sub
```


How do I create and use a custom dialog?

The process of creating and using custom dialog boxes in Word Pro should be familiar if you have used Microsoft Visual Basic or IBM Visual Age:

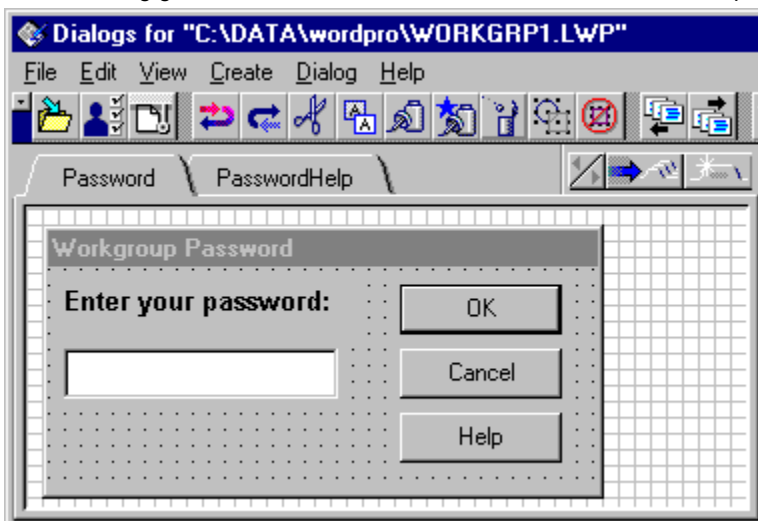
- Create the dialog using the Lotus Dialog Editor.
- Add controls to the dialog using the Lotus Dialog Editor.
- Write scripts for the controls using the Lotus Integrated Development Environment (IDE).
- Run the scripts and display the dialog in Word Pro.

Custom dialogs are stored in Word Pro documents along with other scripts in your application.

Creating a custom dialog

To open the Lotus Dialog Editor for your current document, choose Edit - Script & Macros - Show Dialog Editor in Word Pro.

The Dialog Editor provides a tabbed panel for each dialog in your current document. The following example uses two dialogs, one for a user to enter a password and another to display some Help text about the password dialog. The second dialog gets invoked from the first, if the user clicks on the Help command button.



Use the InfoBox for the dialog to set properties such as its name, title bar caption, and Help context ID.

To create additional dialogs, choose Create - Dialog in the Dialog Editor.

Note Modal and modeless dialogs are not differentiated at design time in the Dialog Editor. You determine whether the dialog is modal or modeless when you call the dialog from your script.

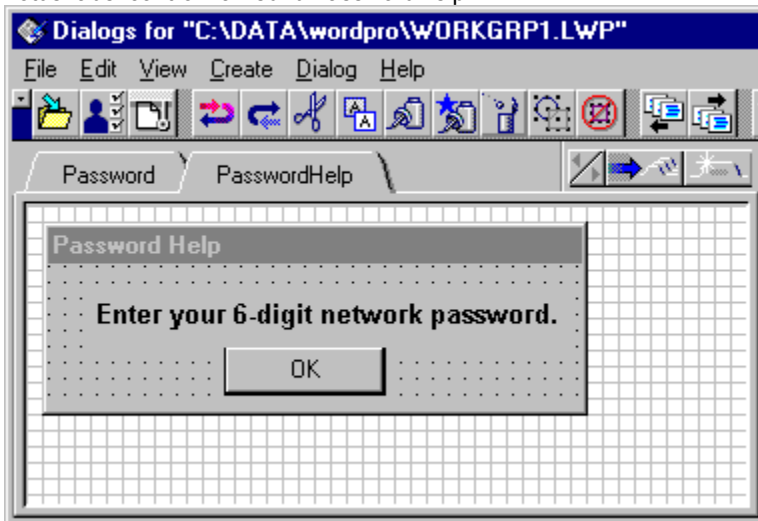
Adding controls to the dialog

The Dialog Editor is an OLE container that supports OLE Custom (OCX) Controls developed by Lotus and other control vendors. The Dialog Editor provides 13 controls and displays icons for each control in the control toolbox.



To add one of these controls to your dialog, click its icon in the toolbar and size the control in the dialog panel. The

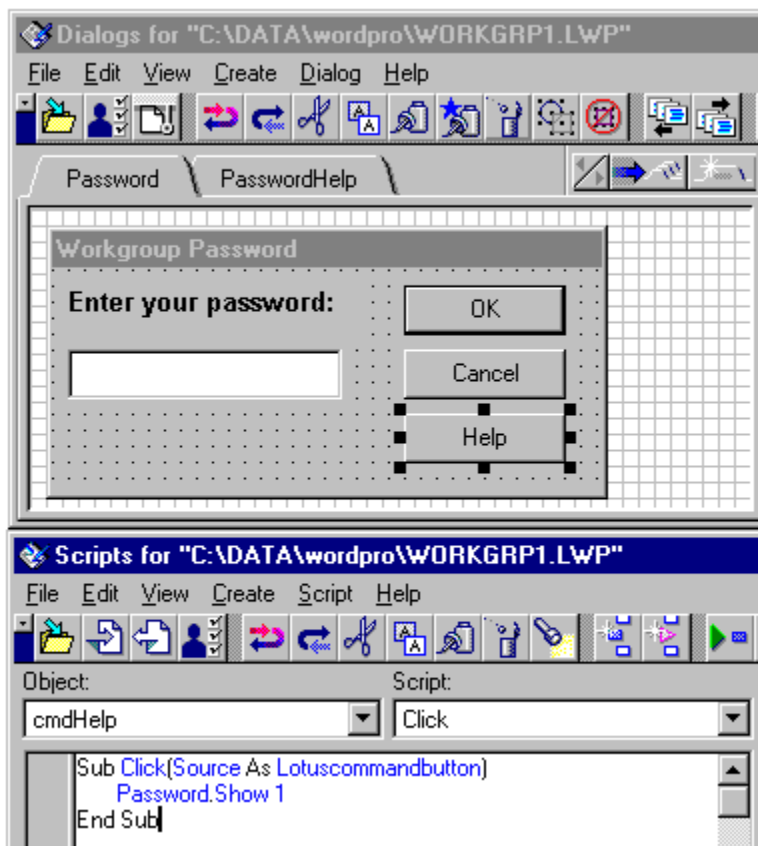
second dialog below ,named PasswordHelp, has two controls: a LotusCommandButton control named cmdOK and a LotusLabel control named lblPasswordHelp.



Use the InfoBox for each control to determine text colors, fonts, borders, names, captions, and default values in lists. **Tip** To add third party controls to your dialogs, choose Create - Control - More ... in the Dialog Editor and select from among the OLE controls registered on your system.

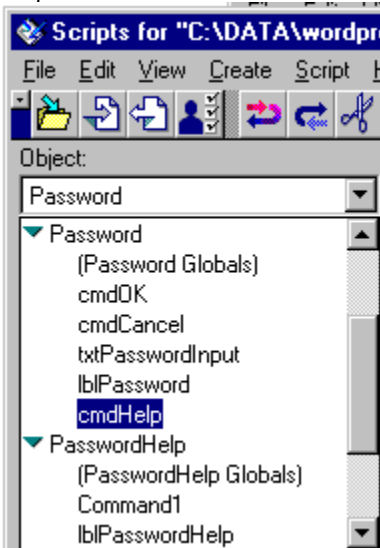
Writing scripts for controls

The Dialog Editor and the Integrated Development Environment (IDE) are closely integrated. To write scripts for a particular control in your dialog, double-click that control. Word Pro displays the IDE and navigates to the default event procedure for the selected control in your dialog. The following illustration shows the control named cmdHelp selected in the Password dialog and its corresponding default event procedure, Click, in the IDE.

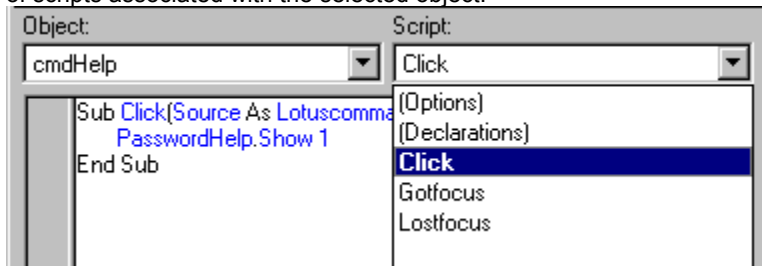


As you add dialogs and controls in dialogs, the IDE adds their names to the list of scriptable objects in its Object

drop-down box.



Click the object in the drop-down list to select it for scripting. Click the Script drop-down box in the IDE to display a list of scripts associated with the selected object.



In this case, the Click procedure for the cmdHelp button contains the statement required to display the dialog named PasswordHelp. When the user clicks the button labeled Help in the Password dialog, the second dialog named PasswordHelp will display.

To process user input in a control, use properties associated with that control. The LotusTextBox control in the Password dialog is named txtPasswordInput and Text property to store its values. To get the current value of the txtPasswordInput control in a script, use the following statements:

```
Dim PasswordValue as String
PasswordValue = Password.txtPasswordInput.Text
Print PasswordValue
```

You can then use the value of PasswordValue to perform validation routines.

Running the dialog from your application

To call a dialog from your scripts in Word Pro, you must use methods for the dialog object. The following example illustrates some of the methods for managing dialogs at runtime.

```
Sub DialogFireDrill
    ' Display the dialog as a modal.
    Password.Show 1
    ' Close the dialog.
    Password.Close
    ' Display the dialog as a modeless.
    Password.Show
    ' Hide the dialog temporarily.
    Password.Hide
    ' Redisplay the dialog.
    Password.Show
End Sub
```


How do I call DLL functions?

Your applications are not limited to calling procedures developed in the LotusScript language. If you have developed procedures in a language such as C, C++, or Visual Basic, and compiled them in a dynamic-link library (DLL), you can call these procedures from within your script application.

To call procedures in a DLL, you need to know the following:

- the name of the DLL
- the file path for the DLL (if it is not on your default path)
- the names and parameters for procedures you want to call

The following example illustrates how to call a Win32 API function named `sndPlaySound` that is stored in the DLL file `C:\WINDOWS\SYSTEM\WINMM.DLL`. To play a Windows .WAV file in your script application using this Win32 function, you must first declare the function and then call it from one of your scripts.

Note You can copy the entire example into the IDE. The Script Editor automatically moves declarations and the sub into the appropriate sections of the script.

```
' Put the following statements in (Declarations) for (Globals) if you
' want to be able to call .WAV files from any script in your application.

' * RUNTIME DEPENDENCIES
' *   Files and paths: WINMM.DLL must be installed in C:\WINDOWS\SYSTEM
' *   or somewhere in your current file path. The sound file
' *   OFF2RACE.WAV must be installed in the subdirectory
' *   C:\WINDOWS\MEDIA.

' Declare a return value to use when you call the DLL
' function in your scripts.
Dim SoundReturnValue As Integer

' Declare the DLL function as a public function in LotusScript.
Declare Public Function sndPlaySound Lib "winmm" _
    Alias "sndPlaySoundA" _
    ( Byval WaveFile As String, Byval theFlags As Long ) _
    As Integer

' Declare some of the constants used by parameters of the DLL function.
Public Const SND_SYNC      = &H0000 ' Play synchronously (default)
Public Const SND_ASYNC     = &H0001 ' Play asynchronously
Public Const SND_NODEFAULT = &H0002 ' Silence (!default) if not found
Public Const SND_MEMORY    = &H0004 ' pszSound points to a memory file
Public Const SND_LOOP      = &H0008 ' Loop until next sndPlaySound
Public Const SND_NOSTOP    = &H0010 ' Don't stop currently playing sound

Sub TestSoundFiles
    ' The sub calls the declared function and specifies a .WAV file to play.
    SoundReturnValue = _
        sndPlaySound( "C:\WINDOWS\MEDIA\OFF2RACE.WAV", SND_SYNC )
End Sub
```

How do I get information about documents?

Word Pro maintains detailed information about each of the documents that you use. While you are writing a document, choose File - Document Properties - Document to view all the information that Word Pro maintains for it. This information is also available to your scripts in the form of DocInfo properties. The current size of your file, for example, is available in the property DocInfo.DocSize.

The following example illustrates how to extract and use DocInfo property information for documents on disk and documents in memory. If you work with a collection of documents, you can build a report summarizing information about each document in the library by opening each document and collecting DocInfo information about it. The sub DocInfoProfiles passes the name of a document to profile to the sub BuildDocInfoProfile. All output from BuildDocInfoProfile is stored in the current document; no documents being profiled are modified.

```
Sub DocInfoProfiles
' * RUNTIME DEPENDENCIES
' *   Files and paths: The document GRP1STAT.LWP must be in the
' *     subdirectory C:\LOTUS\WORK\WORDPRO.

' Call the worker sub that loads the specified document and builds a
' profile of its DocInfo properties.
BuildDocInfoProfile "C:\LOTUS\WORK\WORDPRO\GRP1STAT.LWP"
' Profile a second document.
' BuildDocInfoProfile "YOURDOC.LWP"
End Sub

Sub BuildDocInfoProfile(DocName As String)
' Declare some variables to hold DocInfo information for each document.
Dim DocAuthor As String
Dim DocCreationDate As String
Dim DocPageCount As Long
Dim DocDescription As String

' Declare a variable for DocInfo values displayed in a message box.
Dim MyMsgText as String

' Open the document specified in the parameter DocName.
.OpenDocument DocName, "", "", "", False, True
' Do not close replace the current document with the one opened.
.ApplicationWindow.UserInterfacePrefs.IsReplacement = False

' Extract some DocInfo values for the document just opened.
With .ActiveDocument
    DocName = .FullName
    DocAuthor = .Docinfo.AuthorName
    DocCreationDate = .DocInfo.CreationDateString _
        & " at " & .DocInfo.CreationTimeString
    DocPageCount = .DocInfo.NumPagesInDoc
    DocDescription = .DocInfo.Description
End With

' Close the current document before writing any profile information.
.Close

' Display profile results in a message box.
' Comment out if you don't want the sub to pause.
MyMsgText = "DocInfo: " _
    & DocName _
    & DocAuthor _
    & DocCreationDate _
    & DocPageCount
```

```
' Write the profile information to the end of the current document.
' Go to the end of the current document. Write a little record for
' each document profiled.
.Type "[ctrlEnd]"
.Type "[ENTER]======"
.Type "[ENTER]Document Name:[TAB][TAB]"
.Type DocName
.Type "[ENTER]Document Author:[TAB][TAB]"
.Type DocAuthor
.Type "[ENTER]Document Creation Date:[TAB]"
.Type DocCreationDate
.Type "[ENTER]Document Page Count:[TAB]"
.Type Str(DocPageCount)
.Type "[ENTER]=====[ENTER]"
.Type DocDescription
.Type "[ENTER]=====[ENTER]"
End Sub
```

Here is sample output from the BuildDocInfoProfile sub:

```
=====
Document Name:          C:\LOTUSWORK\WORDPRO\GRP1STAT.LWP
Document Author:       James T. O'Connell
Document Creation Date: 6/14/96 at 1:38PM
Document Page Count:   3
=====
```

How do I add graphics to my document?

You can enhance your document by adding graphic elements such as diagram frames, imported pictures, and tables.

Adding frames, tables, and and bitmaps

The following example illustrates how to add a frame for diagrams, a table, and a bitmap image to your current document.

```
Sub CreateGraphics
' * RUNTIME DEPENDENCIES
' *   Files and paths: You must have the bitmap file HELPBTN.BMP in the
' *       subdirectory C:\LOTUS\WORDPRO.

' Declare a text variable for message box output.
Dim MsgText as String

' Declare a Layout object for the new diagram frame.
Dim MyFrame As Layout

' Declare a text variable for the name and file
' path of a bitmap file to import.
Dim BMPName As String
' Specify a bitmap file to import.
BMPName = "C:\LOTUS\WORDPRO\HELPBTN.BMP"

' Enter some blank lines at the top of the document.
.Type "[ctrlHome][Enter][Enter][Enter][Enter][Enter]"
.Type "[Enter][Enter][Enter][Enter]A new frame.[Enter]"

' Create a new 1" x 1" diagram frame at the cursor.
.CreateFrame False, "Default Frame", 1440, 1440

' Anchor the frame to the preceding paragraph.
.Frame.Anchor 0, $LwpConditionTypeAllPages, _
    $LwpRelativeTypeLytInlineNewline

' Bind the layout variable MyFrame to the new frame.
' The default name for the first frame you create is Framel.
Set MyFrame = Bind("!Body:Framel")
' Display the name of the new diagram frame.
MsgText = "Created " & MyFrame.Name
MsgBox MsgText

' Enter some blank lines at the top of the document.
.Type "[ctrlHome][Enter][Enter][Enter][Enter][Enter]"
.Type "[Enter][Enter][Enter][Enter]A new table.[Enter]"

' Create a table with the following options:
'   Table style = "Default table"
'   Number of columns = 5
'   Number or rows = 6
.CreateTable False, "Default Table", 5, 6

' Enter some blank lines at the top of the document.
.Type "[ctrlHome][Enter][Enter][Enter][Enter][Enter]"
.Type "[Enter][Enter][Enter][Enter]A new bitmap.[Enter]"

' Create a frame at the cursor that contains
' a bitmap image (".bmp" format) imported from a disk file.
.ImportGraphic BMPName, ".bmp", False, False, "INTERNAL_ID"
```

```

' Anchor the frame to the preceding paragraph.
.Frame.Anchor 0, $LwpConditionTypeAllPages, _
  $LwpRelativeTypeLytInlineNewline
' Assign a name to the new frame.
.Layout.Name = "MyBMPFrame"
End Sub

```

Tip It is useful to name objects at the time you create them, because you can subsequently select named objects in a collection by iterating through collection members.

Navigating to graphic objects

The following example illustrates how to navigate to a frame named "MyBMPFrame".

```

Sub GoToBMP
' * RUNTIME DEPENDENCIES
' *   Objects: You must have a frame named MyBMPFrame in your document.

Dim TargetFrame As Layout
Set TargetFrame = Bind("!Body:MyBMPFrame")
TargetFrame.GotoLayout
End Sub

```

Manipulating graphics

You may need to modify the placement or appearance of a graphic. The following example illustrates how to create a frame, change some properties for it, and then change its placement on each page of the document.

```

Sub MoveBannerGraphic
' Declare a Layout object for a new diagram frame.
Dim BannerGraphic as Layout

' Create a 1" x 1" frame at the insertion point.
' Use current defaults for frames.
.CreateFrame False, "Default Frame", 1440, 1440

' Assign a name for the new frame.
.Frame.Layout.Name = "NewTestFrame"
' Assign the variable BannerGraphic to the new frame.
Set BannerGraphic = Bind("!Body:NewTestFrame")

' Set some frame properties: the graphic should appear on every
' page and its placement on each page should be relative
' to page margins (versus paragraphs or columns surrounding it).
.Frame.Anchor $LwpAnchorWhereLayout, _
  $LwpConditionTypeAllpages, _
  $LwpRelativeTypeLytParent
.Frame.Layout.WrapType = $LwpWrapTypeLayoutNoWrapAround

' Place the frame 2" below the top margin on each page.
.Frame.Layout.RelativeYDistance = 2880
' Place the frame 3" from the right margin on each page.
.Frame.Layout.RelativeXDistance = 3960

End Sub

```

How do I navigate to objects in a document?

There are several ways to select or navigate to objects in a document:

- searching collections for named objects
- using GoTo methods for named objects
- searching for text

Selecting or activating named objects

If you opened more than one document in your script application, you can activate a named document by working with one of the collection objects in Word Pro, called DocWindows. Word Pro builds collections for many types of objects: bookmarks, document windows, cells in tables, divisions, footnotes, glossary items, layout objects, and document versions.

The following example illustrates the basic process of navigating to a named object in a collection by iterating through members of the collection.

```
Sub ActivateDocByName
    ' Get the name of an active document.
    Dim DocumentName as String
    DocumentName = Inputbox$("Please Enter the Name of the Document to show")
    ' Iterate through the members of the DocWindows collection.
    Forall myDoc In .ApplicationWindow.DocWindows
        ' Find the name of the active document and activate the
        ' document window.
        If myDoc.Name = DocumentName Then
            myDoc.Show
        End If
    End Forall
End Sub
```

Navigating to named objects with a GoTo method

Another method of navigating to named objects is more direct. The following example illustrates how to navigate to a named frame and a named bookmark, using GoTo methods.

Note The names of bookmarks in Word Pro are case sensitive.

```
Sub GoToNamedObjects
    ' Declare a layout variable for the named frame.
    Dim MyFrame As Layout

    ' Create a named frame.
    .CreateFrame False, "Default Frame", 1440, 1440
    .Frame.Layout.Name = "TestFrame"

    ' Assign the layout variable to the named frame.
    Set MyFrame = Bind("!Body:TestFrame")

    ' Execute a goto method on the named frame.
    MyFrame.GotoLayout

    ' An alternative way to go to the named frame using a goto method.
    .GoToLayout "Body:TestFrame"

    ' Create a bookmark named "ReadThisKyle"
    TEMP__ = Mark($LwpMarkerTypeBookmark)
    .Division.Foundry.Markers(TEMP__).PageNumber = 1
    .Division.BookmarkManager.AddBookmark "ReadThisKyle", TEMP__

    ' Go to the bookmark named "ReadThisKyle".
    .GoToBookmark("ReadThisKyle")
End Sub
```


Although some objects do not have a persistent name, you can use a GoTo method to navigate to them. The following example illustrates how to navigate to a specific page by page number or by its position in the file (first or last).

```
Sub GoToPage
' * RUNTIME DEPENDENCIES
' *   Objects: There must be three or more pages in your document.

'   Go to page three in the current document.
.GoToPage 3
'   Go to the last page in the current document.
.ApplicationWindow.UserInterfacePrefs.GoToSelection = _
    $LwpGoToTypeLastpage
.GoToPage 32767
End Sub
```

Navigating to text with search

Individual words, sentences, and paragraphs in your documents do not have unique names or identifiers. One way to support navigation through passages in your document is to create named bookmarks at specific places; another is to search for unique text within passages.

The following example illustrates how to copy a paragraph containing a keyword, "Bennings Electronics", from the body of your document to the end of your document.

```
Sub ExtractKeywordParagraph
' * RUNTIME DEPENDENCIES
' *   Objects: There must be at least one occurrence of the string
' *       "Bennings Electronics" in your document.

'   Search for the first instance of the keyword.
.Application.FindAndReplace.FindString = "Bennings Electronics"
.InitFindAndReplace True
.Find
'   Select the entire paragraph containing the keyword.
.SelectParagraph
'   Copy the selected paragraph.
.CopySelection
.InitFindAndReplace True
'   Go to the end of the current document.
.Type "[ctrlEnd]"
'   Paste the copied paragraph.
.Type "[ENTER]=====[ENTER]"
.Paste
End Sub
```

How do I use OLE objects?

As an OLE container, Word Pro lets you create and automate a variety of OLE objects in your documents. There are three types of OLE objects that you can create and automate in Word Pro:

- Embedded OLE 2 objects
- Embedded files as OLE objects
- Embedded OLE Custom Controls (OCX)

Embedding OLE 2 objects in your document

The following example illustrates how to create OLE 2 objects in your Word Pro document.

```
Sub EmbedObject
    ' Embed a Paintbrush object.
    .CreateOleNew "PBrush", 0, True
    ' Embed a Lotus Approach application object.
    .CreateOleNew "ApproachApplication", 0, True
End Sub
```

Tip The names of the OLE servers that you can specify in your CreateOleNew statement are available in the Windows Registry in HKEY_CLASSES_ROOT.

Embedding files as OLE objects

You can also add to your Word Pro document OLE objects instantiated from existing documents on disk. Unlike the CreateOleNew method described above, the OLE objects created in the following example contain data.

```
Sub EmbedFile
    ' Embed an Approach database file.
    .CreateOleEmbeddedFile _
        "{00000000-0000-0000-0000-000000000000}", _
        "C:\LOTUS\WORK\APPROACH\TEST.APR", 0
End Sub
```

Embedding OLE Custom Controls

You can add registered OLE Custom Controls to your Word Pro document and develop scripts for those controls, as you would any custom control in the Dialog Editor.

```
Sub EmbedOCX
    ' Embed the OLE control named LotusCommandButton.
    .CreateOleNew "Lotus.CommandButton.1", 0, True
End Sub
```

How do I call Windows applications?

You can call Windows applications and services from your Word Pro script applications with the LotusScript Shell() function. To make a basic call, you must know the name and path of the Windows application, for example C:\WINDOWS\IMPLAYER.EXE. More sophisticated calls require knowing the command line parameters for the application that would allow you to pass information from your script application to the Windows application. The MPLAYER.EXE application will load a multimedia file if you specify one as a parameter in your call, for example C:\WINDOWS\IMPLAYER TADA.WAV.

The following example demonstrates how to call a Windows Help file or particular topics within a custom Help file that you develop for your application, using the Microsoft Windows help compiler. Once you have a script that can call Windows Help topics, you can add context-sensitive help to many objects in your own script application.

```
Sub CallHelp
' * RUNTIME DEPENDENCIES
' *   Files and paths: The Word Pro LotusScript help file WP071EN.HLP
' *       must be in the subdirectory C:\LOTUS\WORDPRO.

' Declare a return variable for the Shell() function.
Dim HelpReturnValue As Integer

' Declare a variable for the name and path of the Help file.
Dim HelpFilename As String
' Specify the name and path of the Help file.
HelpFileName = "C:\LOTUS\WORDPRO\WP0N71EN.HLP"

' Display the Contents of a help file named WP0N71EN.HLP.
HelpReturnValue = Shell("WINHLP32.EXE " & HelpFileName, 1)

' Call a topic in WP0N71EN.HLP named "The Help Method" identified by its
' the help context ID H_HELP_METHOD_MEMDEF. The command-line switch -I
' specifies that WinHelp search for the specified context ID.
HelpReturnValue = Shell("WINHLP32.EXE -I H_HELP_METHOD_MEMDEF & HelpFileName, 1)

' Call the same topic in WP0N71EN.HLP identified by its resource ID
' number 20105. These resource IDs appear in the header file that you
' compile with the help file. The command-line switch -N specifies that
' WinHelp search for the specified resource ID.
HelpReturnValue = Shell("WINHLP32.EXE -N 20105 " & HelpFileName, 1)
End Sub
```

Note Word Pro also supports a Help method that lets you call Help files or specific Help topics within a Help file.

How do I manage find and replace?

Word Pro provides sophisticated Find and Replace features that you can incorporate in your script applications. You can perform Find and Replace at three levels:

- against text in your documents
- against named styles in your document
- against strings that you extract from your document

Finding and replacing text

The following example illustrates how to do a global Find and Replace of text.

```
Sub ReplaceList
' * RUNTIME DEPENDENCIES
' *   Objects: There must be at least one occurrence of the word "OCX"
' *       in your document.

' Specify find and replace strings for the procedure
' managing the global find and replace.
GlobalTextFindReplace "OCX", "ActiveX"
End Sub

Sub GlobalTextFindReplace(tmpFindString As String, tmpReplaceString As String)
' Use the value of the tmpFindString parameter to the
' FindString property.
.Application.FindAndReplace.FindString = tmpFindString
' Use the value of the tmpReplaceString parameter to the
' ReplaceString property.
.Application.FindAndReplace.ReplaceString = tmpReplaceString
' Search the entire document.
.Application.FindAndReplace.Where = $LwpLookWhereEntireDocument
' Include all text streams in the document.
.Application.FindAndReplace.IncludeList = $LwpIncludeListAllText
' Search for exact case matches.
.Application.FindAndReplace.FindExactCase = True
' Replace with an exact case.
.Application.FindAndReplace.ReplaceExactCase = True

.InitFindAndReplace True
' Find the first occurrence.
.Find
' Replace all occurrences.
.ReplaceAll
.InitFindAndReplace False
End Sub
```

Finding and replacing named styles

You can also Find and Replace named styles in your document using a variation of the above statements.

```
Sub GlobalStyleFindReplace
' * RUNTIME DEPENDENCIES
' *   Objects: There must be at least one occurrence of a paragraph in the
' *       named style "Bullet 1".

' (Optional) Go to the beginning of the document as
' a little insurance.
.Type "[ctrlHome]"
' Specify the name of the style to search for.
.Application.FindAndReplace.FindStyleName = "Bullet 1"
' Specify the name of the style to substitute.
.Application.FindAndReplace.ReplaceStyleName = "Bullet 2"
```

```

        .InitFindAndReplace True
        .ReplaceAll
    End Sub

```

Finding and replacing string elements

Find and Replace works against text and styles in your document. You may also need to replace elements of a longer string that you are working with in your scripts. You could write the string to your document, but it would be more effective to perform the Find and Replace within your working script.

```

Sub StringElementReplacement
    ' Declare a string variable for the ReplaceString function.
    Dim ReplaceIt As String
    ' Declare a string variable to hold the string containing
    ' an element to be replaced.
    Dim MyInput As String
    ' Specify the string containing an element to be replaced.
    MyInput = "onetwothreefourfive"
    ' Call the ReplaceTextString function to perform the find and replace.
    ReplaceIt = ReplaceTextString(MyInput, "two", "Three")
    ' Direct the results of the function call to the IDE output panel.
    Print ReplaceIt
    ' Output = "oneThreethreefourfive"
End Sub

```

```

Function ReplaceTextString (inputString As String, findString As String, replaceString
As String) As String
    ' Declare a temporary string variable.
    Dim tempString As String
    ' Declare two variables to manage find and replace boundaries
    ' within the complete input string.
    Dim findPos As Integer
    Dim startSearchAt As Integer

    ' Assign the temporary string variable to the value of the
    ' inputString parameter.
    tempString = inputString
    findPos = Instr(tempString, findString)
    ' Select findString within tempString.
    While findPos
        tempString = Left$(tempString, findPos - 1) _
            & replaceString _
            & Right$(inputString, Len(tempString) _
                - (findPos + Len(findString)) + 1)
        findPos = Instr(findPos + Len(replaceString), _
            tempString, _
            findString)
    Wend

    ' Replace the selected findString element with the value of the
    ' replaceString parameter. Return the entire modified string to
    ' to the calling procedure.
    ReplaceTextString = tempString
End Function

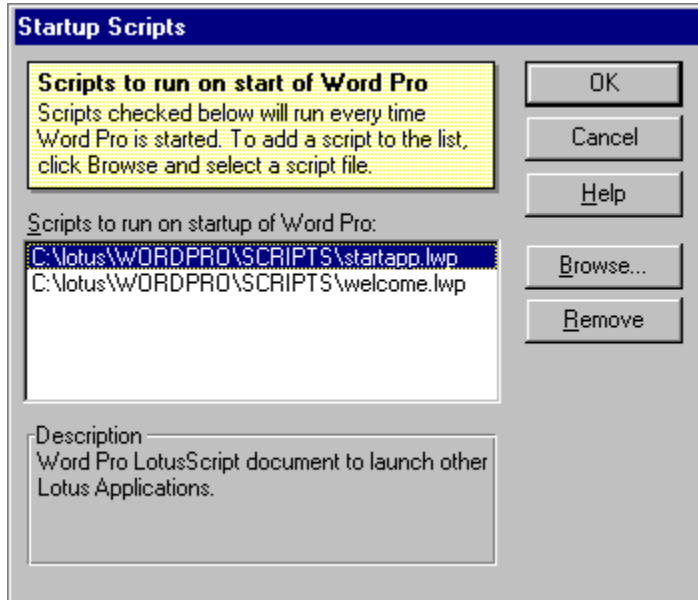
```

How do I run scripts when opening Word Pro or a document?

There are several ways to have scripts execute when you start a Word Pro session or open a document.

Running scripts when you load Word Pro

If you have scripts that you want to execute each time you start a Word Pro session, choose Edit - Scripts & Macros - Set Startup Scripts.



You can specify one or more Word Pro documents containing scripts, text files containing scripts, or LotusScript Object (LSO) files. Any global scripts contained in the documents you specify are executed whenever you start your Word Pro session.

Tip Another way to run a script each time you load Word Pro is to add the name of that script to the startup command that you use when you load Word Pro from Windows:

```
C:\LOTUS\WORDPRO.EXE C:\SCRIPTS\STARTUP.LWP!STARTUPSUB
```

Running scripts when you open a document

If you have procedures that you want to execute whenever you open a particular document, there are several possibilities to consider.

When you open a document, Word Pro executes procedures in the following scripts in the following order:

- !Globals Initialize
- !Document Initialize
- !Document Opened
- !WordPro Documentopened
- !Globals Main

Note The !Globals Initialize and !Document Initialize scripts also execute when you close your document.

Controlling display features at startup

If your script application does not need all the default display features offered in Word Pro, you can selectively turn them off when you load your document.

Place the following statements in any of the scripts mentioned above to simplify the number of features displayed when you run your script application.

```
' Startup settings to turn off palettes and tools.  
' =====  
' CleanScreen mode is useful in setting startup display preferences.  
' If you want to reduce the number of palettes and toolbars  
' displayed while you run your script application, you can turn  
' tools off one by one or you could set preferences for CleanScreen
```

```

' mode and then turn that on.
With .ApplicationWindow.UserInterfacePrefs.WinViewPrefs
    IsViewTitleBarCleanScrn = True
    IsViewMenuCleanScrn = True
    IsViewSmartIconsCleanScrn = False
    IsViewStatusBarCleanScrn = False
    IsVerticalScrollBarCleanScrn = False
    IsHorizontalScrollBarCleanScrn = False
    IsViewReturnIconCleanScrn = True
End With

' Use CleanScreen mode.
.ApplicationWindow.UserInterfacePrefs.CleanScreenMode = True
.ActiveDocWindow.WinViewPrefs.Refresh

' Startup settings to turn off text markers and symbols.
' =====
' Turn off outline tools (if currently displayed).
.ActiveDocWindow.WinViewPrefs.IsInOutline = False

' Turn off highlighting for misspelled words.
.ActiveDocWindow.WinViewPrefs.IsDisplayMisspelled = False

' Turn off displayed symbols for tabs, carriage returns, bookmarks, section
' breaks, and so on.
With .ActiveDocWindow.WinViewPrefs
    IsViewTabs = False
    IsViewReturns = False
    IsViewRulerMarks = False
    IsPageBreakMarks = False
    IsViewSectionBreakMarks = False
    IsViewColumnBreakMarks = False
    IsViewBookmarks = False
    ViewType = &H1
    IsHideHeaderFooter = False
End With

```

How do I style text and paragraphs?

You can style text and paragraphs in your documents in three ways:

- Change the style properties of text.
- Apply a named character style to some text.
- Change the paragraph style for a paragraph.

Changing the style properties of text

The following example illustrate how to change the font, point size, italic, and color of a selected sentence.

```
Sub LocalStyleChange
    ' Select the current sentence.
    .SelectSentence
    ' Apply a new font.
    .Text.Font.FontName = "Gill Sans"
    ' Apply a new point size.
    .Text.Font.Size = 9.00
    ' Make the text italic.
    .Text.Font.RevertToStyle $LwpFontPropertyItalic
    ' Change the text color to red.
    .Text.Font.FontColor.Red = 255
    .Text.Font.FontColor.Blue = 0
    .Text.Font.FontColor.Green = 0
    .Text.Font.FontColor.Override = $LwpColorOverrideRgb
End Sub
```

Applying a named character style

Named character styles in Word Pro let you apply a set of text properties to a piece of selected text without changing its paragraph style. The following example illustrates how to apply a character style named "My Character Style" to a selected sentence.

```
Sub ApplyCharacterStyle
' * RUNTIME DEPENDENCIES
' *   Objects: There must be a character style named "My New Char Style"
' *       in your document or attached SmartMaster.

    ' Select the current sentence.
    .SelectSentence
    ' Apply the named character style to the selected sentence.
    .Text.CharacterStyleName = "My New Char Style"
End Sub
```

Applying a named paragraph style

Named paragraph styles are powerful tools for managing the overall format of your document. Changing the paragraph style for one or more paragraphs in your document is similar to applying a name character style.

```
Sub ApplyNewParagraphStyle
' * RUNTIME DEPENDENCIES
' *   Objects: There must be a paragraph style named "Default Text"
' *       in your document or attached SmartMaster.

    ' Select the current paragraph as insurance.
    .SelectParagraph
    ' Apply the named paragraph style "Default Text" to the
    ' selected paragraph.
    .Text.ParagraphStyleName = "Default Text"
End Sub
```


How do I use timer functions?

It is sometimes necessary to pause execution of a script for a number of seconds. The following scripts illustrate how to use a generic timer sub to manage such pauses in your script application.

```
Sub CallTheTimer
    ' Calls the GenericTimer sub and specifies the
    ' a number of seconds that it should pause before
    ' continuing execution.
    GenericTimer 6
End Sub

Sub GenericTimer(duration As Integer)
    MessageBox("Begin a pause for " & duration & " seconds")

    stopWatch = Timer
    While ((Timer - stopWatch) < duration)
    Wend
    MessageBox(duration & " seconds is up")
End Sub
```

Word Pro: FilePassword property

{button ,AL('H_DOCCONTROL_CLASS','0)} [See list of classes](#)

{button ,AL('H_FILEPASSWORD_PROPERTY_EXSCRIPT','1)} [See example](#)

(Write-only) Allows a script to assign a password to a document.

Data Type

String

Syntax

```
[objectreference].FilePassword = filepasswordvalue
```

Legal values

Any value of type String.

Usage

You can use this property in conjunction with the FileProtectionType property to set a document password in LotusScript. For example, you could use the following lines of code to assign a document password:

```
.ActiveDocument.DocControl.FileProtectionType = $LwpFileProtectTypePassword  
.ActiveDocument.DocControl.FilePassword = "whatever"
```

Note

The document password will only take effect when the DocControlProtection property has the value \$LwpDocProtectTypeNone, or it has a value of \$LwpDocProtectTypeOnlyeditor and the DocControlRestrictedToEditor property matches the current User Name. The User Name can be accessed in the UserName property of the Preferences class.

The reason for this restriction is so that script cannot circumvent the security already in place on a document. None of the DocControl properties are accessible to LotusScript if these conditions are not met.

Word Pro: FileProtectionType property

{button ,AL(`H_DOCCONTROL_CLASS',0)} [See list of classes](#)

{button ,AL(`H_FILEPROTECTIONTYPE_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Restricts access to a specific file.

Data Type

Data type is [Variant](#), which allows the value of this property to be one of the constants listed below or its numeric equivalent (in parentheses).

Syntax

fileprotectiontypevalue = [objectreference].FileProtectionType

Legal values

<u>Value</u>	<u>Effect</u>
\$LwpFileProtectTypeEditors (261)	Only those listed as editors on the Editing Rights panel can open the document.
\$LwpFileProtectTypeNone (260)	Anyone with access to the folder where the document is stored can open the document.
\$LwpFileProtectTypeOrigAuthor (262)	Only the person who created the document can open it.
\$LwpFileProtectTypePassword (263)	Anyone who knows the document password can open the document.
\$LwpFileProtectTypeReserved (264)	Used internally by Word Pro when Word Pro does not recognize the file protection type being utilized by a document. For example, an existing file protection type is not recognized in a corrupted file or in a file created with a later version of Word Pro that added new verification types.

Usage

Equivalent to choosing File - TeamSecurity and selecting one of the options in the "Who can open (access) this file" section on the Access panel.

You can only change this property if the current editor has access rights to the TeamSecurity dialog box. If you do not have access to the dialog box, then changes to this property are ignored.

Word Pro: FilesToCompare property

{button ,AL(^H_REVIEWVERSIONS_CLASS',0)} [See list of classes](#)

(Write-only)

Data Type

String

Syntax

[objectreference].FilesToCompare = filestocomparevalue

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Word Pro: FileType property

{button ,AL('H_TEXTDOCUMENT_CLASS',0)} [See list of classes](#)

(Read-write)

Data Type

String

Syntax

filetypevalue = [objectreference].FileType

[objectreference].FileType = filetypevalue

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Word Pro: FillerPageText property

{button ,AL(^H_DIVISIONINFO_CLASS',0)} [See list of classes](#)

{button ,AL(^H_FILLERPAGETEXT_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-only)

Data Type

[Text](#)

Syntax

fillerpagetextvalue = [objectreference].FillerPageText

Legal values

Always contains an instance of the Text class.

Usage

Word Pro: FillStyle property

{button ,AL('H_TABLEFILL_CLASS',0)} [See list of classes](#)

(Read-write) Allows you to specify whether or not the background color and pattern fills the entire table, every other column, or every other row. You can specify more than one type of fill style.

Data Type

Data type is [Variant](#), which allows the value of this property to be one of the constants listed below or its numeric equivalent (in parentheses).

Syntax

fillstylevalue = [objectreference].FillStyle

[objectreference].FillStyle = fillstylevalue

Legal values

<u>Value</u>	<u>Effect</u>
\$LwpTableFillStyleAll (1870)	Sets a fill style for an entire table object.
\$LwpTableFillStyleEveryothercol (1872)	Sets a fill style for every other column in a table object.
\$LwpTableFillStyleEveryotherrow (1871)	Sets a fill style for every other row in a table object.
\$LwpTableFillStyleMixed (1873)	Sets more than one type of fill style for a table cell object.
\$LwpTableFillStyleNone (1874)	Indicates that no fill style is set for a table cell object.

Usage

You can use this property to return the current FillStyle value stored in the Background property of a specific table object. You can then use this value to specify whether or not the background color and pattern fills the entire table, every other column, or every other row. You can also specify more than one type of fill style. After you set the FillStyle property for a specific table object, it is stored in the Background property of that table object.

For more information, see [Background property](#).

Word Pro: FilterHelper property

{button ,AL(^H_FILTER_CLASS',0)} [See list of classes](#)

(Read-only)

Data Type

[FilterHelper](#)

Syntax

filterhelpervalue = [objectreference].FilterHelper

Legal values

Always contains an instance of the FilterHelper class.

Usage

Word Pro: FilterName property

{button ,AL('H_MERGEOPTIONS_CLASS',0)} [See list of classes](#)

(Read-write) The field names provided by an external data file.

Data Type

String

Syntax

filternamevalue = [objectreference].FilterName

[objectreference].FilterName = filternamevalue

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Word Pro: Filter property

{button ,AL('H_APPLICATIONWINDOW_CLASS',0)} [See list of classes](#)

(Read-only) An instance of the filter class which converts non-Word Pro file formats to a Word Pro file format.

Data Type

[Filter](#)

Syntax

filtervalue = [objectreference].Filter

Legal values

Always contains an instance of the Filter class.

Usage

Use this property to get to filter objects without having a document open.

Word Pro: FindAndReplace property

{button ,AL(^H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL(^H_FINDANDREPLACE_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-only) Contains the FindAndReplace object for the currently active session of Word Pro. The settings stored in this object are used by all documents opened within the active session.

Data Type

[FindAndReplace](#)

Syntax

findandreplacevalue = [objectreference].FindAndReplace

Legal values

Always contains an instance of the FindAndReplace class.

Usage

Accesses the FindAndReplace object, and checks and manipulates the settings for Find & Replace operations.

Word Pro: FindExactCase property

{button ,AL('H_FINDANDREPLACE_CLASS',0)} [See list of classes](#)

{button ,AL('H_FINDEACTCASE_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Enables you to find the exact case of a word or phrase in Find & Replace.

Data Type

[Integer](#)

Syntax

findexactcasevalue = [objectreference].FindExactCase

[objectreference].FindExactCase = findexactcasevalue

Legal values

Always contains an instance of the Text class. The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Checks and manipulates the setting for finding the exact case of a word or phrase in Find & Replace. If True, finds the word that matches the user setting. Equivalent to choosing Edit - Find & Replace Text, clicking Options, and selecting "Match case" in the "Find options" section.

Word Pro: FindFont property

{button ,AL(^H_FINDANDREPLACE_CLASS',0)} [See list of classes](#)

{button ,AL(^H_FINDFONT_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-only) Enables you to find a text font in Find & Replace.

Data Type

[Font](#)

Syntax

findfontvalue = [objectreference].FindFont

Legal values

Always contains an instance of the Font class.

Usage

Finds a font in Find & Replace. Equivalent to choosing Edit - Find & Replace Text, clicking Options, clicking the Font button in the "Find options" box, and selecting a font in the "Font name" list box on the Find panel.

Word Pro: FindForwardDirection property

{button ,AL(^H_FINDANDREPLACE_CLASS',0)} [See list of classes](#)

(Read-write) Enables you to set a forward or backward direction for a search in Find & Replace.

Data Type

Integer

Syntax

findforwarddirectionvalue = [objectreference].FindForwardDirection

[objectreference].FindForwardDirection = findforwarddirectionvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

You should set the forward or backward direction before using Find & Replace. If True, sets Find & Replace to go forward in the document. If False, sets Find & Replace to go backward in the document.

Word Pro: FindString property

{button ,AL('H_FINDANDREPLACE_CLASS',0)} [See list of classes](#)

{button ,AL('H_FINDSTRING_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Enables you to type what to search for in Find & Replace.

Data Type

String

Syntax

findstringvalue = [objectreference].FindString

[objectreference].FindString = findstringvalue

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Equivalent to choosing Edit - Find & Replace Text and typing a text or character string in the "Find" box.

Word Pro: FindStyleName property

{button ,AL('H_FINDANDREPLACE_CLASS',0)} [See list of classes](#)

(Read-write) Enables you to find a paragraph style in Find & Replace.

Data Type

String

Syntax

findstylevalue = [objectreference].FindStyleName

[objectreference].FindStyleName = findstylevalue

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Equivalent to choosing Edit - Find & Replace Text, typing the character "^p" in the "Find" box, clicking Options, clicking the Font button in the "Find options" section, and choosing a paragraph style in the "Style" list box.

Word Pro: FindWithProperties property

{button ,AL('H_FINDANDREPLACE_CLASS',0)} [See list of classes](#)

{button ,AL('H_FINDWITHPROPERTIES_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Enables you to find font properties in Find & Replace.

Data Type

[Integer](#)

Syntax

findwithpropertiesvalue = [objectreference].FindWithProperties

[objectreference].FindWithProperties = findwithpropertiesvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Use this property to find font properties in Find & Replace. If True, finds the font properties that match the user setting. Equivalent to choosing Edit - Find & Replace Text, clicking Options, and selecting "Include properties."

Word Pro: FinishedSpellChecking property

{button ,AL(^H_DIVISION_CLASS;H_TEXTDOCUMENT_CLASS',0)} [See list of classes](#)

(Read-only)

Data Type

Data type is Integer.

Syntax

finishedspellcheckingvalue = [objectreference].FinishedSpellChecking

Legal values

Usage

Word Pro: FirstChild property

{button ,AL(^H_DIVISION_CLASS;H_TEXTDOCUMENT_CLASS',0)} [See list of classes](#)

(Read-only)

Data Type

String

Syntax

firstchildvalue = [objectreference].FirstChild

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Word Pro: FirstCursorableDivision property

{button ,AL(^H_DIVISION_CLASS;H_TEXTDOCUMENT_CLASS',0)} [See list of classes](#)

(Read-only)

The name of the first division in the document into which the cursor will move. Divisions that are hidden, external, or parental are omitted.

Data Type

String

Syntax

firstcursorabledivisionvalue = [objectreference].FirstCursorableDivision

Legal values

This property is read-only. The value of this property cannot be set by a script.

Usage

Returns the internal name of the division.

This property is available from the TextDocument class and the Division class. If you call it from a Division object, it returns the current division's internal name.

Word Pro: FirstDivision property

{button ,AL(^H_DIVISION_CLASS;H_TEXTDOCUMENT_CLASS',0)} [See list of classes](#)

(Read-only)

Data Type

Data type is String.

Syntax

firstdivisionvalue = [objectreference].FirstDivision

Legal values**Usage**

Word Pro: FirstName property

{button ,AL('H_DIVISION_CLASS;H_TEXTDOCUMENT_CLASS',0)} [See list of classes](#)

{button ,AL('H_FIRSTNAME_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-only)

Data Type

String

Syntax

firstnamevalue = [objectreference].FirstName

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Word Pro: FirstPage property

{button ,AL('H_DIVISION_CLASS;H_TEXTDOCUMENT_CLASS',0)} [See list of classes](#)

{button ,AL('H_FIRSTPAGE_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-only)

Data Type

[Integer](#)

Syntax

firstpagevalue = [objectreference].FirstPage

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Word Pro: First property

{button ,AL('H_INDENT_CLASS;H_RELATIVEINDENT_CLASS;H_CLICKHERE_CLASS',0)} [See list of classes](#)

{button ,AL('H_FIRST_PROPERTY_EXSCRIPT',1)} [See example](#)

[Indent, RelativeIndent]

(Read-write)

[ClickHere]

(Read-only)

The name of the first ClickHere block in the division (uses Tab order).

Data Type

[Indent, RelativeIndent]

Long

[ClickHere]

String

Syntax

firstvalue = [objectreference].First

[objectreference].First = firstvalue

Legal values

[Indent, RelativeIndent]

Data type is Long but the unit of measurement used for this property is Twips.

[ClickHere]

The name of the Click Here block which is first in the tab order.

Usage

Word Pro: FitType property

{button ,AL('H_DOCWINDOW_CLASS',0)} [See list of classes](#)

(Read-write)

Data Type

Integer (Enumerated Bitmask).

FitType.

Syntax

fittypevalue = [objectreference].FitType

[objectreference].FitType = fittypevalue

Legal values

LwpFitHorz (&H2)

LwpFitHorzmargin (&H4)

LwpFitVertical (&H1)

Usage

Word Pro: FixAcronymns property

{button ,AL('H_FORMATCHECKPREF_CLASS',0)} [See list of classes](#)

{button ,AL('H_FIXACRONYMNS_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Allows you to improve the appearance of acronyms in Format Check.

Data Type

[Integer](#)

Syntax

fixacronymnsvalue = [objectreference].FixAcronymns

[objectreference].FixAcronymns = fixacronymnsvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Equivalent to selecting "Improve the appearance of acronyms" in the Format Check Options dialog box.

Word Pro: FixBullets property

{button ,AL('H_FORMATCHECKPREF_CLASS',0)} [See list of classes](#)

{button ,AL('H_FIXBULLETS_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Enables format check to correct the format of bulleted lists.

Data Type

[Integer](#)

Syntax

fixbulletsvalue = [objectreference].FixBullets

[objectreference].FixBullets = fixbulletsvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Equivalent to choosing Edit - Format Check, clicking options, and selecting "improve format of bulleted lists."

Word Pro: FontColor property

{button ,AL(^H_FONT_CLASS',0)} [See list of classes](#)

{button ,AL(^H_FONTCOLOR_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-only)

Data Type

[Color](#)

Syntax

fontcolorvalue = [objectreference].FontColor

Legal values

Always contains an instance of the Color class.

Usage

Word Pro: FontFaceSupport property

{button ,AL('H_HTMLOPTIONS_CLASS',0)} [See list of classes](#)

{button ,AL('H_FONTFACESUPPORT_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write)

This language element has not yet been defined. You can check for a newer Help file on the Lotus web site.

Data Type

Integer.

Syntax

fontfacesupportvalue = [objectreference].FontFaceSupport

[objectreference].FontFaceSupport = fontfacesupportvalue

Legal values

The legal values for this property are -1 and 0 but you may use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Equivalent to choosing File - Internet - HTML Export Assistant and, on the Content panel, selecting "Font face support."

Word Pro: FontMatching property

{button ,AL('H_PREFERENCES_CLASS',0)} [See list of classes](#)

{button ,AL('H_FONTMATCHING_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write)

Data Type

Data type is Integer.

Syntax

[objectreference].FontMatching = fontmatchingvalue

fontmatchingvalue = [objectreference].FontMatching

Legal values

Usage

Word Pro: FontMetrics property

{button ,AL(^H_FONT_CLASS',0)} [See list of classes](#)

(Read-only)

Data Type

[FontMetrics](#)

Syntax

fontmetricsvalue = [objectreference].FontMetrics

Legal values

Always contains an instance of the FontMetrics class.

Usage

Word Pro: FontName property

{button ,AL('H_FONT_CLASS;H_FONTMETRICS_CLASS;H_PREFERENCES_CLASS',0)} [See list of classes](#)

{button ,AL('H_FONTNAME_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write)

Data Type

String

Syntax

fontnamevalue = [objectreference].FontName

[objectreference].FontName = fontnamevalue

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Word Pro: FontStyleName property

{button ,AL('H_CHARACTERSTYLE_CLASS;H_PARAGRAPHSTYLE_CLASS',0)} [See list of classes](#)

(Read-write)

Data Type

String

Syntax

fontstylevalue = [objectreference].FontStyleName

[objectreference].FontStyleName = fontstylevalue

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Word Pro: FontUnitName property

{button ,AL(^H_USERINTERFACEPREFS_CLASS',0)} [See list of classes](#)

(Read-only) Stores the name of the unit of measurement used for displaying and setting font size.

Data Type

String

Syntax

fontunitnamevalue = [objectreference].FontUnitName

Legal values

Usage

To actually work with the unit of measurement for fonts, use the FontUnits property. Although the FontUnits property is independent of this property, its value corresponds to the value of this property. For example, if the value of FontUnits is \$LtsScaleModeCentimeter (1056964840), then the value of this property is Centimeters.

Word Pro: FontUnits property

{button ,AL(^H_USERINTERFACEPREFS_CLASS',0)} [See list of classes](#)

(Read-write) Sets or returns the unit of measurement used for displaying and setting font size.

Data Type

Data type is [Variant](#), which allows the value of this property to be one of the constants listed below or its numeric equivalent (in parentheses).

Syntax

fontunitsvalue = [objectreference].FontUnits

[objectreference].FontUnits = fontunitsvalue

Legal values

<u>Value</u>	<u>Effect</u>
\$LtsScaleModeCentimeter (1056964840)	The unit of measurement is Centimeter.
\$LtsScaleModeInch (1056964838)	The unit of measurement is Inch.
\$LtsScaleModePoint (1056964837)	The unit of measurement is Point.
\$LwpScaleModePica (1728)	The unit of measurement is Pica.

Usage

Default is Point. Changes made to this property are only valid during the current session of Word Pro.

This property only affects the units displayed in the InfoBox, not the Word Pro status bar.

The FontUnits property does not affect the values used in the Size property of the font object, which actually sizes the font. Finally, changes to this property do not affect the units seen in the Chart Infobox.

Word Pro: Font property

{button ,AL(^H_CHARACTERSTYLE_CLASS;H_GRAPHICOLEBJECT_CLASS;H_GRAPHIC_CLASS;H_OLEOBJECT_CLASS;H_CLICKHERE_CLASS;H_TEXTMARKER_CLASS;H_PARAGRAPHSTYLE_CLASS;H_TEXT_CLASSES',0)} [See list of classes](#)

{button ,AL(^H_FONT_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-only) Stores the font object for a number of classes.

Data Type

Font

Syntax

fontvalue = [objectreference].Font

Legal values

Always contains an instance of the Font class.

Usage

This property stores a font object for many classes.

For example, In the Text class this property gives you access to the font object for the currently selected text. In the ParagraphStyle class, this property gives you access to the font object for a specific paragraph style.

Once you have accessed the font object, you can access the size and color of the font, whether it is bold or underlined, and so on.

Word Pro: FooterStyles property

{button ,AL('H_FOUNDRY_CLASS',0)} [See list of classes](#)

(Read-only)

Data Type

FooterLayoutCollection

Syntax

footerstylesvalue = [objectreference].FooterStyles

Legal values**Usage**

Word Pro: Footers property

{button ,AL('H_FOUNDRY_CLASS',0)} [See list of classes](#)

(Read-only) An object created from the FooterLayoutCollection class. This object provides access to FooterLayout objects.

Data Type

[FooterLayoutCollection](#)

Syntax

footersvalue = [objectreference].Footers

Legal values

Always contains an instance of the FooterLayoutCollection class.

Usage

When accessed through the Foundry property on a Division object, the collection object in this property provides access to all the FooterLayout objects contained in that Division object.

When accessed through the AppFoundry property on the WPApplication object, this collection object provides access to all the FooterLayout objects contained in the Word Pro Clipboard.

When accessed through the TempFoundry property on the WPApplication object, this collection object provides access to all the FooterLayout objects placed in TempFoundry by WordPro or a script.

When accessed through the Foundry property on the WPApplication object, this collection object provides access to all the FooterLayout objects contained in the currently active Division object.

This property is not used in the Foundry property on the TextDocument object.

For more information about collection classes, see [Overview: Word Pro LotusScript Collection Classes](#).

Word Pro: Footer property

```
{button ,AL('H_LAYOUT_CLASS;H_CELLGROUPLAYOUT_CLASS;H_CELLLAYOUT_CLASS;H_CONNECTEDLAY  
OUT_CLASS;H_COLUMNGROUPLAYOUT_CLASS;H_FOOTERLAYOUT_CLASS;H_FRAMELAYOUT_CLASS;H  
_FRAMEGROUPLAYOUT_CLASS;H_DROP_CAPLAYOUT_CLASS;H_GROUPLAYOUT_CLASS;H_HEADERLAY  
OUT_CLASS;H_NOTELAYOUT_CLASS;H_PAGELAYOUT_CLASS;H_ROWGROUPLAYOUT_CLASS;H_ROWLA  
YOUT_CLASS;H_RUBYLAYOUT_CLASS;H_SUPERTABLEGROUPLAYOUT_CLASS;H_SUPERTABLELAYOUT  
_CLASS;H_TABLELAYOUT_CLASS;H_ENDNOTELAYOUT_CLASS;H_FOOTNOTELAYOUT_CLASS;H_TABLE  
HEADINGLAYOUT_CLASS;H_TOCSUPERTABLELAYOUT_CLASS',0)} See list of classes
```

```
{button ,AL('H_FOOTER_PROPERTY_EXSCRIPT',1)} See example
```

(Read-only) Returns a footer object in a layout object.

Data Type

[Layout](#)

Syntax

footervalue = [objectreference].Footer

Legal values

Always contains an instance of the Layout class.

Usage

Use this property to access the footer layout object of a specific layout. Not all layout objects have footer layout objects. You can check the UseFooter property of a layout object to see if there is a footer layout object currently available.

Word Pro: FootnoteContSep property

{button ,AL('H_FOOTNOTEOPTIONS_CLASS',0)} [See list of classes](#)

{button ,AL('H_FOOTNOTECONTSEP_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-only) Sets or returns the FootnoteContSep object for the specified object.

Data Type

[FootnoteContSep](#)

Syntax

footnotecontsepvalue = [objectreference].FootnoteContSep

Legal values

Always contains an instance of the FootnoteContSep class.

Usage

Word Pro: FootnoteLayouts property

{button ,AL('H_FOUNDRY_CLASS',0)} [See list of classes](#)

(Read-only) An object created from the FootnoteLayoutCollection class. This object provides access to FootnoteLayout objects.

Data Type

[FootnoteLayoutCollection](#)

Syntax

footnotelayoutsvalue = [objectreference].FootnoteLayouts

Legal values

Always contains an instance of the FootnoteLayoutCollection class.

Usage

When accessed through the Foundry property on a Division object, the collection object in this property provides access to all the FootnoteLayout objects contained in that Division object.

When accessed through the AppFoundry property on the WPAApplication object, this collection object provides access to all the FootnoteLayout objects contained in the Word Pro Clipboard.

When accessed through the TempFoundry property on the WPAApplication object, this collection object provides access to all the FootnoteLayout objects placed in TempFoundry by WordPro or a script.

When accessed through the Foundry property on the WPAApplication object, this collection object provides access to all the FootnoteLayout objects contained in the currently active Division object.

This property is not used in the Foundry property on the TextDocument object.

For more information about collection classes, see [Overview: Word Pro LotusScript Collection Classes](#).

Word Pro: FootnoteNumbering property

{button ,AL('H_FOOTNOTEOPTIONS_CLASS',0)} [See list of classes](#)

{button ,AL('H_FOOTNOTENUMBERING_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-only) Sets or returns the FootnoteNumbering object for the specified object.

Data Type

[FootnoteNumbering](#)

Syntax

footnotenumberingvalue = [objectreference].FootnoteNumbering

Legal values

Always contains an instance of the FootnoteNumbering class.

Usage

Word Pro: FootnoteOptions property

{button ,AL('H_DIVISION_CLASS;H_TEXTDOCUMENT_CLASS',0)} [See list of classes](#)

{button ,AL('H_FOOTNOTEOPTIONS_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-only)

Data Type

[FootnoteOptions](#)

Syntax

footnoteoptionsvalue = [objectreference].FootnoteOptions

Legal values

Always contains an instance of the FootnoteOptions class.

Usage

Word Pro: FootnoteSeparator property

{button ,AL('H_FOOTNOTEOPTIONS_CLASS',0)} [See list of classes](#)

{button ,AL('H_FOOTNOTESEPARATOR_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-only) Sets or returns the FootnoteSeparator object for the specified object.

Data Type

[FootnoteSeparator](#)

Syntax

footnoteseparatorvalue = [objectreference].FootnoteSeparator

Legal values

Always contains an instance of the FootnoteSeparator class.

Usage

Word Pro: Footnotes property

{button ,AL('H_FOUNDRY_CLASS',0)} [See list of classes](#)

{button ,AL('H_FOOTNOTES_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-only) An object created from the FootnoteCollection class. This object provides access to Footnote objects.

Data Type

[FootnoteCollection](#)

Syntax

footnotesvalue = [objectreference].Footnotes

Legal values

Always contains an instance of the FootnoteCollection class.

Usage

When accessed through the Foundry property on a Division object, the collection object in this property provides access to all the Footnote objects contained in that Division object.

When accessed through the AppFoundry property on the WPAApplication object, this collection object provides access to all the Footnote objects contained in the Word Pro Clipboard.

When accessed through the TempFoundry property on the WPAApplication object, this collection object provides access to all the Footnote objects placed in TempFoundry by WordPro or a script.

When accessed through the Foundry property on the WPAApplication object, this collection object provides access to all the Footnote objects contained in the currently active Division object.

This property is not used in the Foundry property on the TextDocument object.

For more information about collection classes, see Overview: [Word Pro LotusScript Collection Classes](#).

Word Pro: ForeColorIndex property

{button ,AL(^H_BACKGROUND_CLASS;H_FONT_CLASS;H_BORDER_CLASS',0)} [See list of classes](#)

(Read-write)

Data Type

Data type is Integer.

Syntax

[objectreference].ForeColorIndex = forecolorindexvalue

forecolorindexvalue = [objectreference].ForeColorIndex

Legal values**Usage**

Word Pro: FormatCheckPreferences property

{button ,AL(^H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

(Read-only) An instance of the FormatCheckPref class which represents the options for the Format Check tool.

Data Type

[FormatCheckPref](#)

Syntax

formatcheckpreferencesvalue = [objectreference].FormatCheckPreferences

Legal values

Always contains an instance of the FormatCheckPref class.

Usage

Equivalent to selecting options in the Format Check Options dialog box. To open this dialog box, choose Edit - Check Format - Options.

Word Pro: FormatType property

{button ,AL('H_NUMERICFORMAT_CLASS',0)} [See list of classes](#)

{button ,AL('H_FORMATTYPE_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Allows you to access the format of numeric values within table cells.

Data Type

Variant (Enumerated)

NumberFormat

Syntax

formattypevalue = [objectreference].FormatType

[objectreference].FormatType = formattypevalue

Legal values

\$LtsNumberFormatComma (1056964626)

\$LtsNumberFormatFixed (1056964623)

\$LtsNumberFormatGeneral (1056964622)

\$LtsNumberFormatPercent (1056964627)

\$LtsNumberFormatScientific (1056964624)

\$LwpNumberFormatArgentineanpeso (1532)

\$LwpNumberFormatAustraliandollar (1533)

\$LwpNumberFormatAustrianschilling (1534)

\$LwpNumberFormatBelgianfranc (1535)

\$LwpNumberFormatBraziliancruzeiro (1536)

\$LwpNumberFormatBritishpound (1537)

\$LwpNumberFormatCanadiandollar (1538)

\$LwpNumberFormatChineseyuan (1539)

\$LwpNumberFormatCzechkoruna (1540)

\$LwpNumberFormatDanishkrone (1541)

\$LwpNumberFormatDefault (1576)

\$LwpNumberFormatEcu (1542)

\$LwpNumberFormatFinnishmarkka (1543)

\$LwpNumberFormatFrenchfranc (1544)

\$LwpNumberFormatGermanmark (1545)

\$LwpNumberFormatGreekdrachma (1546)

\$LwpNumberFormatHongkongdollar (1547)

\$LwpNumberFormatHungarianforint (1548)

\$LwpNumberFormatIndianrupee (1549)

\$LwpNumberFormatIndonesianrupiah (1550)

\$LwpNumberFormatIrishpunt (1551)

\$LwpNumberFormatItalianlira (1552)

\$LwpNumberFormatJapaneseyen (1553)

\$LwpNumberFormatLabel (1582)

\$LwpNumberFormatLuxembourgfranc (1554)

\$LwpNumberFormatMalaysianringgit (1555)

\$LwpNumberFormatMexicanpeso (1556)

\$LwpNumberFormatNetherlandsguilder (1557)

\$LwpNumberFormatNewzealanddollar (1558)

\$LwpNumberFormatNone (1531)
\$LwpNumberFormatNorwegiankroner (1559)
\$LwpNumberFormatOthercurrency (1575)
\$LwpNumberFormatPolishzloty (1560)
\$LwpNumberFormatPortugueseescudo (1561)
\$LwpNumberFormatRomanianlei (1562)
\$LwpNumberFormatRussianruble (1563)
\$LwpNumberFormatSingaporedollar (1564)
\$LwpNumberFormatSlovakiankoruna (1565)
\$LwpNumberFormatSloveniantholar (1566)
\$LwpNumberFormatSouthafricanrand (1567)
\$LwpNumberFormatSouthkoreanwon (1568)
\$LwpNumberFormatSpanishpeseta (1569)
\$LwpNumberFormatSwedishkrona (1570)
\$LwpNumberFormatSwissfranc (1571)
\$LwpNumberFormatTaiwandollar (1572)
\$LwpNumberFormatThaibaht (1573)
\$LwpNumberFormatUsdollar (1574)
\$LwpNumberLtsNumberFormatComma (1579)
\$LwpNumberLtsNumberFormatFixed (1578)
\$LwpNumberLtsNumberFormatGeneral (1577)
\$LwpNumberLtsNumberFormatPercent (1580)
\$LwpNumberLtsNumberFormatScientific (1581)

Usage

Equivalent to the "Current format" setting on the Number Format panel of the InfoBox for cell layout objects.

Word Pro: Formula property

{button ,AL('H_CELLLAYOUT_CLASS;H_CONNECTEDLAYOUT_CLASS;H_POWERFIELD_CLASS',0)} [See list of classes](#)

{button ,AL('H_FORMULA_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Allows you to set or return the formula in a table cell.

Data Type

String

Syntax

formulavalue = [objectreference].Formula

[objectreference].Formula = formulavalue

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

If there is no formula in the specified cell, this property contains an empty string.

Word Pro: FrameCaptionOptions property

{button ,AL(^H_TEXTDOCUMENT_CLASS',0)} [See list of classes](#)

(Read-only)

This language element has not yet been defined. Lotus provides updated Help files on its web site.

Data Type

FrameCaptionOptions

Syntax

framecaptionoptionsvalue = [objectreference].FrameCaptionOptions

Legal values

Always contains an instance of the FrameCaptionOptions class.

Usage

Word Pro: FrameStyleName property

{button ,AL('H_PREFERENCES_CLASS',0)} [See list of classes](#)

(Read-write)

Data Type

String

Syntax

framestylevalue = [objectreference].FrameStyleName

[objectreference].FrameStyleName = framestylevalue

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Word Pro: FrameStyles property

{button ,AL('H_FOUNDRY_CLASS',0)} [See list of classes](#)

(Read-only) An object created from the FrameLayoutCollection class. This object provides access to FrameLayout objects which are used as frame styles.

Data Type

[FrameLayoutCollection](#)

Syntax

framestylesvalue = [objectreference].FrameStyles

Legal values

Always contains an instance of the FrameLayoutCollection class.

Usage

When accessed through the Foundry property on a Division object, the collection object in this property provides access to the FrameLayout objects contained in that Division object which are used as frame styles.

When accessed through the AppFoundry property on the WPAApplication object, this collection object provides access to the FrameLayout objects contained in the Word Pro Clipboard which are used as frame styles.

When accessed through the TempFoundry property on the WPAApplication object, this collection object provides access to the FrameLayout objects placed in TempFoundry by WordPro or a script.

When accessed through the Foundry property on the WPAApplication object, this collection object provides access to the FrameLayout objects contained in the currently active Division object which are used as frame styles.

This property is not used in the Foundry property on the TextDocument object.

For more information about collection classes, see Overview: [Word Pro LotusScript Collection Classes](#).

Word Pro: Frames property

{button ,AL('H_FOUNDRY_CLASS',0)} [See list of classes](#)

(Read-only) An object created from the FrameLayoutCollection class. This object provides access to FrameLayout objects including those used as frame styles.

Data Type

[FrameLayoutCollection](#)

Syntax

framesvalue = [objectreference].Frames

Legal values

Always contains an instance of the FrameLayoutCollection class.

Usage

When accessed through the Foundry property on a Division object, the collection object in this property provides access to all the FrameLayout objects contained in that Division object.

When accessed through the AppFoundry property on the WPAApplication object, this collection object provides access to all the FrameLayout objects contained in the Word Pro Clipboard.

When accessed through the TempFoundry property on the WPAApplication object, this collection object provides access to all the FrameLayout objects placed in TempFoundry by WordPro or a script.

When accessed through the Foundry property on the WPAApplication object, this collection object provides access to all the FrameLayout objects contained in the currently active Division object.

This property is not used in the Foundry property on the TextDocument object.

For more information about collection classes, see Overview: [Word Pro LotusScript Collection Classes](#).

Word Pro: Frame property

{button ,AL(^H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

(Read-only) An instance of the FrameContainer class. This is a current context property which only contains an object when the focus of Word Pro includes a frame. If there is no frame in the focus, this property is empty.

Data Type

[FrameContainer](#)

Syntax

framevalue = [objectreference].Frame

Legal values

An instance of the FrameContainer class.

Usage

When the focus includes a frame, this property contains the FrameContainer object which groups together the objects that comprise the frame which has the focus. You can use this property to access the layout or other objects related to that frame.

Word Pro: FreeMenus property

{button ,AL('H_APPLICATIONWINDOW_CLASS',0)} [See list of classes](#)

(Read-only) MenuItem(s) object(s) created by the script writer from the MenuItem class.

Data Type

[MenuItem](#)

Syntax

freemenuvalue = [objectreference].FreeMenus

Legal values

Always contains an instance of the MenuItem class.

Usage

Use this property as a holding area for menu items until you implement them in Word Pro.

Word Pro: FullName property

{button ,AL(^H_APPLICATION_CLASS;H_WPAPPLICATION_CLASS;H_DOCUMENT_CLASS;H_TEXTDOCUMENT_CLASS',0)} [See list of classes](#)

{button ,AL(^H_FULLNAME_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-only) The full name and path of the currently active Word Pro session or document. If the document has not been saved, the FullName property for that document is empty.

Data Type

String

Syntax

fullnamevalue = [objectreference].FullName

Legal values

The value of this property cannot be set by a script.

Usage

Use this property to retrieve the full name and path of the application or a document for OLE automation or any other use. Call this property from the WPAApplication object to get the full directory path and executable name of the application which is running the currently active session of Word Pro. Call this property from a TextDocument object to get the full directory path and name of that document.

Note OLE is not supported under OS/2.

Word Pro: GapBetweenPanes property

{button ,AL(^H_DOCWINDOW_CLASS',0)} [See list of classes](#)

(Read-write)

Data Type

Long

Syntax

gapbetweenpanesvalue = [objectreference].GapBetweenPanes

[objectreference].GapBetweenPanes = gapbetweenpanesvalue

Legal values

Data type is Long but the unit of measurement used for this property is Twips. There are 1440 Twips per inch.

Usage

Word Pro: GetAfidHelpInfo property

{button ,AL(^H_GRAPHICOLEBJECT_CLASS;H_GRAPHIC_CLASS;H_OLEOBJECT_CLASS',0)} [See list of classes](#)

(Read-only)

Data Type

[Integer](#)

Syntax

getafidhelpinfovalue = [objectreference].GetAfidHelpInfo

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Word Pro: GetButtonId property

{button ,AL(^H_STATUSBARBUTTON_CLASS',0)} [See list of classes](#)

(Read-only)

This language element has not yet been defined. You can check for a newer Help file on the Lotus web site.

Data Type

Long

Syntax

getbuttonidvalue = [objectreference].GetButtonId

Legal values

This property is read-only. The value of this property cannot be set by a script.

Usage

Word Pro: GetButtonType property

{button ,AL('H_STATUSBARBUTTON_CLASS',0)} [See list of classes](#)

(Read-only) Returns the type of status bar button (text, graphic, or popup). The button type can correspond to one or more of the parameters found in the GetStandardButtonId method (StatusBar class).

Data Type

Long

Syntax

getbuttontypevalue = [objectreference].GetButtonType

Legal values

The legal values for this property are determined by the following ButtonType parameters in the CreateNewButton method (StatusBar class):

<u>Value</u>	<u>Effect</u>
LwpButtonBehaviorClickable (&H8)	Allows the button to be left-clicked.
LwpButtonBehaviorCollapsible (&H10)	Allows the button to shrink or grow so that the status bar can fill up the window. Only one is allowed per status bar. Word Pro's collapsible button is the date/time button.
LwpButtonBehaviorContainer (&H20)	Allows the button to contain child buttons.
LwpButtonBehaviorLeftclick (&H8)	Allows the button to be left-clicked.
LwpButtonBehaviorPopup (&H4)	Allows the button to pop up a list of alternatives.
LwpButtonBehaviorThermometer (&H80000)	Allows the button to display a thermometer graphic with percentages.
LwpButtonCanBeDepressed (&H40000)	Allows the button to stay depressed.
LwpButtonContentsCenterAligned (&H80)	Allows the button contents to be center-aligned.
LwpButtonContentsGray (&H200)	Allows the button contents to be grayed.
LwpButtonContentsHilited (&H400)	Allows the button contents to be highlighted (red in Word Pro).
LwpButtonContentsLeftAligned (&H40)	Allows the button contents to be left-aligned.
LwpButtonContentsRightAligned (&H100)	Allows the button contents to be right-aligned.
LwpButtonHasAutorepeat (&H4000)	Allows the button to repeat a command.
LwpButtonHasUpdownCtrl (&H20000)	Allows the button to have up/down control.
LwpButtonNoTextFromHost (&H800)	Allows the button to keep its user-defined text without changing; in other words, the text on this button is never going to require text from a host.
LwpButtonReserved (&H8000)	
LwpButtonSpacer (&H10000)	A spacer status bar button.
LwpButtonSupportDbClick (&H2000)	Allows the button to respond to a double-click.
LwpButtonSupportRightClick (&H1000)	Allows the button to support a right mouse click.
LwpButtonTypeGraphics (&H2)	Allows the button to display a graphic.
LwpButtonTypeText (&H1)	Allows the button to display text.

Usage

Determines the properties for the selected button.

If you want to use the text constant values in your script, you need to include WPBITMSK.LSS. For more information on this file and bitmasks in general, see [Overview: Word Pro LotusScript Enumerated Values](#).

Word Pro: GetFilterExtension property

{button ,AL(^H_FILTERHELPER_CLASS',0)} [See list of classes](#)

(Read-only)

Data Type

String

Syntax

getfilterextensionvalue = [objectreference].GetFilterExtension

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Word Pro: GetFilterExtForDialogBox property

{button ,AL(^H_FILTERHELPER_CLASS',0)} [See list of classes](#)

(Read-only)

Data Type

String

Syntax

getfilterextfordialogboxvalue = [objectreference].GetFilterExtForDialogBox

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Word Pro: GetFilterId property

{button ,AL(^H_FILTERHELPER_CLASS',0)} [See list of classes](#)

(Read-only)

Data Type

Integer

Syntax

getfilteridvalue = [objectreference].GetFilterId

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Word Pro: GetFormatName property

{button ,AL('H_DDELINK_CLASS',0)} [See list of classes](#)

(Read-only) Specifies the format of the data (for example, CF_TEXT, CF_BITMAP, and so on).

Data Type

String

Syntax

getformatnamevalue = [objectreference].GetFormatName

Legal values**Usage**

Allows you to inquire about the format of the data.

Word Pro: GetHomeDirectory property

{button ,AL('H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL('H_GETHOMEDIRECTORY_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-only) Contains the name of the home directory for the operating system.

Data Type

[String](#)

Syntax

gethomedirectory = [objectreference].GetHomeDirectory

Legal values**Usage**

Word Pro: GetItemName property

{button ,AL(^H_DDELINK_CLASS',0)} [See list of classes](#)

(Read-write) Sets or returns the name of a DDE link to another application. For example, if you link to Lotus 1-2-3 and paste link a range into a Word Pro document, the name of the range is the ItemName. If you do not name the range, Word Pro names it for you.

Data Type

String

Syntax

getitemnamevalue = [objectreference].GetItemName

[objectreference].GetItemName = getitemnamevalue

Legal values

Usage

If you change this property and update the link, the linked data updates accordingly.

Word Pro: GetRedoWhatDesc property

{button ,AL('H_VERSIONMANAGER_CLASS',0)} [See list of classes](#)

(Read-only)

Data Type

String

Syntax

getredowhatdescvalue = [objectreference].GetRedoWhatDesc

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Word Pro: GetServerName property

{button ,AL(^H_DDELINK_CLASS',0)} [See list of classes](#)

(Read-only) The executable name of the DDE server (for example, "1-2-3.EXE").

Data Type

String

Syntax

getservernamevalue = [objectreference].GetServerName

Legal values

Usage

You can inquire about the server name if you paste link an object. If you create the DDE link object manually, you must provide the name of the server, the topic name (usually the file name), and the item name (for example, a range or a bookmark).

Word Pro: GetStatus property

{button ,AL('H_DDELINK_CLASS',0)} [See list of classes](#)

(Read-write) Determines whether a DDE link is active or inactive.

Data Type

Integer

Syntax

getstatusvalue = [objectreference].GetStatus

[objectreference].GetStatus = getstatusvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

DdeLinks can be active or inactive. If this property is set to True, you automatically get an update when changes are made to a DDE link. For example, if you paste link a range from Lotus 1-2-3 into a Word Pro document and make changes to the range in 1-2-3, the Word Pro document is automatically updated when you save.

If this property is set to False, you do not automatically get an update when changes are made to a DDE link.

Word Pro: GetTopicName property

{button ,AL('H_DDELINK_CLASS',0)} [See list of classes](#)

(Read-write) Sets or returns the topic name (usually the file name) of the DDE Link.

Data Type

String

Syntax

gettopicnamevalue = [objectreference].GetTopicName

[objectreference].GetTopicName = gettopicnamevalue

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Word Pro: GetUndoWhatDesc property

{button ,AL(^H_GRAPHICOLEBJECT_CLASS;H_GRAPHIC_CLASS;H_OLEOBJECT_CLASS;H_VERSIONMANAGER_CLASS',0)} [See list of classes](#)

(Read-only)

Data Type

String

Syntax

getundowhatdescvalue = [objectreference].GetUndoWhatDesc

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Word Pro: GetValue property

{button ,AL('H_CELLLAYOUT_CLASS;H_CONNECTEDLAYOUT_CLASS',0)} [See list of classes](#)

(Read-only) Returns a string representing the numeric content of a cell.

Data Type

String

Syntax

getvaluevalue = [objectreference].GetValue

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

The value returned is rounded to two decimal places. If a cell's content is not numeric, the GetValue property contains an empty string.

Word Pro: GIFAnimationOn property

{button ,AL('H_PREFERENCES_CLASS',0)} [See list of classes](#)

{button ,AL('H_GIFANIMATIONON_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write)

Enables animated GIF files to run while in a Word Pro document.

Data Type

[Integer](#)

Syntax

gifanimationonvalue = [objectreference].GIFAnimationOn

[objectreference].GIFAnimationOn = gifanimationonvalue

Legal values

The legal values for this property are -1 and 0 but you may use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Animated GIFs will run if the value is set to -1. They will not run if the value is 0.

Word Pro: GlossaryDataFileName property

{button ,AL(^H_USERINTERFACEPREFS_CLASS',0)} [See list of classes](#)

{button ,AL(^H_GLOSSARYDATAFILENAME_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Sets or returns the name of the default (first) Glossary file.

Data Type

[String](#)

Syntax

glossarydatafilenamevalue = [objectreference].GlossaryDataFileName

[objectreference].GlossaryDataFileName = glossarydatafilenamevalue

Legal values

Usage

Equivalent to the "Default glossary file(s)" field on the Default files panel of the Word Pro Preferences dialog box.

Word Pro: GlossaryDataFiles property

{button ,AL('H_USERINTERFACEPREFS_CLASS',0)} [See list of classes](#)

{button ,AL('H_GLOSSARYDATAFILES_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-only) Returns the names of all default Word Pro Glossary files.

Data Type

[StringCollection](#)

Syntax

glossarydatafilesvalue = [objectreference].GlossaryDataFiles

Legal values

Always contains an instance of the StringCollection class.

Usage

Equivalent to the "Default glossary file(s)" field on the Default files panel of the Word Pro Preferences dialog box. In Word Pro, the "Default glossary file(s)" field can contain multiple file names. This property returns a collection of String objects which contain the names of all default Word Pro glossary files.

Word Pro: GlossaryDataPaths property

{button ,AL('H_USERINTERFACEPREFS_CLASS',0)} [See list of classes](#)

{button ,AL('H_GLOSSARYDATAPATHS_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-only) Multiple paths (drive and directory) where Word Pro looks for the Glossary file.

Data Type

[StringCollection](#)

Syntax

glossarydatapathsvalue = [objectreference].GlossaryDataPaths

Legal values

Always contains an instance of the StringCollection class.

Usage

Equivalent to the "Glossaries" field on the Locations panel of the Word Pro Preferences dialog box. The "Glossaries" field can contain multiple datafile paths. You can use this property to read these multiple paths, including the primary (default) path stored in the GlossaryPath property.

Word Pro: GlossaryPath property

{button ,AL('H_USERINTERFACEPREFS_CLASS',0)} [See list of classes](#)

{button ,AL('H_GLOSSARYPATH_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-only) The default path (drive and directory) where Word Pro looks for the Glossary file.

Data Type

String

Syntax

glossarypathvalue = [objectreference].GlossaryPath

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Equivalent to the "Glossaries" field on the Locations panel of the Word Pro Preferences dialog box. In Word Pro, the "Glossaries" field can contain multiple paths. This property contains the first path listed in the "Glossaries" field.

{button ,AL('H_INSERTPATH_METHOD_MEMDEF',0)} [See related topics](#)

Word Pro: Glossary's property

{button ,AL('H_FOUNDRY_CLASS',0)} [See list of classes](#)

(Read-only) An object created from the GlossaryCollection class. This object provides access to Glossary objects.

Data Type

[GlossaryCollection](#)

Syntax

glossaryvalue = [objectreference].Glossarys

Legal values

Always contains an instance of the GlossaryCollection class.

Usage

When accessed through the Foundry property on a Division object, the collection object in this property provides access to all the Glossary objects contained in that Division object.

When accessed through the AppFoundry property on the WPApplication object, this collection object provides access to all the Glossary objects contained in the Word Pro Clipboard.

When accessed through the TempFoundry property on the WPApplication object, this collection object provides access to all the Glossary objects placed in TempFoundry by WordPro or a script.

When accessed through the Foundry property on the WPApplication object, this collection object provides access to all the Glossary objects contained in the currently active Division object.

This property is not used in the Foundry property on the TextDocument object.

For more information about collection classes, see Overview: [Word Pro LotusScript Collection Classes](#).

Word Pro: GrammarFormalityLevel property

{button ,AL('H_GRAMMAR_CLASS',0)} [See list of classes](#)

{button ,AL('H_GRAMMARFORMALITYLEVEL_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Indicates the formality level when Grammar Check reviews a document.

Data Type

[Integer](#)

Syntax

grammarformalitylevelvalue = [objectreference].GrammarFormalityLevel

[objectreference].GrammarFormalityLevel = grammarformalitylevelvalue

Legal values

The legal values for this property are 0 (Informal), 1 (Standard), and 2 (Formal). Default is 1.

Usage

Use this property when you want to set the formality level for proofing a document in Grammar Check. Equivalent to choosing Edit - Check Grammar, clicking Options, and selecting the formality level in the "Formality" field on the Rules panel. There are three formality levels: informal, standard and formal.

Word Pro: GrammarOptions property

{button ,AL('H_USERINTERFACEPREFS_CLASS',0)} [See list of classes](#)

{button ,AL('H_GRAMMAROPTIONS_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-only) Allows you to access the grammar check settings available in the Options dialog box on the Grammar Check bar.

Data Type

[Grammar](#)

Syntax

grammaroptionsvalue = [objectreference].GrammarOptions

Legal values

Always contains an instance of the Grammar class.

Usage

Currently, the Word Pro Grammar Check contains 128 options. These options include on or off for all Grammar Check rules, and the values for other Grammar Check options, such as split infinitives, consecutive nouns, maximum number of words, and so on.

Word Pro: GrammarProofLevel property

{button ,AL('H_GRAMMAR_CLASS',0)} [See list of classes](#)

{button ,AL('H_GRAMMARPROOFLEVEL_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Indicates the level of proofing when Grammar Check reviews a document.

Data Type

[Integer](#)

Syntax

grammarprooflevelvalue = [objectreference].GrammarProofLevel

[objectreference].GrammarProofLevel = grammarprooflevelvalue

Legal values

The legal values for this property are 0 (full proof) and 1 (quick proof). Default is 1.

Usage

Use this property when you are choosing the proofing level for a document in Grammar Check. Equivalent to choosing Edit - Check Grammar, clicking Options, and selecting the proofing level in the "Grammar Check level" field on the Rules panel. There are two proofing levels: full proof and quick proof.

Word Pro: GrammarSetName property

{button ,AL('H_OPTIONS_CLASS',0)} [See list of classes](#)

(Read-write)

Data Type

String

Syntax

grammarsetnamevalue = [objectreference].GrammarSetName

[objectreference].GrammarSetName = grammarsetnamevalue

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Word Pro: GraphicExports property

{button ,AL(^H_FILTER_CLASS',0)} [See list of classes](#)

(Read-only)

Data Type

[StringCollection](#)

Syntax

graphicexportsvalue = [objectreference].GraphicExports

Legal values

Always contains an instance of the StringCollection class.

Usage

Word Pro: GraphicImports property

{button ,AL('H_FILTER_CLASS',0)} [See list of classes](#)

{button ,AL('H_GRAPHICIMPORTS_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-only)

Data Type

[StringCollection](#)

Syntax

graphicimportsvalue = [objectreference].GraphicImports

Legal values

Always contains an instance of the StringCollection class.

Usage

Word Pro: GraphicOleObject property

{button ,AL(^H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

(Read-only) Contains the graphic object or Ole object which is uppermost in the focus.

Note OLE is not supported under OS/2.

Data Type

[GraphicOleObject](#)

Syntax

graphicoleobjectvalue = [objectreference].GraphicOleObject

Legal values

Always contains an instance of the GraphicOleObject class.

Usage

Use this property when you want to access a graphic or OLE object, regardless of whether the object is seen by LotusScript as a Graphic object or an OleObject object.

Word Pro: GraphicPaths property

{button ,AL('H_USERINTERFACEPREFS_CLASS',0)} [See list of classes](#)

{button ,AL('H_GRAPHICPATHS_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-only) Returns the names of all default Word Pro graphics paths.

Data Type

[StringCollection](#)

Syntax

graphicpathsvalue = [objectreference].GraphicPaths

Legal values

Always contains an instance of the StringCollection class.

Usage

Equivalent to the "Graphics" field on the Locations panel of the Word Pro Preferences dialog box. In Word Pro, the "Graphics" field can contain multiple file names. This property returns a collection of String objects which contain the names of all default Word Pro graphics paths.

Word Pro: GraphicPath property

{button ,AL('H_USERINTERFACEPREFS_CLASS',0)} [See list of classes](#)

{button ,AL('H_GRAPHICPATH_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Sets or returns the default path (drive and directory) for graphics.

Data Type

String

Syntax

graphicpathvalue = [objectreference].GraphicPath

[objectreference].GraphicPath = graphicpathvalue

Legal values

A valid path including drive and directory.

Usage

In Word Pro, the "Graphics" field on the Locations panel of the Word Pro Preferences dialog box can contain multiple paths. This property contains the the first path listed in the "Graphics" field.

Word Pro: Graphics property

{button ,AL('H_FOUNDRY_CLASS',0)} [See list of classes](#)

(Read-only) An object created from the GraphicCollection class. This object provides access to Graphic objects.

Data Type

[GraphicCollection](#)

Syntax

graphicsvalue = [objectreference].Graphics

Legal values

Always contains an instance of the GraphicCollection class.

Usage

When accessed through the Foundry property on a Division object, the collection object in this property provides access to all the Graphic objects contained in that Division object.

When accessed through the AppFoundry property on the WPAApplication object, this collection object provides access to all the Graphic objects contained in the Word Pro Clipboard.

When accessed through the TempFoundry property on the WPAApplication object, this collection object provides access to all the Graphic objects placed in TempFoundry by WordPro or a script.

When accessed through the Foundry property on the WPAApplication object, this collection object provides access to all the Graphic objects contained in the currently active Division object.

This property is not used in the Foundry property on the TextDocument object.

For more information about collection classes, see [Overview: Word Pro LotusScript Collection Classes](#).

Word Pro: GraphicType property

{button ,AL('H_HTMLOPTIONS_CLASS',0)} [See list of classes](#)

{button ,AL('H_GRAPHICTYPE_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) When exporting Word Pro files as HTML, graphics can be saved as JPEG or PNG files. This property specifies what format is currently chosen.

Data Type

Integer.

Syntax

graphicstypevalue = [objectreference].GraphicType

[objectreference].GraphicType = graphicstypevalue

Legal values

<u>Value</u>	<u>What it means</u>
0	Saves exported graphics as JPEG files.
2	Saves exported graphics as PNG files.

Note that 1
is not a
legal value.

Usage

This property only affects graphics which are exported to HTML.

Equivalent to choosing File - Internet - HTML Export Assistant, and, on the Preview & Save panel, specifying "JPEG" or "PNG" in the "Save graphics as" section. The Preview & Save panel allows you to specify options for exporting Word Pro files as HTML.

Word Pro: Graphic property

{button ,AL('H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

(Read-only) Contains the Graphic object which is uppermost in the focus.

Data Type

[Graphic](#)

Syntax

graphicvalue = [objectreference].Graphic

Legal values

Always contains an instance of the Graphic class.

Usage

Use this property when you want to access the Graphic object that currently has the focus. If you want to access a graphic that is an OLE object, use the OleObject property on WPAApplication. If you're not sure if a graphic is an OLE object, use the GraphicOleObject property which is capable of containing both Graphic objects and OleObject objects.

Note OLE is not supported under OS/2.

Word Pro: Green property

{button ,AL('H_COLOR_CLASS',0)} [See list of classes](#)

{button ,AL('H_GREEN_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) The green component of a color.

Data Type

Integer

Syntax

greenvalue = [objectreference].Green

[objectreference].Green = greenvalue

Legal values

The value of the Green property can range from 0 - 255.

Usage

Use the Green property to access the current level of green in a specific object's color. For example, if you want to change the green value of a frame's background color, you can use the following statement:

```
.Frame.Layout.Background.Color.Green = 128
```

Word Pro: Greeting property

{button ,AL('H_DOCCONTROL_CLASS',0)} [See list of classes](#)

{button ,AL('H_GREETING_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) The text that displays after the greeting option is set.

Data Type

[String](#)

Syntax

greetingvalue = [objectreference].Greeting

[objectreference].Greeting = greetingvalue

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Equivalent to choosing File - TeamSecurity and entering text in the "Display Greeting with this text" box on the Editing Rights panel. You must set the UseGreeting property which tells Word Pro to display a greeting box, before you can enter text to display in it.

Word Pro: GridDistance property

```
{button ,AL(`H_LAYOUT_CLASS;H_CELLGROUPLAYOUT_CLASS;H_CELLLAYOUT_CLASS;H_CONNECTEDLAY  
OUT_CLASS;H_COLUMNGROUPLAYOUT_CLASS;H_FOOTERLAYOUT_CLASS;H_FRAMELAYOUT_CLASS;H  
_FRAMEGROUPLAYOUT_CLASS;H_DROP_CAPLAYOUT_CLASS;H_GROUPLAYOUT_CLASS;H_HEADERLAY  
OUT_CLASS;H_NOTELAYOUT_CLASS;H_PAGELAYOUT_CLASS;H_ROWGROUPLAYOUT_CLASS;H_ROWLA  
YOUT_CLASS;H_RUBYLAYOUT_CLASS;H_SUPERTABLEGROUPLAYOUT_CLASS;H_SUPERTABLELAYOUT  
_CLASS;H_TABLELAYOUT_CLASS;H_ENDNOTELAYOUT_CLASS;H_FOOTNOTELAYOUT_CLASS;H_TABLE  
HEADINGLAYOUT_CLASS;H_TOCSUPERTABLELAYOUT_CLASS',0)} See list of classes
```

```
{button ,AL(`H_GRIDDISTANCE_PROPERTY_EXSCRIPT',1)} See example
```

(Read-write) Allows you to set the grid spacing for layout objects.

Data Type

Long

Syntax

griddistancevalue = [objectreference].GridDistance

[objectreference].GridDistance = griddistancevalue

Legal values

Data type is Long but the unit of measurement used for this property is Twips. There are 1440 Twips per inch.

Usage

Equivalent to the "Grid settings" option on the Misc panel of the InfoBox for certain layout objects.

```
{button ,AL(`H_GRIDTYPE_PROPERTY_MEMDEF;H_ISSNAPTO_PROPERTY_MEMDEF',0)} See related topics
```

Word Pro: GridType property

```
{button ,AL('H_LAYOUT_CLASS;H_CELLGROUPLAYOUT_CLASS;H_CELLLAYOUT_CLASS;H_CONNECTEDLAY  
OUT_CLASS;H_COLUMNGROUPLAYOUT_CLASS;H_FOOTERLAYOUT_CLASS;H_FRAMELAYOUT_CLASS;H  
_FRAMEGROUPLAYOUT_CLASS;H_DROP_CAPLAYOUT_CLASS;H_GROUPLAYOUT_CLASS;H_HEADERLAY  
OUT_CLASS;H_NOTELAYOUT_CLASS;H_PAGELAYOUT_CLASS;H_ROWGROUPLAYOUT_CLASS;H_ROWLA  
YOUT_CLASS;H_RUBYLAYOUT_CLASS;H_SUPERTABLEGROUPLAYOUT_CLASS;H_SUPERTABLELAYOUT  
_CLASS;H_TABLELAYOUT_CLASS;H_ENDNOTELAYOUT_CLASS;H_FOOTNOTELAYOUT_CLASS;H_TABLE  
HEADINGLAYOUT_CLASS;H_TOCSUPERTABLELAYOUT_CLASS',0)} See list of classes
```

```
{button ,AL('H_GRIDTYPE_PROPERTY_EXSCRIPT',1)} See example
```

(Read-write) Sets the type of grid that is displayed in a layout object.

Data Type

Data type is Variant, which allows the value of this property to be one of the constants listed below or its numeric equivalent (in parentheses).

Syntax

gridtypevalue = [objectreference].GridType

[objectreference].GridType = gridtypevalue

Legal values

<u>Value</u>	<u>Effect</u>
\$LwpGridTypeDots (2061)	Displays a specific type of dotted grid in the layout object.
\$LwpGridTypeLines (2060)	Displays a specific type of lined grid in the layout object.
\$LwpGridTypeNone (2059)	Prevents a grid from showing in the layout object.

Usage

Equivalent to the "Grid settings" option on the Misc panel of the InfoBox for certain layout objects.

```
{button ,AL('H_GRIDDISTANCE_PROPERTY_MEMDEF;H_ISSNAPTO_PROPERTY_MEMDEF',0)} See related topics
```

Word Pro: Groups property

{button ,AL('H_FOUNDRY_CLASS',0)} [See list of classes](#)

(Read-only) An object created from the GroupLayoutCollection class. This object provides access to GroupLayout objects.

Data Type

[GroupLayoutCollection](#)

Syntax

groupsvalue = [objectreference].Groups

Legal values

Always contains an instance of the GroupLayoutCollection class.

Usage

When accessed through the Foundry property on a Division object, the collection object in this property provides access to all the GroupLayout objects contained in that Division object.

When accessed through the AppFoundry property on the WPAApplication object, this collection object provides access to all the GroupLayout objects contained in the Word Pro Clipboard.

When accessed through the TempFoundry property on the WPAApplication object, this collection object provides access to all the GroupLayout objects placed in TempFoundry by WordPro or a script.

When accessed through the Foundry property on the WPAApplication object, this collection object provides access to all the GroupLayout objects contained in the currently active Division object.

This property is not used in the Foundry property on the TextDocument object.

For more information about collection classes, see [Overview: Word Pro LotusScript Collection Classes](#).

Word Pro: Gutter property

```
{button ,AL('H_LAYOUT_CLASS;H_CELLGROUPLAYOUT_CLASS;H_CELLLAYOUT_CLASS;H_CONNECTEDLAYOUT_CLASS;H_COLUMNGROUPLAYOUT_CLASS;H_FOOTERLAYOUT_CLASS;H_FRAMELAYOUT_CLASS;H_FRAMEGROUPLAYOUT_CLASS;H_DROP_CAP_LAYOUT_CLASS;H_GROUPLAYOUT_CLASS;H_HEADERLAYOUT_CLASS;H_NOTELAYOUT_CLASS;H_PAGELAYOUT_CLASS;H_ROWGROUPLAYOUT_CLASS;H_ROWLAYOUT_CLASS;H_RUBY_LAYOUT_CLASS;H_SUPERTABLEGROUPLAYOUT_CLASS;H_SUPERTABLELAYOUT_CLASS;H_TABLELAYOUT_CLASS;H_ENDNOTELAYOUT_CLASS;H_FOOTNOTELAYOUT_CLASS;H_TABLEHEADINGLAYOUT_CLASS;H_TOCSUPERTABLELAYOUT_CLASS',0)} See list of classes
```

```
{button ,AL('H_GUTTER_PROPERTY_EXSCRIPT',1)} See example
```

(Read-only) This property contains a Gutter object.

Data Type

Gutter

Syntax

guttervalue = [objectreference].Gutter

[objectreference].Gutter = guttervalue

Legal values

Always contains an instance of the Gutter class.

Usage

Word Pro: Hang property

{button ,AL('H_INDENT_CLASS;H_RELATIVEINDENT_CLASS',0)} [See list of classes](#)

(Read-write)

Data Type

Integer

Syntax

hangvalue = [objectreference].Hang

[objectreference].Hang = hangvalue

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Word Pro: HasContents property

{button ,AL(^H_DIVISIONINFO_CLASS',0)} [See list of classes](#)

(Read-write)

Data Type

Integer

Syntax

hascontentsvalue = [objectreference].HasContents

[objectreference].HasContents = hascontentsvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Word Pro: HasFocus property

{button ,AL(^H_DOCWINDOW_CLASS',0)} [See list of classes](#)

(Read-only)

Data Type

Integer

Syntax

hasfocusvalue = [objectreference].HasFocus

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Word Pro: HasIndex property

{button ,AL(^H_DIVISION_CLASS;H_TEXTDOCUMENT_CLASS',0)} [See list of classes](#)

(Read-only)

Data Type

Integer.

Syntax

hasindexvalue = [objectreference].HasIndex

Legal values**Usage**

Word Pro: HasLocalTabs property

{button ,AL(^H_CLICKHERE_CLASS;H_TEXTMARKER_CLASS;H_TEXT_CLASS',0)} [See list of classes](#)

(Read-only) Indicates whether or not the current paragraph has local tab settings, as opposed to tab settings found in the paragraph style.

Data Type

Variant (Enumerated)

Syntax

haslocaltabsvalue = [objectreference].HasLocalTabs

Legal values

\$LwpCommandResponseNo (148)

\$LwpCommandResponseYes (149)

Usage

Word Pro: HasTabs property

{button ,AL('H_PARAGRAPHSTYLE_CLASS',0)} [See list of classes](#)

{button ,AL('H_HASTABS_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-only)

Data Type

[Integer](#)

Syntax

hastabsvalue = [objectreference].HasTabs

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Word Pro: HasTOC property

{button ,AL(^H_DIVISION_CLASS;H_TEXTDOCUMENT_CLASS',0)} [See list of classes](#)

(Read-only)

Data Type

Integer.

Syntax

hastocvalue = [objectreference].HasTOC

Legal values**Usage**

Word Pro: HeaderStyles property

{button ,AL('H_FOUNDRY_CLASS',0)} [See list of classes](#)

(Read-only)

Data Type

HeaderLayoutCollection

Syntax

headerstylesvalue = [objectreference].HeaderStyles

Legal values**Usage**

Word Pro: Headers property

{button ,AL('H_FOUNDRY_CLASS',0)} [See list of classes](#)

(Read-only) An object created from the HeaderLayoutCollection class. This object provides access to HeaderLayout objects.

Data Type

[HeaderLayoutCollection](#)

Syntax

headersvalue = [objectreference].Headers

Legal values

Always contains an instance of the HeaderLayoutCollection class.

Usage

When accessed through the Foundry property on a Division object, the collection object in this property provides access to all the HeaderLayout objects contained in that Division object.

When accessed through the AppFoundry property on the WPAApplication object, this collection object provides access to all the HeaderLayout objects contained in the Word Pro Clipboard.

When accessed through the TempFoundry property on the WPAApplication object, this collection object provides access to all the HeaderLayout objects placed in TempFoundry by WordPro or a script.

When accessed through the Foundry property on the WPAApplication object, this collection object provides access to all the HeaderLayout objects contained in the currently active Division object.

This property is not used in the Foundry property on the TextDocument object.

For more information about collection classes, see [Overview: Word Pro LotusScript Collection Classes](#).

Word Pro: Header property

```
{button ,AL('H_LAYOUT_CLASS;H_CELLGROUPLAYOUT_CLASS;H_CELLLAYOUT_CLASS;H_CONNECTEDLAY  
OUT_CLASS;H_COLUMNGROUPLAYOUT_CLASS;H_FOOTERLAYOUT_CLASS;H_FRAMELAYOUT_CLASS;H  
_FRAMEGROUPLAYOUT_CLASS;H_DROP_CAPLAYOUT_CLASS;H_GROUPLAYOUT_CLASS;H_HEADERLAY  
OUT_CLASS;H_NOTELAYOUT_CLASS;H_PAGELAYOUT_CLASS;H_ROWGROUPLAYOUT_CLASS;H_ROWLA  
YOUT_CLASS;H_RUBY_LAYOUT_CLASS;H_SUPERTABLEGROUPLAYOUT_CLASS;H_SUPERTABLELAYOUT  
_CLASS;H_TABLELAYOUT_CLASS;H_ENDNOTELAYOUT_CLASS;H_FOOTNOTELAYOUT_CLASS;H_TABLE  
HEADINGLAYOUT_CLASS;H_TOCSUPERTABLELAYOUT_CLASS',0)} See list of classes
```

```
{button ,AL('H_HEADER_PROPERTY_EXSCRIPT',1)} See example
```

(Read-only) Returns a header object in a layout object.

Data Type

[Layout](#)

Syntax

headervalue = [objectreference].Header

Legal values

Always contains an instance of the Layout class.

Usage

Use this property to access the header layout object of the current layout. Not all layout objects have header layouts. You can check the UseHeader property of a layout object to see whether there is a header layout object currently available.

Word Pro: Heading property

```
{button ,AL('H_NUMBERING_CLASS;H_OUTLINESYSEQUENCE_CLASS;H_PARAGRAPHSTYLE_CLASS',0)}
```

[See list of classes](#)

```
{button ,AL('H_HEADING_PROPERTY_EXSCRIPT',1)} See example
```

(Read-write)

Data Type

[Integer](#)

Syntax

headingvalue = [objectreference].Heading

[objectreference].Heading = headingvalue

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Word Pro: Height property

```
{button ,AL('H_WINDOW_CLASS;H_APPLICATIONWINDOW_CLASS;H_STATUSBAR_CLASS;H_DOCWINDOW_CLASS;H_BASECONTAINER_CLASS;H_CELLCONTAINER_CLASS;H_FRAMECONTAINER_CLASS;H_NOTECONTAINER_CLASS;H_RUBYCONTAINER_CLASS;H_DROPCAPCONTAINER_CLASS;H_PAGECONTAINER_CLASS;H_SUBPAGECONTAINER_CLASS;H_SUPERPAGECONTAINER_CLASS;H_ROWCONTAINER_CLASS;H_SUPERTABLECONTAINER_CLASS;H_TABLECONTAINER_CLASS;H_PARALLELCOLSCONTAINER_CLASS;H_TABLEONLYCONT_CLASS;H_FONT_CLASS;H_GRAPHICOLEBJECT_CLASS;H_GRAPHIC_CLASS;H_OLEOBJECT_CLASS;H_ICONBAR_CLASS;H_LAYOUT_CLASS;H_CELLGROUPLAYOUT_CLASS;H_CELLLAYOUT_CLASS;H_CONNECTEDLAYOUT_CLASS;H_COLUMNGROUPLAYOUT_CLASS;H_FOOTERLAYOUT_CLASS;H_FRAMELAYOUT_CLASS;H_FRAMEGROUPLAYOUT_CLASS;H_DROPCAPLAYOUT_CLASS;H_GROUPLAYOUT_CLASS;H_HEADERLAYOUT_CLASS;H_NOTELAYOUT_CLASS;H_PAGELAYOUT_CLASS;H_ROWGROUPLAYOUT_CLASS;H_ROWLAYOUT_CLASS;H_RUBYLAYOUT_CLASS;H_SUPERTABLEGROUPLAYOUT_CLASS;H_SUPERTABLELAYOUT_CLASS;H_TABLELAYOUT_CLASS;H_ENDNOTELAYOUT_CLASS;H_FOOTNOTELAYOUT_CLASS;H_TABLEHEADINGLAYOUT_CLASS;H_TOCSUPERTABLELAYOUT_CLASS',0)} See list of classes
```

```
{button ,AL('H_HEIGHT_PROPERTY_EXSCRIPT',1)} See example
```

(Read-write) The distance in Twips from the top border of an object to the bottom border of an object; the actual returned height of the entire object.

[ApplicationWindow]

The height of the application window.

[IconBar]

(Read-only) The height of an icon bar object.

[StatusBar]

The height of the status bar window.

[Layout]

The height of a layout object.

[BaseContainer]

(Read-only) The height of a container object.

Data Type

[Font]

Single

[All others]

Long

Syntax

heightvalue = [objectreference].Height

[objectreference].Height = heightvalue

Legal values

[Font]

Represents the actual height in points; a single precision floating point value.

[All others]

Data type is Long but the unit of measurement used for this property is Twips. There are 1440 Twips per inch.

Usage

Used to set or return the height of an object, such as the status bar, an icon bar, or the application window.

Word Pro: HelpText property

{button ,AL('H_CLICKHERE_CLASS',0)} [See list of classes](#)

{button ,AL('H_HELPTEXT_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) The string that goes in the bubble help for a ClickHere block.

Data Type

String

Syntax

helpertextvalue = [objectreference].HelpText

[objectreference].HelpText = helpertextvalue

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Word Pro: HiddenMode property

{button ,AL('H_ATTRIBUTES_CLASS',0)} [See list of classes](#)

{button ,AL('H_HIDDENMODE_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Sets or returns the hidden attribute of text.

Data Type

[Integer](#)

Syntax

hiddenmodevalue = [objectreference].HiddenMode

[objectreference].HiddenMode = hiddenmodevalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

This property is equivalent to the "Hidden" setting in the "Attributes" list, located on the Text Font properties panel of the InfoBox.

Setting this property to True enables the Hidden attribute. Setting this property to False disables the Hidden attribute.

{button ,AL('H_SHOWHIDDENTEXT_PROPERTY_MEMDEF',0)} [See related topics](#)

Word Pro: HideFormula property

{button ,AL(^H_POWERFIELD_CLASS',0)} [See list of classes](#)

(Read-write)

Data Type

Integer

Syntax

hideformulavalue = [objectreference].HideFormula

[objectreference].HideFormula =hideformulavalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Word Pro: HideOutlineLevels property

{button ,AL('H_ATTRIBUTES_CLASS',0)} [See list of classes](#)

(Read-write)

Data Type

Integer

Syntax

hideoutlinelevelsvalue = [objectreference].HideOutlineLevels

[objectreference].HideOutlineLevels = hideoutlinelevelsvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Word Pro: HighLightMode property

{button ,AL('H_ATTRIBUTES_CLASS',0)} [See list of classes](#)

{button ,AL('H_HIGHLIGHTMODE_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Enables or disables the highlighted attribute.

Data Type

[Integer](#)

Syntax

highlightmodevalue = [objectreference].HighLightMode

[objectreference].HighLightMode = highlightmodevalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

The HighLightMode property of an attributes object serves different purposes depending on the class from which it is accessed.

For example, when accessed from the attributes object of the Text class, setting the HighLightMode property to True is equivalent to highlighting the text with the highlighter revision/comment tool.

Word Pro: HiLiteColor property

{button ,AL('H_EDITOR_CLASS;H_PREFERENCES_CLASS',0)} [See list of classes](#)

{button ,AL('H_HILITECOLOR_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-only) The color that is used to highlight text in a document.

Data Type

[Color](#)

Syntax

hilitecolorvalue = [objectreference].HiLiteColor

Legal values

Always contains an instance of the Color class.

Usage

[Editor]

Equivalent to the "Highlighter/comment color" option in the Markup Options dialog box. You can reach this dialog box by clicking Markup Options on the General panel of the Word Pro Preferences dialog box.

Word Pro: HorizontalSplitWindow property

{button ,AL('H_USERINTERFACEPREFS_CLASS',0)} [See list of classes](#)

{button ,AL('H_HORIZONTALSPLITWINDOW_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Sets or returns whether Word Pro splits a document area horizontally when the next document is opened or created.

Data Type

[Integer \(Bool\)](#)

Syntax

horizontalSplitWindowvalue = [objectreference].HorizontalSplitWindow

[objectreference].HorizontalSplitWindow = horizontalSplitWindowvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values. Default value is False (0).

Usage

Word Pro automatically sets this property to False after a document is opened or created in a split window.

Word Pro: HorizScrollBarVisible property

{button ,AL(^H_DOCWINDOW_CLASS',0)} [See list of classes](#)

(Read-only)

Data Type

Integer

Syntax

horizscrollbarvisiblevalue = [objectreference].HorizScrollBarVisible

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Word Pro: HorzRuler property

{button ,AL('H_APPLICATIONWINDOW_CLASS',0)} [See list of classes](#)

(Read-only) Contains a ruler object that indicates tab settings, indents, margins, and columns.

Data Type

[Ruler](#)

Syntax

horzrulervalue = [objectreference].HorzRuler

Legal values

Always contains an instance of the Ruler class.

Usage

Use this property to display the horizontal ruler when no document is open.

Word Pro: HTMLOptions property

{button ,AL('H_USERINTERFACEPREFS_CLASS',0)} [See list of classes](#)

{button ,AL('H_HTMLOPTIONS_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-only)

This language element has not yet been defined. Lotus provides updated Help files on its web site.

Data Type

HTMLOptions

Syntax

htmloptionsvalue = [objectreference].HTMLOptions

Legal values

Always contains an instance of the HTMLOptions class.

Usage

Word Pro: Hwnd property

{button ,AL(^H_APPLICATIONWINDOW_CLASS;H_DOCWINDOW_CLASS',0)} [See list of classes](#)

(Read) The Windows handle for the current application window.

Data Type

Long

Syntax

hwndvalue = [objectreference].Hwnd

Legal values**Usage**

Use this property to send script calls to the application window. For example, you can use this property to send .DLL files to the application window.

Word Pro: HyphenateLastWordInColumnOrPage property

{button ,AL(`H_HYPHENATIONOPTIONS_CLASS',0)} [See list of classes](#)

{button ,AL(`H_HYPHENATELASTWORDINCOLUMNORPAGE_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Determines whether or not Word Pro will hyphenate the last word in a column or on a page.

Data Type

[Integer](#)

Syntax

hyphenatelastwordincolumnorpagevalue = [objectreference].HyphenateLastWordInColumnOrPage

[objectreference].HyphenateLastWordInColumnOrPage = hyphenatelastwordincolumnorpagevalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

This property is equivalent to the "Hyphenate last word in:" "Column/page" box on the Options panel of the Document Properties dialog box.

{button ,AL(`H_NOHYPHENATE_PROPERTY_MEMDEF',0)} [See related topics](#)

Word Pro: HyphenateLastWordInPara property

{button ,AL(`H_HYPHENATIONOPTIONS_CLASS',0)} [See list of classes](#)

{button ,AL(`H_HYPHENATELASTWORDINPARA_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Determines whether Word Pro will hyphenate the last word in a paragraph.

Data Type

[Integer](#)

Syntax

HyphenateLastWordInParavalue = [objectreference].HyphenateLastWordInPara

[objectreference].HyphenateLastWordInPara = HyphenateLastWordInParavalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Equivalent to the "Hyphenate last word in: Paragraph" setting on the Options panel of the Document Properties dialog box.

{button ,AL(`H_NOHYPHENATE_PROPERTY_MEMDEF',0)} [See related topics](#)

Word Pro: HyphenationOptions property

{button ,AL('H_DIVISIONOPTIONS_CLASS',0)} [See list of classes](#)

{button ,AL('H_HYPHENATIONOPTIONS_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-only)

Data Type

[HyphenationOptions](#)

Syntax

hyphenationoptionsvalue = [objectreference].HyphenationOptions

Legal values

Always contains an instance of the HyphenationOptions class.

Usage

Word Pro: HyphZoneAfter property

{button ,AL(`H_HYPHENATIONOPTIONS_CLASS',0)} [See list of classes](#)

{button ,AL(`H_HYPHZONEAFTER_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Indicates the minimum number of characters that must appear after a hyphen.

Data Type

[Integer](#)

Syntax

hyphzoneaftervalue = [objectreference].HyphZoneAfter

[objectreference].HyphZoneAfter = hyphzoneaftervalue

Legal values

The legal values for this property are 2 - 9.

Usage

This property is equivalent to the "After hyphen" setting in the Document Properties dialog box.

{button ,AL(`H_NOHYPHENATE_PROPERTY_MEMDEF',0)} [See related topics](#)

Word Pro: HyphZoneBefore property

{button ,AL(`H_HYPHENATIONOPTIONS_CLASS',0)} [See list of classes](#)

{button ,AL(`H_HYPHZONEBEFORE_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Indicates the minimum number of characters that must appear before a hyphen.

Data Type

[Integer](#)

Syntax

HyphZoneBeforevalue = [objectreference].HyphZoneBefore

[objectreference].HyphZoneBefore = [objectreference].HyphZoneBefore

Legal values

The legal values for this property are 2 - 9.

Usage

This property is equivalent to the "Before hyphen" setting in the Document Properties dialog box.

{button ,AL(`H_NOHYPHENATE_PROPERTY_MEMDEF',0)} [See related topics](#)

Word Pro: IconBarManager property

{button ,AL('H_APPLICATIONWINDOW_CLASS',0)} [See list of classes](#)

(Read-only) The IconBarManager object for the current application window

Data Type

[IconBarManager](#)

Syntax

iconbarmanagervalue = [objectreference].IconBarManager

Legal values

Always contains an instance of the IconBarManager class.

Usage

You must go through this property to get to the IconBar for the application window. The application window can have multiple iconbars, but only one IconBarManager. You can use the IconBarManager to select, find, add, or remove icon bar objects.

Word Pro: IconBarPositionState property

{button ,AL('H_ICONBAR_CLASS',0)} [See list of classes](#)

{button ,AL('H_ICONBARPOSITIONSTATE_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-only) Indicates whether or not the icon bar object is fixed or floating on the workspace.

Data Type

Data type is [Variant](#), which allows the value of this property to be one of the constants listed below or its numeric equivalent (in parentheses).

Syntax

iconbarpositionstatevalue = [objectreference].IconBarPositionState

Legal values

<u>Value</u>	<u>Effect</u>
\$LwplIconBarPositionStateFixed (392)	The IconBar object is fixed at one side of the workspace (left, right, top, or bottom).
\$LwplIconBarPositionStateFloating (393)	The IconBar object is floating, not fixed to any side of the workspace.

Usage

Indicates whether or not the icon bar object is fixed or floating. A fixed icon bar object is attached on the left, right, top, or bottom of the workspace. A icon bar object is floating if it is not attached to an edge of the workspace.

Word Pro: IconBarSets property

{button ,AL('H_ICONBARMANAGER_CLASS',0)} [See list of classes](#)

{button ,AL('H_ICONBARSETS_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-only) Enumerates all icon bar sets by name, whether or not they are currently showing. The icon bar sets are listed by icon bar name, not by file name.

Data Type

[StringCollection](#)

Syntax

iconbarsetsvalue = [objectreference].IconBarSets

Legal values

Always contains an instance of the StringCollection class.

Usage

If you know the name of the icon bar set, you can select it through the IconBar class. You can then display it, hide it, add and remove icons from the set, and so on.

Word Pro: IconBars property

{button ,AL('H_ICONBARMANAGER_CLASS',0)} [See list of classes](#)

{button ,AL('H_ICONBARS_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-only) An object created from the IconBarCollection class. This object provides access to the collection of icon bars that are currently displayed in Word Pro. Usually, all icon bars currently showing in Word Pro are listed.

Data Type

[IconBarCollection](#)

Syntax

iconbarsvalue = [objectreference].IconBars

Legal values

Always contains an instance of the IconBarCollection class.

Usage

The collection of [IconBar](#) objects contained in the IconBars property are indexed by their icon bar name. For example, if the Universal icon bar is currently displayed in Word Pro, the IconBars property contains an IconBar object named "Universal."

In Word Pro, click the file button adjacent to an icon bar object to display a list of icon bars that can display in the current context. The icon bar names with check marks next to them are the bars currently showing.

If you want to access a bar that is not currently showing, specify the bar you want to manipulate by name. The names of all icon bars are available through the IconBarSets collection. Note that while you won't see the non-displayed bars when you iterate through this collection, you still can access them.

{button ,AL('H_ICONBARSETS_PROPERTY_MEMDEF',0)} [See related topics](#)

Word Pro: IconHelpText property

{button ,AL('H_ICONBARMANAGER_CLASS',0)} [See list of classes](#)

(Read-write) Allows you to inquire about or change the text that appears in the bubble help when the cursor hovers over an icon.

Data Type

String

Syntax

iconhelptextvalue = [objectreference].IconHelpText

[objectreference].IconHelpText = iconhelptextvalue

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

If you change this property for a specific icon, your changes display the next time the cursor hovers over the icon. You must first select the icon, using either the SelectStandardIcon or SelectCustomIcon method, for this property to function.

Word Pro: IconPaths property

{button ,AL('H_USERINTERFACEPREFS_CLASS',0)} [See list of classes](#)

(Read-only) Contains multiple paths (drive and directory) for SmartIcons.

Data Type

[StringCollection](#)

Syntax

iconpathsvalue = [objectreference].IconPaths

Legal values

Always contains an instance of the StringCollection class.

Usage

Equivalent to the "SmartIcons" field on the Locations panel of the Word Pro Preferences dialog box. The "SmartIcons" field can contain multiple document paths. This property returns a collection of String objects which contain the names of all default Word Pro icon paths.

Word Pro: IconPath property

{button ,AL(^H_USERINTERFACEPREFS_CLASS',0)} [See list of classes](#)

(Read-write) Stores the default path (drive and directory) for SmartIcons.

Data Type

String

Syntax

iconpathvalue = [objectreference].IconPath

[objectreference].IconPath = iconpathvalue

Legal values

A valid path including drive and directory.

Usage

In Word Pro, the "SmartIcons" field on the Locations panel of the Word Pro Preferences dialog box can contain multiple paths. This property contains the the first path listed in the "SmartIcons" field.

Word Pro: IconScript property

{button ,AL(^H_ICONBARMANAGER_CLASS',0)} [See list of classes](#)

(Read-write) Sets a script or macro for a custom icon.

Data Type

[StringCollection](#)

This field is not used for properties with CLASS data types.

Syntax

iconscript = [objectreference].IconScript

[objectreference].IconScript = iconscriptvalue

Legal values

String.

Usage

Before you can write a script for an icon, you must first select the icon using the SelectCustomIcon method.

Word Pro: IconSetName property

{button ,AL(^H_ICONBAR_CLASS',0)} [See list of classes](#)

(Read-only) Indicates the name of the icon bar object.

Data Type

String

Syntax

iconsetnamevalue = [objectreference].IconSetName

Legal values**Usage**

Equivalent to the "Bar name" field in the SmartIcons Setup dialog box.

Use this property to retrieve the name of an icon bar. To get access to any currently displayed icon bars, use the IconBars property of the IconBarManager object.

Word Pro: IconSize property

{button ,AL('H_ICONBARMANAGER_CLASS',0)} [See list of classes](#)

{button ,AL('H_ICONSIZE_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Indicates the size in which the icons are displayed; refers to all icons on every bar. You can set the size, either regular or large, in the "Icon size" field in the SmartIcons Setup dialog box.

Data Type

Variant (Enumerated)

IconSize

Syntax

iconsizevalue = [objectreference].IconSize

[objectreference].IconSize = iconsizevalue

Legal values

<u>Value</u>	<u>Effect</u>
\$LwplIconSizeSupervga (395)	Sets or changes the size of displayed icons to large.
\$LwplIconSizeVga (394)	Sets or changes the size of displayed icons to regular.

Usage

You can write a script to query or change the size of displayed icons.

Word Pro: IDispatch property

{button ,AL(^H_OLEOBJECT_CLASS',0)} [See list of classes](#)

(Read-only)

Note This property is not supported for the class OLEObject within OS/2.

Data Type

Variant.

Syntax

idispachvalue = [objectreference].IDispatch

Legal values**Usage**

Word Pro: ID property

{button ,AL('H_ICONBAR_CLASS;H_MENUITEM_CLASS',0)} [See list of classes](#)

(Read-only)

[IconBar]

The ID of the icon bar.

[MenuItem]

The menu ID for the menu item. ID is a read-only property which Word Pro sets each time you create your menu item.

Data Type

Long

Syntax

idvalue = [objectreference].ID

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Word Pro: IgnoreSoftHyphens property

{button ,AL('H_HYPHENATIONOPTIONS_CLASS',0)} [See list of classes](#)

{button ,AL('H_IGNORESOFTHYPHENS_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write)

Data Type

[Integer](#)

Syntax

ignoresofthyphensvalue = [objectreference].IgnoreSoftHyphens

[objectreference].IgnoreSoftHyphens = ignoresofthyphensvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Word Pro: IgnoreTab property

{button ,AL(^H_DIVISIONINFO_CLASS',0)} [See list of classes](#)

(Read-write)

Data Type

Integer

Syntax

ignoretabvalue = [objectreference].IgnoreTab

[objectreference].IgnoreTab = ignoretabvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Word Pro: ImportAsSource property

{button ,AL('H_HTMLOPTIONS_CLASS',0)} [See list of classes](#)

{button ,AL('H_IMPORTASSOURCE_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write)

This language element has not yet been defined. You can check for a newer Help file on the Lotus web site.

Data Type

Integer.

Syntax

importassourcevalue = [objectreference].ImportAsSource

[objectreference].ImportAsSource = importassourcevalue

Legal values

The legal values for this property are -1 and 0 but you may use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Equivalent to choosing File - Internet - HTML Import Options and selecting "Import as source code."

Word Pro: IncludeFooter property

{button ,AL('H_HTMLOPTIONS_CLASS',0)} [See list of classes](#)

{button ,AL('H_INCLUDEFOOTER_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write)

This language element has not yet been defined. You can check for a newer Help file on the Lotus web site.

Data Type

Integer.

Syntax

includefootervalue = [objectreference].IncludeFooter

[objectreference].IncludeFooter = includefootervalue

Legal values

The legal values for this property are -1 and 0 but you may use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Equivalent to choosing File - Internet - HTML Export Assistant and, on the Content panel, selecting "Include footer from document."

Word Pro: IncludeHeader property

{button ,AL('H_HTMLOPTIONS_CLASS',0)} [See list of classes](#)

{button ,AL('H_INCLUDEHEADER_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write)

This language element has not yet been defined. You can check for a newer Help file on the Lotus web site.

Data Type

Integer.

Syntax

includeheadervalue = [objectreference].IncludeHeader

[objectreference].IncludeHeader = includeheadervalue

Legal values

The legal values for this property are -1 and 0 but you may use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Equivalent to choosing File - Internet - HTML Export Assistant and, on the Content panel, selecting "Include header from document."

Word Pro: IncludeInitialsInNotes property

{button ,AL('H_OPTIONS_CLASS',0)} [See list of classes](#)

{button ,AL('H_INCLUDEINITIALSINNOTES_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Enables Word Pro to show the editor's initials in comments.

Data Type

[Integer](#)

Syntax

includeinitialsinnotesvalue = [objectreference].IncludeInitialsInNotes

[objectreference].IncludeInitialsInNotes = includeinitialsinnotesvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Equivalent to choosing File - Document Properties - Document, clicking the Options tab, and selecting the "Show editor initials in comments" checkbox.

Word Pro: IncludeList property

{button ,AL(^H_FINDANDREPLACE_CLASS',0)} [See list of classes](#)

(Read-write) Enables you to instruct Find & Replace to search for text in different areas in the document.

Data Type

Integer (Enumerated Bitmask)

Variant

Syntax

includelistvalue = [objectreference].IncludeList

[objectreference].IncludeList = includelistvalue

Legal values

<u>Value</u>	<u>Effect</u>
LwpIncludeListAllText (&H1)	Searches all text in the document.
LwpIncludeListFootnotes (&H20)	Searches all text in footnotes.
LwpIncludeListFrames (&H10)	Searches all text in frames.
LwpIncludeListHeadersFooters (&H4)	Searches all text in headers and footers.
LwpIncludeListMainDocText (&H2)	Searches text only in the main part of the document.
LwpIncludeListTables (&H8)	Searches all text in tables.

Usage

Equivalent to choosing Edit - Find & Replace Text, clicking Options, and selecting an option in the "Include" box in the "Find & replace scope" section.

Word Pro: IncludeURL property

{button ,AL('H_HTMLOPTIONS_CLASS',0)} [See list of classes](#)

{button ,AL('H_INCLUDEURL_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write)

This language element has not yet been defined. You can check for a newer Help file on the Lotus web site.

Data Type

Integer.

Syntax

includeurlvalue = [objectreference].IncludeURL

[objectreference].IncludeURL = includeurlvalue

Legal values

The legal values for this property are -1 and 0 but you may use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Equivalent to choosing File - Internet - HTML Export Assistant and, on the Layout panel, selecting "Include link to URL."

Word Pro: IndentFromLeft property

{button ,AL(^H_FOOTNOTESEPOPT_CLASS;H_FOOTNOTECONTSEP_CLASS;H_FOOTNOTESEPARATOR_CLASSES',0)} [See list of classes](#)

{button ,AL(^H_INDENTFROMLEFT_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Determines the distance that a footnote separator is indented from the left margin.

Data Type

Long

Syntax

indentfromleftvalue = [objectreference].IndentFromLeft

[objectreference].IndentFromLeft = indentfromleftvalue

Legal values

Data type is Long but the unit of measurement used for this property is Twips. There are 1440 Twips per inch.

Usage

Equivalent to the "Indent from left" box on the Separators panel of the Footnote and Endnote Options dialog box.

Word Pro: IndentStyleName property

{button ,AL('H_PARAGRAPHSTYLE_CLASS',0)} [See list of classes](#)

(Read-write)

Data Type

String

Syntax

indentstylevalue = [objectreference].IndentStyleName

[objectreference].IndentStyleName = indentstylevalue

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Word Pro: Indent property

{button ,AL(^H_CLICKHERE_CLASS;H_TEXTMARKER_CLASS;H_PARAGRAPHSTYLE_CLASS;H_TEXT_CLASS',
0)} [See list of classes](#)

{button ,AL(^H_INDENT_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-only)

Data Type

[Indent](#)

Syntax

indentvalue = [objectreference].Indent

Legal values

Always contains an instance of the Indent class.

Usage

Word Pro: IndexAlphabeticSeparator property

{button ,AL(^H_INDEXSECTION_CLASS',0)} [See list of classes](#)

(Read-write)

Data Type

Integer

Syntax

indexalphabeticseparatorvalue = [objectreference].IndexAlphabeticSeparator

[objectreference].IndexAlphabeticSeparator = indexalphabeticseparatorvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Word Pro: IndexDivision property

{button ,AL(^H_INDEXSECTION_CLASS',0)} [See list of classes](#)

(Read-write)

Data Type

String

Syntax

indexdivisionvalue = [objectreference].IndexDivision

[objectreference].IndexDivision = indexdivisionvalue

Legal values

Usage

Word Pro: IndexIndentType property

{button ,AL('H_INDEXSECTION_CLASS',0)} [See list of classes](#)

(Read-write)

Data Type

Integer

Syntax

indexindentypevalue = [objectreference].IndexIndentType

[objectreference].IndexIndentType = indexindentypevalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Word Pro: IndexParent property

{button ,AL('H_INDEXSECTION_CLASS',0)} [See list of classes](#)

(Read-write)

Data Type

String

Syntax

indexparentvalue = [objectreference].IndexParent

[objectreference].IndexParent = indexparentvalue

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

'Example: FindWithProperties property

'This example types two sentences, one of which is bold.

'A Find is then executed, specifying that bold text should be found.

```
.Type "No the time for all good men to come to the aid of their country.[Enter]"
```

```
.Bold
```

```
.Type "Now is the time for all good men to come to the aid of their country.[Enter]"
```

```
.Bold
```

```
.Application.FindAndReplace.FindFont.Clear
```

```
.Application.FindAndReplace.SearchAttributes.Clear
```

```
.Application.FindAndReplace.FindFont.Bold = True
```

```
.Application.FindAndReplace.FindWithProperties = True
```

```
.Application.FindAndReplace.UseFindStyle = 0
```

```
.InitFindAndReplace True
```

```
.Find
```

```
'Example: Find method
' This example inserts three identical sentences into the current document,
' clears the FindAndReplace settings, sets the FindString to "cat", and then displays
a message box.
' When you click OK, the script finds the first 'cat'. Clicking OK a second time
repeats the find.
' RUNTIME DEPENDENCIES: You must have a document open for this script to work.

Dim SentenceCount As Integer
For SentenceCount = 1 To 3
    .Text.InsertText "The sleep-deprived fox jumped over the gravity-challenged cat."
    .Text.SplitParagraph
Next
.Application.ResetFindAndReplace
.Application.FindAndReplace.FindString = "cat"
.InitFindAndReplace True
MessageBox "Click OK to find the first instance.",MB_OK,"Example Script"
.Find

MessageBox "Click OK to find the next instance.",MB_OK,"Example Script"
.Find
```

'Example: FirstName property

'This example inserts a new division into the document, and then displays

'the names of the first and last divisions in a message box.

```
.ApplicationWindow.SectionTabs.AddNewSectionTabs
```

```
FirstDiv = .ActiveDocument.Divisions(.ActiveDocument.FirstName).DivisionInfo.Name
```

```
LastDiv = .ActiveDocument.Divisions(.ActiveDocument.LastName).DivisionInfo.Name
```

```
MessageBox "The first division in this document is " & FirstDiv & _
```

```
    " and the last division is " & LastDiv & ".", MB_OK, "Example Script"
```

'Example: FirstPage property

'This example creates a new division in the current document, then displays

'a message box with the first page of the document, and the first

'page of the division.

.ApplicationWindow.SectionTabs.AddNewSectionTabs

DocFirst = .ActiveDocument.**FirstPage**

DivFirst = .Division.**FirstPage**

MessageBox "The first page of the document is " & DocFirst & " and the first page of
the division is " _

& DivFirst & " ", MB_OK, "Example Script"

```
'Example: First property
'This example indents the first line of the paragraph 1/2 inch.

.NewDocument
For i = 1 To 20
    .type "Indention test "
Next
.Text.Indent.First = 720 ' indent the first line 1/2 inch
MessageBox "Click OK to revert to the style indents.", MB_OK, "Example Script"
.Text.Indent.RevertToStyle
```

```
'Example: FixAcronymns property
'This example asks the user whether to improve the appearance of acronyms in
'Format Check, and then sets the appropriate option.

stat = MessageBox ("Do you want to improve the appearance of acronyms?", 36, "Example
Script")
If stat = 6 Then ' user said yes
    .Application.FormatCheckPreferences.FixAcronymns = True
Else
    .Application.FormatCheckPreferences.FixAcronymns = False
End If
```

'Example: FixBullets property

'This example asks the user whether to improve the appearance of bullets in

'Format Check, and then sets the appropriate option.

```
stat = MessageBox ("Do you want to improve the appearance of bullets?", 36, "Example  
Script")
```

```
If stat = 6 Then ' user said yes
```

```
    .Application.FormatCheckPreferences.FixBullets = True
```

```
Else
```

```
    .Application.FormatCheckPreferences.FixBullets = False
```

```
End If
```

'Example: FontColor property

'This example sets the font color to red, inserts some text, then

'changes the font color to black.

```
.Text.Font.FontColor.Red = 255
```

```
.Text.Font.FontColor.Blue = 0
```

```
.Text.Font.FontColor.Green = 0
```

```
.Text.Font.FontColor.Override = $LwpColorOverrideRgb
```

```
.Text.InsertText "This is red text. "
```

```
.Text.Font.FontColor.Red = 0
```

```
.Text.Font.FontColor.Blue = 0
```

```
.Text.Font.FontColor.Green = 0
```

```
.Text.Font.FontColor.Override = $LwpColorOverrideRgb
```



```
'Example: FontFaceSupport property
'This example asks the user if he wants to include specific font information
'when exporting HTML files, then sets the appropriate option.

stat = MessageBox ("Do you want to export font information to HTML?", 36, "Example
Script")
If stat = 6 Then ' user said yes
    .ApplicationWindow.UserInterfacePrefs.HTMLOptions.FontFaceSupport = True
Else
    .ApplicationWindow.UserInterfacePrefs.HTMLOptions.FontFaceSupport = False
End If
```

```
'Example: FontName property
Sub Main
Print "======"
Forall x In .division.foundry.paragraphstyles
    Print x.font.ActualName & " = " & x.font.size
End Forall
Forall x In .Division.foundry.paragraphstyles
    x.font.FontName = "Arial"
End Forall
Print " -----"
Forall x In .Division.foundry.paragraphstyles
    Print x.font.ActualName & " = " & x.font.size
End Forall
End Sub
```

```
'Example: Font property
Sub Main
Print "======"
Forall x In .division.foundry.paragraphstyles
    Print x.font.ActualName & " = " & x.font.size
End Forall
Forall x In .Division.foundry.paragraphstyles
    x.font.FontName = "Arial"
End Forall
Print " -----"
Forall x In .Division.foundry.paragraphstyles
    Print x.font.ActualName & " = " & x.font.size
End Forall
End Sub
```

'Example: Footer property

'This example changes the footer margins to two inches on each side.

.Page.Layout.RightPage.**Footer**.IsMarginSameAsParent = False

.Page.Layout.RightPage.**Footer**.MarginLeft = 2880 ' 2880 twips = 2 inches

.Page.Layout.RightPage.**Footer**.MarginRight = 2880

'Example: FootnoteContSep property

'This example sets options for the line separating continued footnotes.

'A fixed length line the width of the margins is set.

.Division.FootnoteOptions.**FootnoteContSep**.UseSeparatorLine = True

.Division.FootnoteOptions.**FootnoteContSep**.IsFixedLength = True

.Division.FootnoteOptions.**FootnoteContSep**.CustomLength = 0

.Division.FootnoteOptions.**FootnoteContSep**.IndentFromLeft = 0

.Division.FootnoteOptions.**FootnoteContSep**.SpaceAbove = 0

.Division.FootnoteOptions.**FootnoteContSep**.SpaceBelow = 80

'Example: FootnoteNumbering property

'This example sets numbering options for footnotes in the current division.

```
.Division.FootnoteOptions.FootnoteNumbering.StartingNumber = 1  
.Division.FootnoteOptions.FootnoteNumbering.LeadingText = "("  
.Division.FootnoteOptions.FootnoteNumbering.TrailingText = ")"  
.Division.FootnoteOptions.FootnoteNumbering.UseSuperscriptReferenceNum = False  
.Division.FootnoteOptions.FootnoteNumbering.ResetWhen = $LwpResetOptionEachPage
```

'Example: FootnoteOptions property

'This example sets reset numbering options for footnotes and endnotes.

.Division.**FootnoteOptions**.FootnoteNumbering.ResetWhen = \$LwpResetOptionEachPage

.Division.**FootnoteOptions**.EndnoteDivisionNum.ResetWhen = \$LwpResetOptionEachDivision

.Division.**FootnoteOptions**.EndnoteDivisionGroupNum.ResetWhen =
\$LwpResetOptionEachDivisiongroup

.Division.**FootnoteOptions**.EndnoteDocNum.ResetWhen = \$LwpResetOptionEachDoc

'Example: FootnoteSeparator property

'This example sets options for the separator line between the main text

'and footnotes. A custom length line of four inches is set.

.Division.FootnoteOptions.**FootnoteSeparator**.UseSeparatorLine = True

.Division.FootnoteOptions.**FootnoteSeparator**.IsFixedLength = False

.Division.FootnoteOptions.**FootnoteSeparator**.CustomLength = 5760 ' 5760 twips = 4
inches

.Division.FootnoteOptions.**FootnoteSeparator**.IndentFromLeft = 0

.Division.FootnoteOptions.**FootnoteSeparator**.SpaceAbove = 0

.Division.FootnoteOptions.**FootnoteSeparator**.SpaceBelow = 80

'Example: Footnotes property
'This example inserts several footnotes in the current document.
'The footnote collection is then iterated to find the first footnote,
'and the contents of the first footnote is modified.

```
.InsertFootnote $LwpFnTypeAtBottomOfPage  
.Type "This is footnote number 1.[Esc] This is some text."  
.InsertFootnote $LwpFnTypeAtBottomOfPage  
.Type "This is footnote number 2.[Esc]"  
Forall Footnote In .Division.Foundry.Footnotes  
    If Footnote.Number = 1 Then  
        Footnote.GoTo  
            .Text.InsertText "New text for footnote number one. "  
    End If  
End Forall
```

```
'Example: FormatType property
'This example creates a table, and types a number in the first cell.
'It then changes the numeric format of the cell to percentage.

.CreateTable
.Type "123.45"
.CurrentCell.NumericFormat.FormatType = $LtsNumberFormatPercent
```

'Example: Formula property

'This example creates a table. It inserts numbers in the first two rows of
'the first column, and a formula to add them in the third row.

```
.CreateTable False, "Default Table", 2, 3
```

```
.Type "1[Down]2[Down]"
```

```
.Table.CurrentCell.Formula = "A1+A2"
```

```
'Example: Forward method
' This example inserts a sentence of text into the current document. Each word
' starting with the first is then selected.
' RUNTIME DEPENDENCIES: You must have a document open for this script to work.

.Type "This is a Sentence of text."
.Text.MoveToStart $LwpLocationTypeLine

Do
    .SelectWord
    NextWord = .Text.Forward ($LwpNavigateObjectTypeWord, 1)
Loop Until (.Text.AtBeginningOfParagraph = True) Or (NextWord = False)
```

'Example: FrameRevert method

' This example inserts a frame into the current document and changes the
' frame's background color. After the message box is closed the frame is
' reverted to the default frame style attributes.
' RUNTIME DEPENDENCIES: You must have a document open for this script to work.

```
.NewFrame 3285, 1200, 1575, 1830  
.Frame.Layout.Background.Color.Red = 82  
.Frame.Layout.Background.Color.Blue = 239  
.Frame.Layout.Background.Color.Green = 145  
.Frame.Layout.Background.Color.Override = $LwpColorOverrideRgb  
.Frame.Anchor $LwpAnchorWhereLayout, $LwpConditionTypeOnlyspecificpage,  
$LwpRelativeTypeLytParent  
MessageBox "Click OK to revert frame to default attributes in frame style.  
",MB_OK,"Example Script"
```

.FrameRevert

```
'Example: FullName property
With WordPro.ActiveDocument
MsgTxt = "Current Word Pro Doc is " & .FullName
MsgTxt = MsgTxt & ", the author is " & .Docinfo.AuthorName
Msgbox MsgTxt,64,"Word Pro Information"
MsgTxt = "It was created on " & .DocInfo.CreationDateString & " at "
& .DocInfo.CreationTimeString
Msgbox MsgTxt,64,"Word Pro Information"
End With
```

```
'Example: GetCopyFormatCategories method
' This example retrieves the copy format categories and tests whether the
'text or table category is available.
' RUNTIME DEPENDENCIES: You must have a document open for this script to work.
```

```
Dim FormatCategories as Integer
FormatCategories = .Text.GetCopyFormatCategories
If (FormatCategories And &H1) then
    MsgBox "Text format available.",MB_OK,"Example Script"
End If
If (FormatCategories And &H2) then
    MsgBox "Table format available.",MB_OK,"Example Script"
End If
```

```
'Example: GetCount method
' This example inserts some text into the current document, and uses the
'GetCount method to perform a character count and a word count.

.Text.InsertText "This is an example of the GetCount method"
CharCount = .Content.GetCount ($LwpGetCountWhatText, $LwpGetCountWhichCharacter)
WordCount = .Content.GetCount ($LwpGetCountWhatText, $LwpGetCountWhichWord)
MessageBox "There are " & CharCount & " characters and " & WordCount & " words in the
text.", MB_OK, "Example Script"
```



```
'Example: GetCurrentMarkerName method
' This example displays the name of any clickhere or bookmark located
' at the cursor position. If the cursor position is not located on a
' clickhere or bookmark then no message is displayed.
' RUNTIME DEPENDENCIES: You must have a document open and the cursor
' located on a clickhere or bookmark for this script to work.
```

```
Dim MarkerName as string
```

```
MarkerName = .Text.GetCurrentMarkerName($LwpMarkerTypeClickhere)
```

```
If MarkerName <> "" then
    MessageBox "Clickhere found: " & MarkerName
Else
    MarkerName = .Text.GetCurrentMarkerName($LwpMarkerTypeBookmark)
    If MarkerName <> "" then
        Forall Marks In .Division.BookMarkManager.BookMarks
            If MarkerName = Marks.MarkerName Then
                MessageBox "Bookmark found: " & Marks.Name
            End If
        End Forall
    End If
End If
```

'Example: GetData method

' This example creates a dataset named 'ExampleDataSet' off of the active
' document. The 'FirstName' and 'LastName' items are created and filled with
' data. Finally the values for the dataset items are printed to the Script
' Editor Output panel. Since no dataset item named 'Address' was defined, the
' default dataset value will be printed in the last statement.
' RUNTIME DEPENDENCIES: You must have a document open for this script to work.

```
Dim DataSetName as String
```

```
Dim Defaultvalue as String
```

```
Dim DataSet As WPDataSetCollection
```

```
Set DataSet = .ActiveDocument.WPDataSets
```

```
DataSetName = "ExampleDataSet"
```

```
Defaultvalue = "Default"
```

```
DataSet(DataSetName).SetData "FirstName","John"
```

```
DataSet(DataSetName).SetData "LastName","Doe"
```

```
Print DataSet(DataSetName).GetData("FirstName",Defaultvalue)
```

```
Print DataSet(DataSetName).GetData("LastName",Defaultvalue)
```

```
Print DataSet(DataSetName).GetData("Address",Defaultvalue)
```

```
'Example: GetEnum method
' This example prints the numeric equivalent for the $LwpMergeActionNewfile
' constant to the Lotus Script Output panel.
' RUNTIME DEPENDENCIES: You must have a document open for this script to work.

Dim Value As Long
Value = .GetEnum("$LwpMergeActionNewfile")
Print Value
```

'Example: GetEnvelopeDefaults method

'This example determines the paper feed method to use for printing envelopes,

'and displays the method in a message box.

```
FeedType = .ActiveDocument.PrintManager.GetEnvelopeDefaults()
```

```
MessageBox "The current printer uses " & FeedType & " as the envelope feeder.", MB_OK,  
"Example Script"
```

```
'Example: GetFormula method
' This example creates a table and enters a formula. The formula is then
' retrieved and printed to the Lotus Script Output panel.
' RUNTIME DEPENDENCIES: You must have a document open for this script to work.

.CreateTable False, "Default Table", 5,5
.Table.CellLayout(4,0).GotoLayout
.Table.CellEngine.SetFormula 4,0,"@SUM(A1:A2)"
Print .Table.CellEngine.GetFormula(4, 0)
```

'Example: GetHomeDirectory property

'This example displays the Windows directory in a message box.

```
Messagebox "Windows is installed in " & .GetHomeDirectory & ".", MB_OK, "Example  
Script"
```

'Example: GetLastUsedFilter method

' This example returns the last text filter type used in Word Pro.

' RUNTIME DEPENDENCIES: You must have a document open for this script to work.

Print .ApplicationWindow.Filter.GetLastUsedFilter(\$LwpFilterTypeText)

```
'Example: GetLineStyle method
' This example creates a table, changes the line style and then prints
' the line style to the Lotus Script Output panel.
' RUNTIME DEPENDENCIES: You must have a document open for this script to work.

.CreateTable False, "Default Table", 5,5
.SelectEntireTable
.Table.TableLine.ChgLineStyle $LwpTableLineStyleAll
Print .Table.TableLine.GetLineStyle()
```



```
'Example: GetMarkedText method
' This example creates a temporary text marker for the selected text and then prints
the text to the LotusScript output panel.
' RUNTIME DEPENDENCIES: You must have a document open with selected text
' for this script to work.
```

```
Dim MarkerName As String
Dim NewMarker As TextMarker
Dim MarkerText As String
MarkerName = .Mark($LWPMarkerTypeDefault)
Set NewMarker = .Division.Foundry.TextMarkers.Item(MarkerName)
MarkerText = NewMarker.GetMarkedText
If MarkerText = "" then MsgBox "Select some text, then try this script again!", MB_OK,
"Example Script"
Print MarkerText
```

```
'Example: GetMarkerName method
' This example displays the name of any clickhere or bookmark located
' at the cursor position. If the cursor position is not located on a
' clickhere or bookmark then no message is displayed.
' RUNTIME DEPENDENCIES: You must have a document open and the cursor
' located on a clickhere or bookmark for this script to work.

Dim MarkerName as string
MarkerName = .Text.GetCurrentMarkerName($LwpMarkerTypeClickhere)
If MarkerName <> "" then
    MsgBox "Clickhere found: " & MarkerName
Else
    MarkerName = .Text.GetCurrentMarkerName($LwpMarkerTypeBookmark)
    If MarkerName <> "" then
        Forall Marks In .Division.BookMarkManager.BookMarks
            If MarkerName = Marks.MarkerName Then
                MsgBox "Bookmark found: " & Marks.Name
            End If
        End Forall
    End If ' got a bookmark
End If ' got something
```

```

'Example: GetMisspelledWord method
' This example sequentially highlights all misspelled words in the current
' document.
' RUNTIME DEPENDENCIES: You must have a document open for this script to work.

'move to the start of the document
.Text.MoveToStart $LwpLocationTypeDocument

Do
  ' get the current insertion point position on the page
  XPos1 = .Text.PositionXOnPage
  YPos1 = .Text.PositionYOnPage

  ' highlights the nearest misspelled word relative to insertion point
  Word = .Text.GetMisspelledWord
  If (Word <> "") Then
    MsgBox "This word is Misspelled", MB_OK, "Example Script"
    .Deselect
    .Type "[Right]"
  End If

  ' check position again to see if insertion point has moved
  XPos2 = .Text.PositionXOnPage
  YPos2 = .Text.PositionYOnPage

  ' make sure that don't keep checking the last misspelled word
  If ((XPos1 = XPos2) And (YPos1 = YPos2)) Then
    Exit Do
  End If
Loop While (Word <> "")

```

```
'Example: GetNamedProperty method
' This example creates a named property, 'ExampleProp' on the active document
' and assigns it a value. The value is then printed to the Lotus Script Output
' panel.
' RUNTIME DEPENDENCIES: You must have a document open for this script to work.

.ActiveDocument.SetNamedProperty "ExampleProp", "Here is some data."
Print .ActiveDocument.GetNamedProperty( "ExampleProp")
```

'Example: GetNameFromPage method

' This example prints the division name on page 1 to the LotusScript Output panel.

' RUNTIME DEPENDENCIES: You must have a document open for this script to work.

Print .Division.GetNameFromPage(1) ' print the 'user' name

Print .Division.GetNameFromPage(1, True) ' print the 'internal' name

'Example: GetObjectList method

'This example script displays a message box displaying the container name

' at the insertion point.

```
Dim ContName as String
Dim ContNum as Integer
ContNum = .Container.GetObjectList()
ContName = ""
Select Case Contnum
    Case 7
        ContName = "PageContainer"
    Case 8
        ContName = "FrameContainer"
    Case 23
        ContName = "TableContainer"
    Case 24
        ContName = "RowContainer"
    Case 25
        ContName = "CellContainer"
    Case 550
        ContName = "SuperTableContainer"
    Case 107
        ContName = "HeaderContainer"
    Case 108
        ContName = "FooterContainer"
End Select
If ContName <> "" then
    MessageBox "The current container is " & ContName, MB_OK, "Example Script"
End If
```

'Example: GetParaNumber method

'This example inserts three lines of text using the default outline sequence.

'For each line, the paragraph number is printed on the line.

```
.Text.Bullet.Name = "Default Outline"
```

```
.Text.Numbering.Position = 1
```

```
.Text.Indent.First = 360
```

```
.Text.Indent.Rest = 360
```

```
.Text.InsertText "This is heading paragraph 1. "
```

```
Lev1Number = .Text.GetParaNumber (1)
```

```
Lev2Number = .Text.GetParaNumber (2)
```

```
.Text.InsertText "The level 1 number is " & Lev1Number & " and the level 2 number is "  
& Lev2Number & ".", True
```

```
.Text.Bullet.Name = "Default Outline"
```

```
.Text.Numbering.Position = 2
```

```
.Text.Indent.First = 360
```

```
.Text.Indent.Rest = 360
```

```
.Text.InsertText "This is heading paragraph 2. "
```

```
Lev1Number = .Text.GetParaNumber (1)
```

```
Lev2Number = .Text.GetParaNumber (2)
```

```
.Text.InsertText "The level 1 number is " & Lev1Number & " and the level 2 number is "  
& Lev2Number & ".", True
```

```
.Text.Bullet.Name = "Default Outline"
```

```
.Text.Numbering.Position = 2
```

```
.Text.Indent.First = 360
```

```
.Text.Indent.Rest = 360
```

```
.Text.InsertText "This is heading paragraph 3. "
```

```
Lev1Number = .Text.GetParaNumber (1)
```

```
Lev2Number = .Text.GetParaNumber (2)
```

```
.Text.InsertText "The level 1 number is " & Lev1Number & " and the level 2 number is "  
& Lev2Number & "."
```

```
'Example: GetPasteFormatCategories method
' This example retrieves the paste format categories and tests whether several
' categories are available.
' RUNTIME DEPENDENCIES: You must have a document open for this script to work.

Dim FormatCategories As Integer
FormatCategories = .Text.GetPasteFormatCategories
Print FormatCategories
If (FormatCategories And &H1) Then
    MsgBox "Text format available.",MB_OK,"Example Script"
End If
If (FormatCategories And &H2) Then
    MsgBox "Table format available.",MB_OK,"Example Script"
End If
```



```

'Example: GetPosition method
' This example retrieves the position of the insertion point relative
' to the specified Marker object. A message box is displayed indicating
' the relative position.
' RUNTIME DEPENDENCIES: You must have a document open for this script to work.

Dim MarkerName As String
Dim NewMarker As TextMarker

.Type "[Enter]"
.Type "Sample Text"
.SelectWord

' Create a new marker for the selected text
MarkerName = .Mark($LWPMarkerTypeDefault)

' Move the insertion point one line above the new marker
.Text.Deselect
.Text.MoveUp(1)

Position = .Text.GetPosition(MarkerName)
If Position = 0 Then
    MsgBox "Insertion point is located within the marker"
Elseif Position = 1 Then
    MsgBox "Insertion point is located after the marker"
Elseif Position = -1 Then
    MsgBox "Insertion point is located before the marker"
Elseif Position = -2 Then
    MsgBox "Insertion point is located in a different text stream than the marker"
End If

```

'Example: GetPrinterInfo method

' This example prints the printer name for the current document to the Lotus

' Script Output panel.

' RUNTIME DEPENDENCIES: You must have a document open for this script to work.

Print .ActiveDocument.PrintManager.GetPrinterInfo()

'Example: GetRGB method

' This example creates a table with 5 rows and 5 columns into the current
' document. The background and pattern colors are changed for the current
' cell and the RGB color value is then printed to Lotus Script Output panel.
' RUNTIME DEPENDENCIES: You must have a document open for this script to work.

```
.CreateTable False, "Default Table", 5,5  
.Table.TableFill.Background.Pattern = $LtsFillNwToSeGrad  
.Table.CurrentCell.Background.BackColor.SetRGB 255,255,255  
.Table.TableFill.Background.Color.Override = $LwpColorOverrideRgb  
.Table.TableFill.Background.BackColor.Red = 82  
.Table.TableFill.Background.BackColor.Blue = 239  
.Table.TableFill.Background.BackColor.Green = 145  
.Table.TableFill.Background.BackColor.SetRGB 82,239,145  
.Table.TableFill.Background.BackColor.Override = $LwpColorOverrideRgb  
.Table.TableFill.FillStyle = $LwpTableFillStyleAll
```

```
Print .Table.CurrentCell.Background.BackColor.GetRGB()
```

Word Pro: Shadow property

{button ,AL(^H_FONTMETRICS_CLASS;H_FRAMECAPTIONOPTIONS_CLASS;H_LAYOUT_CLASS;H_CELLGROUP_LAYOUT_CLASS;H_CELL_LAYOUT_CLASS;H_CONNECTED_LAYOUT_CLASS;H_COLUMNGROUP_LAYOUT_CLASS;H_FOOTER_LAYOUT_CLASS;H_FRAME_LAYOUT_CLASS;H_FRAMEGROUP_LAYOUT_CLASS;H_DROP_CAP_LAYOUT_CLASS;H_GROUP_LAYOUT_CLASS;H_HEADER_LAYOUT_CLASS;H_NOTE_LAYOUT_CLASS;H_PAGE_LAYOUT_CLASS;H_ROWGROUP_LAYOUT_CLASS;H_ROW_LAYOUT_CLASS;H_RUBY_LAYOUT_CLASS;H_SUPER_TABLEGROUP_LAYOUT_CLASS;H_SUPER_TABLE_LAYOUT_CLASS;H_TABLE_LAYOUT_CLASS;H_END_NOTE_LAYOUT_CLASS;H_FOOTNOTE_LAYOUT_CLASS;H_TABLEHEADING_LAYOUT_CLASS;H_TOCSUPER_TABLE_LAYOUT_CLASS;H_PARAGRAPHBORDER_CLASS',0)} [See list of classes](#)

(Read-write) Turns the shadow effect on for a text object.

Data Type

Boolean

Syntax

[objectreference.]Shadowvalue = Shadowvalue

Shadowvalue = [objectreference.]Shadowvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Word Pro: Shadow property

{button ,AL('H_LAYOUT_CLASS;H_PARAGRAPHBORDER_CLASS',0)} [See list of classes](#)

{button ,AL('H_LAYOUT_PARAGRAPHBORDER_SHADOW_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-only) The shadow object for a layout or paragraph border object.

Data Type

[Shadow](#)

Syntax

Shadowvalue = [objectreference.]Shadowvalue

Legal values

Always contains an instance of the Shadow class.

Usage

Word Pro: ParagraphStyleName property

{button ,AL(^H_CLICKHERE_CLASS;H_TEXTMARKER_CLASS;H_TEXT_CLASS',0)} [See list of classes](#)

(Read-write) The name of the paragraph style assigned to the current paragraph.

Data Type

String

Syntax

paragraphstylevalue = [objectreference].ParagraphStyleName

[objectreference].ParagraphStyleName = paragraphstylevalue

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Word Pro: ParagraphStyles property

{button ,AL('H_FOUNDRY_CLASS',0)} [See list of classes](#)

{button ,AL('H_PARAGRAPHSTYLES_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-only) An object created from the ParagraphStyleCollection class. This object provides access to ParagraphStyle objects.

Data Type

[ParagraphStyleCollection](#)

Syntax

paragraphstylesvalue = [objectreference].ParagraphStyles

Legal values

Always contains an instance of the ParagraphStyleCollection class.

Usage

When accessed through the Foundry property on a Division object, the collection object in this property provides access to all the ParagraphStyle objects contained in that Division object.

When accessed through the AppFoundry property on the WPAApplication object, this collection object provides access to all the ParagraphStyle objects contained in the Word Pro Clipboard.

When accessed through the TempFoundry property on the WPAApplication object, this collection object provides access to all the ParagraphStyle objects placed in TempFoundry by WordPro or a script.

When accessed through the Foundry property on the WPAApplication object, this collection object provides access to all the ParagraphStyle objects contained in the currently active Division object.

This property is not used in the Foundry property on the TextDocument object.

For more information about collection classes, see [Overview: Word Pro LotusScript Collection Classes](#).

Word Pro: ParagraphStyle property

{button ,AL(^H_CLICKHERE_CLASS;H_TEXTMARKER_CLASS;H_TEXT_CLASS',0)} [See list of classes](#)

(Read-only)

Data Type

[ParagraphStyle](#)

Syntax

paragraphstylevalue = [objectreference].ParagraphStyle

Legal values

Always contains an instance of the ParagraphStyle class.

Usage

Word Pro: ParagraphSymbolChar property

{button ,AL(^H_CHARACTERSET_CLASS',0)} [See list of classes](#)

(Read-only)

Data Type

Integer

Syntax

paragraphsymbolcharvalue = [objectreference].ParagraphSymbolChar

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Word Pro: ParallelColumns property

{button ,AL(^H_WPAPPLICATION_CLASS;H_FOUNDRY_CLASS',0)} [See list of classes](#)

(Read-only) The ParallelColumns object which is uppermost in the focus when this property is called.

Data Type

[ParallelColumns](#)

Syntax

parallelcolumnsvalue = [objectreference].ParallelColumns

Legal values

Always contains an instance of the ParallelColumns class.

Usage

Word Pro: ParentMenuHWND property

{button ,AL(^H_MENUITEM_CLASS',0)} [See list of classes](#)

(Read-only) The handle to the parent window for a menu item.

Data Type

Long

Syntax

parentmenuhwndvalue = [objectreference].ParentMenuHWND

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Returns the handle of the parent window of a specific menu item. Use this handle to make API calls. This property is rarely used inside Word Pro under normal circumstances.

Word Pro: ParentName property

{button ,AL('H_DIVISION_CLASS;H_TEXTDOCUMENT_CLASS;H_TOCSUPERTABLELAYOUT_CLASS',0)} [See list of classes](#)

(Read-only)

Data Type

String

Syntax

parentnamevalue = [objectreference].ParentName

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Word Pro: Parent property

{button ,AL('H_BASEOBJECT_CLASS',0)} [See list of classes](#)

(Read-only) The parent (or containing) object for the object from which you are calling the Parent property. The object stored in this property is determined by the object from which you call this property.

Data Type

[BaseObject](#)

Syntax

parentvalue = [objectreference].Parent

Legal values

Data type for this property is BaseObject, which allows this property to contain any object derived directly or indirectly from the BaseObject class. However, this also means that you can only make use of the six properties inherited from BaseObject. For example, if the parent object in this property is a Text object, you can only access the six properties that text inherits from BaseObject.

Usage

The Parent property allows you to access an object's parent object. This is useful when you must get to the object that contains another object as a property.

For example, if you are working with a Color object and you are unsure of where the color object is contained, you can determine what object contains the color object by accessing that Color object's Parent property. Once you have the name of the object in the Parent property, you can assign that object to a variable with the same data type as the object. This gives you complete access to that object and all its members.

Word Pro: Partial property

{button ,AL(^H_CLICKHERE_CLASS;H_TEXTMARKER_CLASS;H_TEXT_CLASS',0)} [See list of classes](#)

(Read-write)

Data Type

Integer

Syntax

partialvalue = [objectreference].Partial

[objectreference].Partial = partialvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Word Pro: PathName property

{button ,AL(^H_DIVISION_CLASS;H_TEXTDOCUMENT_CLASS',0)} [See list of classes](#)

(Read-only)

Data Type

String

Syntax

pathnamevalue = [objectreference].PathName

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Word Pro: Path property

{button ,AL(^H_APPLICATION_CLASS;H_WPAPPLICATION_CLASS;H_DOCUMENT_CLASS;H_TEXTDOCUMENT_CLASS',0)} [See list of classes](#)

(Read-only) The path in which the Word Pro executable is installed. Word Pro uses this path to find the .DLLs necessary for launching and running Word Pro.

Data Type

String

Syntax

pathnamevalue = [objectreference].Path

Legal values

The value of this property cannot be set by a script.

Usage

Some Word Pro users may install Word Pro in a path that is different from the default path provided during installation. You can use this property when you need the path to the Word Pro application or its components, but you don't know if everyone installed to the same directory.

Word Pro: Pattern property

{button ,AL('H_BACKGROUND_CLASS;H_BORDER_CLASS',0)} [See list of classes](#)

{button ,AL('H_PATTERN_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write)

Data Type

Data type is [Variant](#), which allows the value of this property to be one of the constants listed below or its numeric equivalent (in parentheses).

Syntax

patternvalue = [objectreference].Pattern

[objectreference].Pattern = patternvalue

Legal values

[Background]

\$LtsFillBarLeftDiag (1056964681)

\$LtsFillBarRightDiag (1056964688)

\$LtsFillBasket (1056964710)

\$LtsFillBigCheck (1056964717)

\$LtsFillBottomTopGrad (1056964729)

\$LtsFillBrick (1056964712)

\$LtsFillBubbles (1056964724)

\$LtsFillChevron (1056964708)

\$LtsFillCircles (1056964719)

\$LtsFillClumpedNarrowDiagHatch (1056964696)

\$LtsFillClumpedZs (1056964723)

\$LtsFillDarkNarrowDiagHatch (1056964693)

\$LtsFillDiagBasket (1056964711)

\$LtsFillDiagBrick (1056964713)

\$LtsFillDiagHatch (1056964695)

\$LtsFillDiamonds (1056964725)

\$LtsFillDottedDarkHash (1056964864)

\$LtsFillDottedDarkHatch (2499)

\$LtsFillDottedZigzag (1056964726)

\$LtsFillDoubleLeftDiag (1056964684)

\$LtsFillDoubleRightDiag (1056964690)

\$LtsFillGray1 (1056964669)

\$LtsFillGray10 (1056964678)

\$LtsFillGray2 (1056964670)

\$LtsFillGray3 (1056964671)

\$LtsFillGray4 (1056964672)

\$LtsFillGray5 (1056964673)

\$LtsFillGray6 (1056964674)

\$LtsFillGray7 (1056964675)

\$LtsFillGray8 (1056964676)

\$LtsFillGray9 (1056964677)

\$LtsFillHoriz (1056964699)

\$LtsFillHorizBar (1056964698)

\$LtsFillHorizCheckerboard (1056964716)
\$LtsFillIrregularDiagScales (1056964721)
\$LtsFillLeftDiag (1056964682)
\$LtsFillLeftNarrowDiagHatch (1056964694)
\$LtsFillLeftRightGrad (1056964728)
\$LtsFillNarrowDoubleLeftDiag (1056964685)
\$LtsFillNarrowDoubleRightDiag (1056964863)
\$LtsFillNarrowHoriz (1056964697)
\$LtsFillNarrowVert (1056964701)
\$LtsFillNeToSwDiagStripGrad (1056964738)
\$LtsFillNeToSwGrad (1056964730)
\$LtsFillNone (1056964667)
\$LtsFillNwToSeDiagStripGrad (1056964739)
\$LtsFillNwToSeGrad (1056964731)
\$LtsFillRandomBar (1056964680)
\$LtsFillRandomSquare (1056964679)
\$LtsFillRegularCheck (1056964718)
\$LtsFillRegularHatch (1056964706)
\$LtsFillRightDiag (1056964689)
\$LtsFillRtLeftGrad (1056964744)
\$LtsFillRunningDash (1056964714)
\$LtsFillScalesDown (1056964722)
\$LtsFillScalesUp (1056964720)
\$LtsFillSolid (1056964668)
\$LtsFillSteel (1056964709)
\$LtsFillTinyHatch (1056964705)
\$LtsFillTopBottomGrad (1056964745)
\$LtsFillTripleLeftDiag (1056964686)
\$LtsFillTripleRightDiag (1056964691)
\$LtsFillVert (1056964703)
\$LtsFillVertBar (1056964702)
\$LtsFillVertCheckerboard (1056964715)
\$LtsFillWideHatch (1056964707)
\$LtsFillWideHoriz (1056964700)
\$LtsFillWideLeftDiag (1056964687)
\$LtsFillWideRightDiag (1056964692)
\$LtsFillWideVert (1056964704)
\$LtsIntervalYear (1056964854)
\$LtsTimeFormatHhMm24hr (1056964853)
\$LwpFillIndian3 (273)
\$LwpFillPattern (2000)
\$LwpFillPeachpie (274)

[BorderLines]

\$LtsBorderPatternBorderDot (1056964663)
\$LtsBorderPatternDashDot (1056964659)
\$LtsBorderPatternDashDotDot (1056964660)

\$LtsBorderPatternDashed (1056964662)
\$LtsBorderPatternDot (2498)
\$LtsBorderPatternDouble (1056964666)
\$LtsBorderPatternLongDash (1056964661)
\$LtsBorderPatternNone (1056964657)
\$LtsBorderPatternSolid (1056964658)
\$LwpBorderPattern13space (36)
\$LwpBorderPattern31space (37)
\$LwpBorderPatternButttdown (35)
\$LwpBorderPatternButtonup (34)
\$LwpBorderPatternCircle (41)
\$LwpBorderPatternDbIThick (51)
\$LwpBorderPatternDbIWavy (56)
\$LwpBorderPatternDeco1 (44)
\$LwpBorderPatternDeco2 (45)
\$LwpBorderPatternDeco3 (50)
\$LwpBorderPatternDiagonal (38)
\$LwpBorderPatternGirder (2571)
\$LwpBorderPatternPin (47)
\$LwpBorderPatternRain (46)
\$LwpBorderPatternRope (43)
\$LwpBorderPatternRose (48)
\$LwpBorderPatternStar (42)
\$LwpBorderPatternSunf (49)
\$LwpBorderPatternTaro (39)
\$LwpBorderPatternThickDbIwavy (58)
\$LwpBorderPatternThickThin (53)
\$LwpBorderPatternThickWavy (57)
\$LwpBorderPatternThinThick (54)
\$LwpBorderPatternThinThickThin (52)
\$LwpBorderPatternWarning (2379)
\$LwpBorderPatternWavy (55)
\$LwpLtsBorderPatternDot (40)

Usage

{button ,AL(`H_BACKCOLOR_PROPERTY_MEMDEF;H_BACKGROUND_PROPERTY_MEMDEF;H_COLOR_PROPERTY_MEMDEF',0)} [See related topics](#)

Word Pro: Percentage property

{button ,AL('H_JOIN_CLASS',0)} [See list of classes](#)

(Read-write) Defines the percentage property used in calculating the width and height of the bounding area of a scaleable join object.

Data Type

[Integer](#)

Syntax

percentagevalue = [objectreference].Percentage

[objectreference].Percentage = percentagevalue

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

A join object can be scaleable or fixed.

Scaleable join objects

You can set the ScaleMode property of a scaleable join object to scaling or no scaling. Setting this property to scaling causes the join object's width and height to be a function of the layout object's height and width, and also the Percentage property. (For more information, see ScaleMode property and Percentage property.)

Setting this property to no scaling causes the width and height of a join object to be the same as the width and height properties of the join object. The width and height of the join object does not change as the layout object's width and height changes.

The width and height of a scaleable join with a no scaling setting are fixed. However, you can change the width and height properties of a join object.

Fixed join objects

A fixed join object has predefined Word Pro width and height properties. Therefore, you cannot change its width or height.

The width and height of a scaleable join object is determined by the following algorithm:

```
IF join_percentage = 100 THEN
    joinwidth = container_width / 2
    joinheight = container_height / 2
ELSE
    IF container_height < container_width THEN
        join_width = (container_height / 200) * percentage
        join_height = (container_height / 200) * percentage
    ELSE
        join_width = (container_width / 200) * percentage
        join_height = (container_width / 200) * percentage
    END IF
END IF
```

Word Pro: PersonalData1 property

{button ,AL('H_PREFERENCES_CLASS',0)} [See list of classes](#)

{button ,AL('H_PERSONALDATA1_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write)

Data Type

[String](#)

Syntax

personaldata1value = [objectreference].PersonalData1

[objectreference].PersonalData1 = personaldata1value

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Word Pro: PersonalData2 property

{button ,AL('H_PREFERENCES_CLASS',0)} [See list of classes](#)

{button ,AL('H_PERSONALDATA2_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write)

Data Type

[String](#)

Syntax

personaldata2value = [objectreference].PersonalData2

[objectreference].PersonalData2 = personaldata2value

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Word Pro: PersonalData3 property

{button ,AL('H_PREFERENCES_CLASS',0)} [See list of classes](#)

{button ,AL('H_PERSONALDATA3_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write)

Data Type

String

Syntax

personaldata3value = [objectreference].PersonalData3

[objectreference].PersonalData3 = personaldata3value

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Word Pro: PersonalData4 property

{button ,AL('H_PREFERENCES_CLASS',0)} [See list of classes](#)

{button ,AL('H_PERSONALDATA4_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write)

Data Type

String

Syntax

personaldata4value = [objectreference].PersonalData4

[objectreference].PersonalData4 = personaldata4value

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Word Pro: PhoneNumber property

{button ,AL('H_PREFERENCES_CLASS',0)} [See list of classes](#)

{button ,AL('H_PHONENUMBER_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write)

Data Type

String

Syntax

phonenum value = [objectreference].PhoneNumber

[objectreference].PhoneNumber = phonenum value

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Word Pro: PitchAndFamily property

{button ,AL(^H_FONTMETRICS_CLASS',0)} [See list of classes](#)

(Read-only)

Data Type

[Integer](#)

Syntax

pitchandfamilyvalue = [objectreference].PitchAndFamily

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Word Pro: Placement property

{button ,AL(^H_RUBYLAYOUT_CLASS',0)} [See list of classes](#)

(Read-write)

Data Type

Integer.

Syntax

[objectreference].Placement = placementvalue

placementvalue = [objectreference].Placement

Legal values**Usage**

Word Pro: Plain property

{button ,AL(^H_FONT_CLASS',0)} [See list of classes](#)

(Read-write)

Data Type

Integer

Syntax

[objectreference].Plain = plainvalue

plainvalue = [objectreference].Plain

Legal values

Usage

Word Pro: PointSize property

{button ,AL('H_PREFERENCES_CLASS',0)} [See list of classes](#)

(Read-write)

Data Type

Single

Syntax

pointsizevalue = [objectreference].PointSize

[objectreference].PointSize = pointsizevalue

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Word Pro: PositionType property

{button ,AL('H_ICONBAR_CLASS',0)} [See list of classes](#)

{button ,AL('H_POSITIONTYPE_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Indicates whether or not the icon bar object is at the left, right, top, bottom, or floating position on the workspace.

Data Type

Variant (Enumerated)

Syntax

positiontypevalue = [objectreference].PositionType

[objectreference].PositionType = positiontypevalue

Legal values

<u>Value</u>	<u>Effect</u>
\$LwplconBarPosBottomOfWindow (390)	Indicates if a set of SmartIcons is at the bottom of the workspace in a fixed position.
\$LwplconBarPosFloating (391)	Indicates if a set of SmartIcons is in a floating position anywhere inside or outside the workspace.
\$LwplconBarPosLeftSideOfWindow (388)	Indicates if a set of SmartIcons is at the left side of the workspace in a fixed position.
\$LwplconBarPosRightSideOfWindow (389)	Indicates if a set of SmartIcons is at the right side of the workspace in a fixed position.
\$LwplconBarPosTopOfWindow (387)	Indicates if a set of SmartIcons is at the top of the workspace in a fixed position.

Usage

Allows you to move an icon bar object in the left, right, top, bottom, or floating position on the workspace.

Word Pro: PositionXInContainer property

{button ,AL('H_CLICKHERE_CLASS;H_TEXTMARKER_CLASS;H_TEXT_CLASS',0)} [See list of classes](#)

{button ,AL('H_POSITIONXINCONTAINER_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-only) The X coordinate of the insertion point relative to the top left corner of the container in which the insertion point is located.

Data Type

Long

Syntax

positionxincontainervalue = [objectreference].PositionXInContainer

Legal values

Data type is Long but the unit of measurement used for this property is Twips.

Usage

A container can be a comment note, a footer, a page, a cell, a frame, and so on.

Word Pro: PositionXOnPage property

```
{button ,AL('H_BASECONTAINER_CLASS;H_CELLCONTAINER_CLASS;H_FRAMECONTAINER_CLASS;H_NOTE  
CONTAINER_CLASS;H_RUBYCONTAINER_CLASS;H_DROPCAPCONTAINER_CLASS;H_PAGECONTAINER_  
CLASS;H_SUBPAGECONTAINER_CLASS;H_SUPERPAGECONTAINER_CLASS;H_ROWCONTAINER_CLASS;  
H_SUPERTABLECONTAINER_CLASS;H_TABLECONTAINER_CLASS;H_PARALLELCOLSCONTAINER_CLASS  
;H_TABLEONLYCONT_CLASS;H_GRAPHICOLEBJECT_CLASS;H_GRAPHIC_CLASS;H_OLEOBJECT_CLAS  
S;H_CLICKHERE_CLASS;H_TEXTMARKER_CLASS;H_TEXT_CLASS',0)} See list of classes
```

```
{button ,AL('H_POSITIONXONPAGE_PROPERTY_EXSCRIPT',1)} See example
```

(Read-only) The X coordinate of an object relative to the top left corner of the page.

Data Type

Long

Syntax

positionxonpagevalue = [objectreference].PositionXOnPage

Legal values

Data type is Long but the unit of measurement used for this property is Twips.

Usage

Word Pro: PositionYInContainer property

{button ,AL('H_CLICKHERE_CLASS;H_TEXTMARKER_CLASS;H_TEXT_CLASS',0)} [See list of classes](#)

{button ,AL('H_POSITIONYINCONTAINER_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-only) The Y coordinate of the insertion point relative to the top left corner of the container in which the insertion point is located.

Data Type

Long

Syntax

positionyincontainervalue = [objectreference].PositionYInContainer

Legal values

Data type is Long but the unit of measurement used for this property is Twips.

Usage

A container can be a comment note, a footer, a page, a cell, a frame, and so on.

Word Pro: PositionYOnPage property

```
{button ,AL('H_BASECONTAINER_CLASS;H_CELLCONTAINER_CLASS;H_FRAMECONTAINER_CLASS;H_NOTE  
CONTAINER_CLASS;H_RUBYCONTAINER_CLASS;H_DROPDOWNCONTAINER_CLASS;H_PAGECONTAINER_  
CLASS;H_SUBPAGECONTAINER_CLASS;H_SUPERPAGECONTAINER_CLASS;H_ROWCONTAINER_CLASS;  
H_SUPERTABLECONTAINER_CLASS;H_TABLECONTAINER_CLASS;H_PARALLELCOLSCONTAINER_CLASS  
;H_TABLEONLYCONT_CLASS;H_GRAPHICOLEBJECT_CLASS;H_GRAPHIC_CLASS;H_OLEOBJECT_CLAS  
S;H_CLICKHERE_CLASS;H_TEXTMARKER_CLASS;H_TEXT_CLASS',0)} See list of classes
```

```
{button ,AL('H_POSITIONYONPAGE_PROPERTY_EXSCRIPT',1)} See example
```

(Read-only) The Y coordinate of an object relative to the top left corner of the page.

Data Type

Long

Syntax

positionyonpagevalue = [objectreference].PositionYOnPage

Legal values

Data type is Long but the unit of measurement used for this property is Twips.

Usage

Word Pro: Position property

{button ,AL(`H_ALIGNMENT_CLASS;H_FRAMECAPTIONOPTIONS_CLASS;H_NUMBERING_CLASS;H_OUTSEQU
TEM_CLASS',0)} [See list of classes](#)

{button ,AL(`H_POSITION_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Indicates the position of an object in units of Twips.

Data Type

Long

Syntax

positionvalue = [objectreference].Position

[objectreference].Position = positionvalue

Legal values

Data type is Long, but the unit of measurement used for this property is Twips. There are 1440 Twips per inch.

Usage

In the Alignment class, this property corresponds to the "Distance from margin" value on the Misc panel of the Text InfoBox. It indicates the distance between the margin to where the decimal point lines up, in numeric alignment.

{button ,AL(`H_ALIGNMENTTYPE_PROPERTY_MEMDEF',0)} [See related topics](#)

Word Pro: PowerFields property

{button ,AL('H_FOUNDRY_CLASS',0)} [See list of classes](#)

(Read-only) An object created from the PowerFieldCollection class. This object provides access to PowerField objects.

Data Type

[PowerFieldCollection](#)

Syntax

powerfieldsvalue = [objectreference].PowerFields

Legal values

Always contains an instance of the PowerFieldCollection class.

Usage

When accessed through the Foundry property on a Division object, the collection object in this property provides access to all the PowerField objects contained in that Division object.

When accessed through the AppFoundry property on the WPAApplication object, this collection object provides access to all the PowerField objects contained in the Word Pro Clipboard.

When accessed through the TempFoundry property on the WPAApplication object, this collection object provides access to all the PowerField objects placed in TempFoundry by WordPro or a script.

When accessed through the Foundry property on the WPAApplication object, this collection object provides access to all the PowerField objects contained in the currently active Division object.

This property is not used in the Foundry property on the TextDocument object.

For more information about collection classes, see [Overview: Word Pro LotusScript Collection Classes](#).

Word Pro: PowerField property

{button ,AL('H_OPTIONS_CLASS',0)} [See list of classes](#)

{button ,AL('H_POWERFIELD_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Displays power field formulas in text.

Data Type

[Integer](#)

Syntax

powerfieldvalue = [objectreference].PowerField

[objectreference].PowerField = powerfieldvalue

Legal values

The legal values for this property are -1 and 0 but you may use the LotusScript constants of True (-1) to show power fields, and False (0) to hide power fields, instead of the integer values.

Usage

Equivalent to choosing File - Document Properties - Document, clicking the Fields tab and selecting "Show power field formulas in text."

Word Pro: Preferences property

{button ,AL(^H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

(Read-only) The preferences object for the currently active session of Word Pro.

Data Type

[Preferences](#)

Syntax

preferencesvalue = [objectreference].Preferences

Legal values

Always contains an instance of the Preferences class.

Usage

The properties in this object contain the settings in the Word Pro Preferences dialog box. You can open this dialog box in Word Pro by choosing File - User Setup - Word Pro Preferences.

Word Pro: Prefix property

{button ,AL('H_NUMERICFORMATSUBSET_CLASS',0)} [See list of classes](#)

{button ,AL('H_PREFIX_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Allows you to include prefix text with a specific number format.

Data Type

String

Syntax

prefixvalue = [objectreference].Prefix

[objectreference].Prefix = prefixvalue

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Equivalent to the "Text before" option in the Edit Format dialog box. You can access this dialog box by clicking the Format Options button on the Number Format panel of the InfoBox for cell layout objects.

Word Pro: Presentation property

```
{button ,AL('H_BASECONTAINER_CLASS;H_CELLCONTAINER_CLASS;H_FRAMECONTAINER_CLASS;H_NOTE  
CONTAINER_CLASS;H_RUBYCONTAINER_CLASS;H_DROPDOWNCONTAINER_CLASS;H_PAGECONTAINER_  
CLASS;H_SUBPAGECONTAINER_CLASS;H_SUPERPAGECONTAINER_CLASS;H_ROWCONTAINER_CLASS;  
H_SUPERTABLECONTAINER_CLASS;H_TABLECONTAINER_CLASS;H_PARALLELCOLSCONTAINER_CLASS  
;H_TABLEONLYCONT_CLASS',0)} See list of classes
```

```
{button ,AL('H_PRESENTATION_PROPERTY_EXSCRIPT',1)} See example
```

(Read-only) Returns the presentation object of any container.

Data Type

Presentation

Syntax

presentationvalue = [objectreference].Presentation

Legal values

Always contains an instance of the Presentation class.

Usage

Word Pro: PreviousClickHere property

{button ,AL(^H_CLICKHERE_CLASS',0)} [See list of classes](#)

(Read-only) The name of the ClickHere block which precedes the current ClickHere in the division (uses Tab order.)

Data Type

String

Syntax

previousclickherevalue = [objectreference].PreviousClickHere

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Word Pro: PreviousCursorableDivision property

{button ,AL(^H_DIVISION_CLASS;H_TEXTDOCUMENT_CLASS',0)} [See list of classes](#)

(Read-only)

This language element has not yet been defined. You can check for a newer Help file on the Lotus web site.

Data Type

String

Syntax

previouscursorabledivisionvalue = [objectreference].PreviousCursorableDivision

Legal values

This property is read-only. The value of this property cannot be set by a script.

Usage

Word Pro: PreviousName property

{button ,AL(^H_DIVISION_CLASS;H_TEXTDOCUMENT_CLASS',0)} [See list of classes](#)

(Read-only)

Data Type

String

Syntax

previousnamevalue = [objectreference].PreviousName

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Word Pro: PreviousNeighbor property

{button ,AL(^H_DIVISION_CLASS;H_TEXTDOCUMENT_CLASS',0)} [See list of classes](#)

(Read-only)

Data Type

String

Syntax

previousneighborvalue = [objectreference].PreviousNeighbor

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Word Pro: PrintDestination property

{button ,AL('H_PRINTMANAGER_CLASS',0)} [See list of classes](#)

{button ,AL('H_PRINTDESTINATION_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) The port the printer uses for output, such as LPT1.

Data Type

[String](#)

Syntax

printdestinationvalue = [objectreference].PrintDestination

[objectreference].PrintDestination = printdestinationvalue

Legal values

The legal values for this property are dependent on the operating system and its valid printer ports.

Usage

Word Pro: PrinterName property

{button ,AL('H_PRINTMANAGER_CLASS',0)} [See list of classes](#)

{button ,AL('H_PRINTERNAME_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) The name of the printer assigned to the specified TextDocument object.

Data Type

String

Syntax

printernamevalue = [objectreference].PrinterName

[objectreference].PrinterName = printernamevalue

Legal values

A String value that corresponds to the name of a printer as it is seen your operating system's printer setup utility.

Usage

This property value is case-sensitive. For example, Word Pro does not equate "HP Laserjet" to "HP LaserJet."

After assigning a new value to the PrinterName property, use the UpdatePrinterChanges method to update the TextDocument object's print settings.

Word Pro: PrintGraphics property

{button ,AL('H_PRINTSETTINGS_CLASS',0)} [See list of classes](#)

{button ,AL('H_PRINTGRAPHICS_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Indicates whether or not Word Pro prints any graphics in a document.

Data Type

[Integer](#)

Syntax

printgraphicsvalue = [objectreference].PrintGraphics

[objectreference].PrintGraphics = printgraphicsvalue

Legal values

<u>Value</u>	<u>Effect</u>
0	Word Pro will not print graphics.
1	Word Pro will print graphics.

Usage

Equivalent to the "Without pictures" option in the Print Options dialog box.

Word Pro: PrintManager property

{button ,AL('H_TEXTDOCUMENT_CLASS',0)} [See list of classes](#)

{button ,AL('H_PRINTMANAGER_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-only)

Data Type

[PrintManager](#)

Syntax

printmanagervalue = [objectreference].PrintManager

Legal values

Always contains an instance of the PrintManager class.

Usage

Word Pro: PrintPagesFrom property

{button ,AL('H_PRINTSETTINGS_CLASS',0)} [See list of classes](#)

{button ,AL('H_PRINTPAGESFROM_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Specifies the first page number within a range of pages that you want to print in a document.

Data Type

[Integer](#)

Syntax

printpagesfromvalue = [objectreference].PrintPagesFrom

[objectreference].PrintPagesFrom = printpagesfromvalue

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Equivalent to choosing File - Print and selecting a number in the "Pages from" box.

For this property to take effect, the PrintRange property must be set to \$LwpPrintRangeOfPages.

Word Pro: PrintPagesTo property

{button ,AL('H_PRINTSETTINGS_CLASS',0)} [See list of classes](#)

{button ,AL('H_PRINTPAGESTO_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Specifies the last page within a range of pages that you want to print in a document.

Data Type

[Integer](#)

Syntax

printpagestovalue = [objectreference].PrintPagesTo

[objectreference].PrintPagesTo = printpagestovalue

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Equivalent to choosing File - Print and selecting a number in the "Pages to" box.

Note The PrintRange property must be set to \$LwpPrintRangeOfPages for this property to take effect.

Word Pro: PrintPageType property

{button ,AL('H_PRINTSETTINGS_CLASS',0)} [See list of classes](#)

{button ,AL('H_PRINTPAGETYPE_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Specifies the printing of odd and even pages, only even pages, or only odd pages in a document.

Data Type

Data type is [Variant](#), which allows the value of this property to be one of the constants listed below or its numeric equivalent (in parentheses).

Syntax

printpagetypevalue = [objectreference].PrintPageType

[objectreference].PrintPageType = printpagetypevalue

Legal values

<u>Value</u>	<u>Effect</u>
\$LwpPrintPageEvenAndOddPages (1651)	Prints odd and even pages in a document.
\$LwpPrintPageEvenPages (1652)	Prints only even pages in a document.
\$LwpPrintPageOddPages (1653)	Prints only odd pages in a document.

Usage

Equivalent to choosing File - Print and selecting an option from the "Including" box.

Word Pro: PrintRange property

{button ,AL('H_PRINTSETTINGS_CLASS',0)} [See list of classes](#)

{button ,AL('H_PRINTRANGE_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Allows you to specify which pages in a document you want to print. You can print all pages, the current page or division, specific pages or divisions, or a range of pages, such as pages 2 - 10.

Data Type

Data type is [Variant](#), which allows the value of this property to be one of the constants listed below or its numeric equivalent (in parentheses).

Syntax

prinrangevalue = [objectreference].PrintRange

[objectreference].PrintRange = prinrangevalue

Legal values

<u>Value</u>	<u>Effect</u>
\$LtsPrintRangeAllPages (1056964848)	Prints all the pages in a document.
\$LtsPrintRangeCurrentPage (1056964849)	Prints only the current page in a document.
\$LtsPrintRangeSelectedPages (1056964850)	Prints only selected pages in a document.
\$LwpPrintRangeCurrentDivision (1655)	Prints only the current division in a document.
\$LwpPrintRangeRangeOfPages (1654)	Allows you to specify a range of pages to print in a document.
\$LwpPrintRangeSelectedDivisions (1656)	Allows you to specify a range of divisions to print in a document.

Usage

Equivalent to choosing File - Print, and selecting the desired options in the "Print range" or "Print" section in the Print dialog box.

Word Pro: PrintSettings property

{button ,AL('H_DOCUMENT_CLASS;H_TEXTDOCUMENT_CLASS',0)} [See list of classes](#)

{button ,AL('H_PRINTSETTINGS_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-only)

Data Type

[PrintSettings](#)

Syntax

printsettingsvalue = [objectreference].PrintSettings

Legal values

Always contains an instance of the PrintSettings class.

Usage

Word Pro: Private property

{button ,AL(^H_POWERFIELD_CLASS',0)} [See list of classes](#)

(Read-only)

Data Type

Integer

Syntax

Privatevalue = [Objectreference].Private

Legal values**Usage**

Word Pro: ProductVersion property

{button ,AL(^H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL(^H_PRODUCTVERSION_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-only)

This language element has not yet been defined. You can check for a newer Help file on the Lotus web site.

Data Type

Integer

Syntax

productversionvalue = [objectreference].ProductVersion

Legal values

This property is read-only. The value of this property cannot be set by a script.

Usage

Word Pro: ProgID property

{button ,AL(^H_OLEOBJECT_CLASS',0)} [See list of classes](#)

(Read-only)

Note This property is not supported for the class OLEObject within OS/2.

Data Type

String.

Syntax

progidvalue = [objectreference].ProgID

Legal values**Usage**

Word Pro: PromptHidden property

{button ,AL('H_CLICKHERE_CLASS',0)} [See list of classes](#)

{button ,AL('H_PROMPTHIDDEN_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) True hides the Prompt text; False makes the Prompt text visible.

Data Type

[Integer](#)

Syntax

prompthiddenvalue = [objectreference].PromptHidden

[objectreference].PromptHidden = prompthiddenvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Word Pro: Prompt property

{button ,AL('H_GRAPHIC_CLASS;H_CLICKHERE_CLASS',0)} [See list of classes](#)

{button ,AL('H_PROMPT_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-only) The prompt text for a ClickHere object.

Data Type

[Text](#)

Syntax

promptvalue = [objectreference].Prompt

Legal values

Always contains an instance of the Text class.

Usage

Word Pro: ProtectedMode property

{button ,AL('H_ATTRIBUTES_CLASS',0)} [See list of classes](#)

{button ,AL('H_PROTECTEDMODE_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Enables or disables the protected attribute.

Data Type

[Integer](#)

Syntax

protectedmodevalue = [objectreference].ProtectedMode

[objectreference].ProtectedMode = protectedmodevalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

For the Alignment object in the Text class, the ProtectedMode property is equivalent to the "Protected" attribute on the Text Font properties panel of the InfoBox.

{button ,AL('H_ISTEXTLOCKED_PROPERTY_MEMDEF',0)} [See related topics](#)

Word Pro: QueueName property

{button ,AL('H_PRINTMANAGER_CLASS',0)} [See list of classes](#)

(Read-write)

Data Type

String

Syntax

queuenamevalue = [objectreference].QueueName

[objectreference].QueueName =queuenamevalue

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Word Pro: ReadCompressed property

{button ,AL('H_DIVISION_CLASS;H_TEXTDOCUMENT_CLASS',0)} [See list of classes](#)

(Read-only)

Data Type

Integer.

Syntax

readcompressedvalue = [objectreference].ReadCompressed

Legal values**Usage**

Word Pro: ReadOnly property

{button ,AL('H_DOCUMENT_CLASS;H_TEXTDOCUMENT_CLASS',0)} [See list of classes](#)

{button ,AL('H_READONLY_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write)

Data Type

[Integer](#)

Syntax

readonly value = [objectreference].ReadOnly

[objectreference].ReadOnly = readonlyvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Word Pro: Recursive property

{button ,AL('H_HTMLOPTIONS_CLASS',0)} [See list of classes](#)

{button ,AL('H_RECURSIVE_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write)

This language element has not yet been defined. You can check for a newer Help file on the Lotus web site.

Data Type

Integer.

Syntax

recursivevalue = [objectreference].Recursive

[objectreference].Recursive = recursivevalue

Legal values

The legal values for this property are -1 and 0 but you may use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Equivalent to choosing File - Internet - HTML Export Assistant and, on the Preview & Save panel, selecting "Include linked files."

Word Pro: Red property

{button ,AL('H_COLOR_CLASS',0)} [See list of classes](#)

{button ,AL('H_RED_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) The red component of a color.

Data Type

[Integer](#)

Syntax

redvalue = [objectreference].Red

[objectreference].Red = redvalue

Legal values

The value of the Red property can range from 0 - 255.

Usage

Use the Red property to access the current level of red in a specific object's color. For example, if you want to change the red value of a frame's background color, you can use the following statement:

```
.Frame.Layout.Background.Color.Red = 128
```


Word Pro: RelativeIndent property

{button ,AL(^H_CLICKHERE_CLASS;H_TEXTMARKER_CLASS;H_PARAGRAPHSTYLE_CLASS;H_TEXT_CLASS',
0)} [See list of classes](#)

(Read-only)

Data Type

RelativeIndent

Syntax

relativeindentvalue = [objectreference].RelativeIndent

Legal values

Always contains an instance of the RelativeIndent class.

Usage

Word Pro: RelativePageNum property

```
{button ,AL('H_BASECONTAINER_CLASS;H_CELLCONTAINER_CLASS;H_FRAMECONTAINER_CLASS;H_NOTE  
CONTAINER_CLASS;H_RUBYCONTAINER_CLASS;H_DROPDOWNCONTAINER_CLASS;H_PAGECONTAINER_  
CLASS;H_SUBPAGECONTAINER_CLASS;H_SUPERPAGECONTAINER_CLASS;H_ROWCONTAINER_CLASS;  
H_SUPERTABLECONTAINER_CLASS;H_TABLECONTAINER_CLASS;H_PARALLELCOLSCONTAINER_CLASS  
;H_TABLEONLYCONT_CLASS',0)} See list of classes
```

```
{button ,AL('H_RELATIVEPAGENUM_PROPERTY_EXSCRIPT',1)} See example
```

(Read-only) The number of a page relative to the beginning of its division.

Data Type

Integer

Syntax

relativepagenumvalue = [objectreference].RelativePageNum

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Word Pro: RelativeType property

{button ,AL(^H_LAYOUT_CLASS;H_CELLGROUPLAYOUT_CLASS;H_CELLLAYOUT_CLASS;H_CONNECTEDLAYOUT_CLASS;H_COLUMNGROUPLAYOUT_CLASS;H_FOOTERLAYOUT_CLASS;H_FRAMELAYOUT_CLASS;H_FRAMEGROUPLAYOUT_CLASS;H_DROPDOWNLAYOUT_CLASS;H_GROUPLAYOUT_CLASS;H_HEADERLAYOUT_CLASS;H_NOTELAYOUT_CLASS;H_PAGELAYOUT_CLASS;H_ROWGROUPLAYOUT_CLASS;H_ROWLAYO
UT_CLASS;H_RUBYLAYOUT_CLASS;H_SUPERTABLEGROUPLAYOUT_CLASS;H_SUPERTABLELAYOUT
_CLASS;H_TABLELAYOUT_CLASS;H_ENDNOTELAYOUT_CLASS;H_FOOTNOTELAYOUT_CLASS;H_TABLE
HEADINGLAYOUT_CLASS;H_TOCSUPERTABLELAYOUT_CLASS',0)} [See list of classes](#)

(Read-write) Determines where the layout is anchored in the parent layout.

Data Type

Data type is [Variant](#), which allows the value of this property to be one of the constants listed below or its numeric equivalent (in parentheses).

Syntax

relativetypevalue = [objectreference].RelativeType

[objectreference].RelativeType = relativetypevalue

Legal values

<u>Value</u>	<u>Effect</u>
\$LwpRelativeTypeLytContent (1670)	Anchors the layout relative to the content box of the parent.
\$LwpRelativeTypeLytInline (1668)	Anchors the layout to the text flow as a single character and affects the line height accordingly.
\$LwpRelativeTypeLytInlineNewline (1669)	Places the layout on a new line by itself.
\$LwpRelativeTypeLytInlineVert (1671)	Anchors the layout so that it always moves vertically.
\$LwpRelativeTypeLytPara (1667)	Anchors the layout in the text relative to a paragraph.
\$LwpRelativeTypeLytParent (1666)	The coordinates of the layout are relative to the parent layout and the anchor position is on the parent container.

Usage

Word Pro: RelativeXDistance property

```
{button ,AL('H_LAYOUT_CLASS;H_CELLGROUPLAYOUT_CLASS;H_CELLLAYOUT_CLASS;H_CONNECTEDLAY  
OUT_CLASS;H_COLUMNGROUPLAYOUT_CLASS;H_FOOTERLAYOUT_CLASS;H_FRAMELAYOUT_CLASS;H  
_FRAMEGROUPLAYOUT_CLASS;H_DROP_CAPLAYOUT_CLASS;H_GROUPLAYOUT_CLASS;H_HEADERLAY  
OUT_CLASS;H_NOTELAYOUT_CLASS;H_PAGELAYOUT_CLASS;H_ROWGROUPLAYOUT_CLASS;H_ROWLA  
YOUT_CLASS;H_RUBYLAYOUT_CLASS;H_SUPERTABLEGROUPLAYOUT_CLASS;H_SUPERTABLELAYOUT  
_CLASS;H_TABLELAYOUT_CLASS;H_ENDNOTELAYOUT_CLASS;H_FOOTNOTELAYOUT_CLASS;H_TABLE  
HEADINGLAYOUT_CLASS;H_TOCSUPERTABLELAYOUT_CLASS',0)} See list of classes
```

```
{button ,AL('H_RELATIVEXDISTANCE_PROPERTY_EXSCRIPT',1)} See example
```

(Read-write) The horizontal length of the anchor tether from a layout to its anchor point in the parent layout.

Data Type

Long

Syntax

relativexdistancevalue = [objectreference].RelativeXDistance

[objectreference].RelativeXDistance = relativexdistancevalue

Legal values

Data type is Long but the unit of measurement used for this property is Twips. There are 1440 Twips per inch.

Usage

Word Pro: RelativeYDistance property

```
{button ,AL('H_LAYOUT_CLASS;H_CELLGROUPLAYOUT_CLASS;H_CELLLAYOUT_CLASS;H_CONNECTEDLAY  
OUT_CLASS;H_COLUMNGROUPLAYOUT_CLASS;H_FOOTERLAYOUT_CLASS;H_FRAMELAYOUT_CLASS;H  
_FRAMEGROUPLAYOUT_CLASS;H_DROP_CAPLAYOUT_CLASS;H_GROUPLAYOUT_CLASS;H_HEADERLAY  
OUT_CLASS;H_NOTELAYOUT_CLASS;H_PAGELAYOUT_CLASS;H_ROWGROUPLAYOUT_CLASS;H_ROWLA  
YOUT_CLASS;H_RUBYLAYOUT_CLASS;H_SUPERTABLEGROUPLAYOUT_CLASS;H_SUPERTABLELAYOUT  
_CLASS;H_TABLELAYOUT_CLASS;H_ENDNOTELAYOUT_CLASS;H_FOOTNOTELAYOUT_CLASS;H_TABLE  
HEADINGLAYOUT_CLASS;H_TOCSUPERTABLELAYOUT_CLASS',0)} See list of classes
```

```
{button ,AL('H_RELATIVEYDISTANCE_PROPERTY_EXSCRIPT',1)} See example
```

(Read-write) The vertical length of the anchor tether from a layout to its anchor point in the parent layout.

Data Type

Long

Syntax

relativeydistancevalue = [objectreference].RelativeYDistance

[objectreference].RelativeYDistance = relativeydistancevalue

Legal values

Data type is Long but the unit of measurement used for this property is Twips. There are 1440 Twips per inch.

Usage

Word Pro: Relative property

{button ,AL('H_INDENT_CLASS;H_RELATIVEINDENT_CLASS',0)} [See list of classes](#)

{button ,AL('H_RELATIVE_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write)

Data Type

The data type for this property is [Variant](#) which allows the value of this property to be one of the constants listed below or its numeric equivalent (in parentheses).

Syntax

relativevalue = [objectreference].Relative

[objectreference].Relative = relativevalue

Legal values

\$LwpIndentPropertyAll (403)

\$LwpIndentPropertyBodyonly (409)

\$LwpIndentPropertyEnabled (410)

\$LwpIndentPropertyEvery (402)

\$LwpIndentPropertyFirst (404)

\$LwpIndentPropertyHang (407)

\$LwpIndentPropertyRelative (411)

\$LwpIndentPropertyRest (405)

\$LwpIndentPropertyRight (406)

\$LwpIndentPropertySidesequal (408)

Usage

Equivalent to the "Hierarchy indent for next lower level" combobox in the Indent Options dialog box. Indicates which indent property will be used to set the relative indent level to the preceding paragraph.

Word Pro: ReleaseNumber property

{button ,AL(`H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL(`H_RELEASENUMBER_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-Only)

Data Type

Integer

Syntax

releasenumvalue = [objectreference].ReleaseNumber

Legal values

Usage

Word Pro: RenderedPageNumber property

{button ,AL('H_CLICKHERE_CLASS;H_TEXTMARKER_CLASS;H_TEXT_CLASS',0)} [See list of classes](#)

{button ,AL('H_RENDEREDPAGENUMBER_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-only)

Data Type

[String](#)

Syntax

renderedpagenumbervalue = [objectreference].RenderedPageNumber

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Word Pro: ReplaceAttributes property

{button ,AL('H_FINDANDREPLACE_CLASS',0)} [See list of classes](#)

(Read-only) Enables the user to replace specific text attributes in Find & Replace.

Data Type

[Attributes](#)

Syntax

replaceattributesvalue = [objectreference].ReplaceAttributes

Legal values

Always contains an instance of the Attributes class. The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Use this property to replace text attributes in Find & Replace. If set to True, replaces the text attributes that match the user setting. Equivalent to choosing Edit - Find & Replace Text, clicking Options, clicking the Font button in the "Replace options" section, and choosing text attributes.

Word Pro: ReplaceExactCase property

{button ,AL(^H_FINDANDREPLACE_CLASS',0)} [See list of classes](#)

(Read-write) Enables the user to replace the exact case of a word or phrase in Find & Replace.

Data Type

Integer

Syntax

replaceexactcasevalue = [objectreference].ReplaceExactCase

[objectreference].ReplaceExactCase = replaceexactcasevalue

Legal values

Always contains an instance of the Text class. The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Use this property to check and manipulate the setting for replacing the exact case of a word or phrase in Find & Replace. If True, replaces the word that matches the user setting. Equivalent to choosing Edit - Find & Replace Text, clicking Options, and selecting "Exact case" in the "Replace options" section.

Word Pro: ReplaceFont property

{button ,AL(^H_FINDANDREPLACE_CLASS',0)} [See list of classes](#)

(Read-only) Enables the user to replace a text font in Find & Replace.

Data Type

[Font](#)

Syntax

replacefontvalue = [objectreference].ReplaceFont

Legal values

Always contains an instance of the Font class. The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Use this property to replace a font in Find & Replace. If True, replaces the font that matches the user setting. Equivalent to choosing Edit - Find & Replace Text, clicking Options, clicking the Font button in the "Replace options" section, and selecting a font in the "Font name" list box on the Replace with panel.

Word Pro: ReplaceLanguage property

{button ,AL(^H_FINDANDREPLACE_CLASS',0)} [See list of classes](#)

(Read-only) Enables the user to replace the language used in Find & Replace.

Data Type

[Language](#)

Syntax

replacelanguagevalue = [objectreference].ReplaceLanguage

Legal values

Always contains an instance of the Language class.

Usage

Word Pro: Replacements property

{button ,AL('H_FORMATCHECKPREF_CLASS',0)} [See list of classes](#)

(Read-write)

Data Type

Long (Enumerated Bitmask)

ReplacementChoices

Syntax

replacementsvalue = [objectreference].Replacements

[objectreference].Replacements = replacementsvalue

Legal values

LwpReplacementChoicesListBox22 (&H200000)

LwpReplacementChoicesListBox1 (&H1)

LwpReplacementChoicesListBox10 (&H200)

LwpReplacementChoicesListBox11 (&H400)

LwpReplacementChoicesListBox12 (&H800)

LwpReplacementChoicesListBox13 (&H1000)

LwpReplacementChoicesListBox14 (&H2000)

LwpReplacementChoicesListBox15 (&H4000)

LwpReplacementChoicesListBox16 (&H8000)

LwpReplacementChoicesListBox17 (&H10000)

LwpReplacementChoicesListBox18 (&H20000)

LwpReplacementChoicesListBox19 (&H40000)

LwpReplacementChoicesListBox2 (&H2)

LwpReplacementChoicesListBox20 (&H80000)

LwpReplacementChoicesListBox21 (&H100000)

LwpReplacementChoicesListBox22 (&H200000)

LwpReplacementChoicesListBox23 (&H400000)

LwpReplacementChoicesListBox24 (&H800000)

LwpReplacementChoicesListBox25 (&H1000000)

LwpReplacementChoicesListBox26 (&H2000000)

LwpReplacementChoicesListBox3 (&H4)

LwpReplacementChoicesListBox4 (&H8)

LwpReplacementChoicesListBox5 (&H10)

LwpReplacementChoicesListBox6 (&H20)

LwpReplacementChoicesListBox7 (&H40)

LwpReplacementChoicesListBox8 (&H80)

LwpReplacementChoicesListBox9 (&H100)

LwpReplacementChoicesListBoxAll (&H3FFFFFF)

Usage

Word Pro: ReplaceString property

{button ,AL('H_FINDANDREPLACE_CLASS',0)} [See list of classes](#)

{button ,AL('H_REPLACESTRING_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Enables the user to type what to replace in Find & Replace.

Data Type

[String](#)

Syntax

replacestringvalue = [objectreference].ReplaceString

[objectreference].ReplaceString = replacestringvalue

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Equivalent to choosing Edit - Find & Replace Text and typing a text or character string in the "Replace with" box.

Word Pro: ReplaceStyleName property

{button ,AL('H_FINDANDREPLACE_CLASS',0)} [See list of classes](#)

(Read-write) Enables the user to replace a paragraph style in Find & Replace.

Data Type

String

Syntax

replacestylevalue = [objectreference].ReplaceStyleName

[objectreference].ReplaceStyleName = replacestylevalue

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Equivalent to choosing Edit - Find & Replace Text, typing the character, ^p, in the "Replace with" box, clicking Options, clicking the Font button in the "Replace options" section, and choosing a paragraph style in the "Style" list box.

Word Pro: ReplaceWithProperties property

{button ,AL(^H_FINDANDREPLACE_CLASS',0)} [See list of classes](#)

(Read-write) Enables the user to replace font properties in Find & Replace.

Data Type

Integer

Syntax

replacewithpropertiesvalue = [objectreference].ReplaceWithProperties

[objectreference].ReplaceWithProperties = replacewithpropertiesvalue

Legal values

The legal values for this property are -1 and 0 but you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Use this property to replace font properties in Find & Replace. If True, replaces the font properties that match the user setting. Equivalent to choosing Edit - Find & Replace Text, clicking Options, selecting "Include properties," clicking the Font button in the "Replace options" section, and selecting properties.

Word Pro: RequestRemarkOnClose property

{button ,AL('H_DOCCONTROL_CLASS',0)} [See list of classes](#)

{button ,AL('H_REQUESTREMARKONCLOSE_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Indicates whether or not a dialog box displays when a document closes and requests the current editor to enter a remark.

Data Type

[Integer](#)

Syntax

requestremarkonclosevalue = [objectreference].RequestRemarkOnClose

[objectreference].RequestRemarkOnClose = requestremarkonclosevalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Equivalent to choosing File - TeamSecurity and selecting "Request Editor's remark on close" on the Editing Rights panel.

Word Pro: RequireStartupScripts property

{button ,AL('H_DOCCONTROL_CLASS',0)} [See list of classes](#)

{button ,AL('H_REQUIRESTARTUPSCRIPTS_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Indicates whether or not startup scripts are required to run when a file is opened.

Data Type

[Integer](#)

Syntax

requirestartupscriptsvalue = [objectreference].RequireStartupScripts

[objectreference].RequireStartupScripts = requirestartupscriptsvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Equivalent to choosing File - TeamSecurity and selecting "Require running of startup scripts" on the Other Protection panel.

Word Pro: ResetOnEachPage property

{button ,AL('H_LINENUMBEROPTIONS_CLASS',0)} [See list of classes](#)

{button ,AL('H_RESETONEACHPAGE_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write)

Data Type

[Integer](#)

Syntax

resetoneachpagevalue = [objectreference].ResetOnEachPage

[objectreference].ResetOnEachPage = resetoneachpagevalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Word Pro: ResetWhen property

{button ,AL(^H_FOOTNOTENUMOPT_CLASS;H_ENDNOTEDIVISIONGROUPNUM_CLASS;H_ENDNOTEDIVISIONNUM_CLASS;H_ENDNOTEDOCNUM_CLASS;H_FOOTNOTENUMBERING_CLASS',0)} [See list of classes](#)

{button ,AL(^H_RESETWHEN_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Resets the numbering for endnotes and footnotes.

Data Type

Data type is [Variant](#), which allows the value of this property to be one of the constants listed below or its numeric equivalent (in parentheses).

Syntax

resetwhenvalue = [objectreference].ResetWhen

[objectreference].ResetWhen = resetwhenvalue

Legal values

<u>Value</u>	<u>Effect</u>
\$LwpResetOptionEachDivision (1710)	Increases endnote numbers throughout a division and resets with the first endnote in the next division.
\$LwpResetOptionEachDivisiongroup (1711)	Increases endnote numbers throughout a division group and resets with first endnote in the next division group.
\$LwpResetOptionEachDoc (1708)	Increases endnote numbers each time you add a new endnote and continues increasing throughout the document.
\$LwpResetOptionEachPage (1709)	Increases a footnote number on a page and resets with the first footnote on the next page. Only resets footnote numbers on a page. Cannot be used to reset page numbers.

Usage

Equivalent to choosing Create - Footnote/Endnote, clicking Options, and selecting an option from the "Reset footnote numbers on each" box.

Word Pro: RestartStyleName property

{button ,AL('H_SILVERBULLET_CLASS',0)} [See list of classes](#)

{button ,AL('H_RESTARTSTYLENAME_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write)

Data Type

[String](#)

Syntax

restartstylevalue = [objectreference].RestartStyleName

[objectreference].RestartStyleName = restartstylevalue

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Word Pro: Rest property

{button ,AL('H_INDENT_CLASS;H_RELATIVEINDENT_CLASS',0)} [See list of classes](#)

{button ,AL('H_REST_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) The amount to indent all but the first line of a paragraph.

Data Type

Long

Syntax

restvalue = [objectreference].Rest

[objectreference].Rest = restvalue

Legal values

Data type is Long, but the unit of measurement used for this property is Twips. There are 1440 Twips per inch.

Usage

Equivalent to clicking the Alignment tab of the InfoBox, clicking Options, and setting the "Rest of paragraph" option in the Indent Options dialog box.

Word Pro: Result property

{button ,AL('H_POWERFIELD_CLASS',0)} [See list of classes](#)

(Read-write)

Data Type

String

Syntax

resultvalue = [objectreference].Result

[objectreference].Result = resultvalue

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Word Pro: ResumePausedMacro property

{button ,AL(^H_MACRO_CLASS',0)} [See list of classes](#)

(Write-only)

Data Type

Integer

Syntax

[objectreference].ResumePausedMacro = resumepausedmacrovalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Word Pro: RetainNameOfImportedFile property

{button ,AL('H_USERINTERFACEPREFS_CLASS',0)} [See list of classes](#)

{button ,AL('H_RETAINNAMEOFIMPORTEDFILE_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Sets or returns whether Word Pro uses existing file names of imported documents.

Data Type

[Integer](#)

Syntax

retainnameofimportedfilevalue = [objectreference].RetainNameOfImportedFile

[objectreference].RetainNameOfImportedFile = retainnameofimportedfilevalue

Legal values

Data type is Integer. The legal values for this property are -1 and 0 but you may use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Equivalent to the "Retain name of imported files" option on the Locations panel of the Word Pro Preferences dialog box. If this property is set to True (-1), Word Pro uses the existing file names of imported documents. If set to False (0), Word Pro does not assign a name to imported documents.

When Word Pro is installed, this property is set to False.

Word Pro: ReturnExits property

{button ,AL(^H_CLICKHERE_CLASS;',0)} [See list of classes](#)

(Read-write) Determines if pressing ENTER in a Click Here Block causes a return to be placed in the block, or if the return exits and moves on to the next block in the tab order.

Data Type

Integer.

Syntax

returnexitsvalue = [objectreference].ReturnExits

[objectreference].ReturnExits = returnexitsvalue

Legal values

The legal values for this property are:

<u>Value</u>	<u>Effect</u>
-1	True, don't allow returns in block, exit instead.
0	False, allow returns in block.

Usage

Equivalent to choosing Create - Click Here Block, clicking the Options tab and selecting "Allow returns to be inserted as data in this block." This property is similar to the TabExits property.

Word Pro: ReviewVersions property

{button ,AL('H_APPLICATIONWINDOW_CLASS',0)} [See list of classes](#)

(Read-only) An instance of the ReviewVersions class which allows you to review versions and create new versions.

Data Type

[ReviewVersions](#)

Syntax

reviewversionsvalue = [objectreference].ReviewVersions

Legal values

Always contains an instance of the ReviewVersions class.

Usage

Use this property to compare different files and versions when no document is open.

Word Pro: RevisionDisplay property

{button ,AL(^H_PREFERENCES_CLASS',0)} [See list of classes](#)

(Read-only)

Data Type

[RevisionDisplay](#)

Syntax

revisiondisplayvalue = [objectreference].RevisionDisplay

Legal values

Always contains an instance of the RevisionDisplay class.

Usage

Word Pro: RevisionMarkMode property

{button ,AL('H_DIVISION_CLASS;H_TEXTDOCUMENT_CLASS',0)} [See list of classes](#)

{button ,AL('H_REVISIONMARKMODE_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Determines whether Revision Marking is on (1) or off (0).

Data Type

[Integer](#)

Syntax

revisionmarkmodevalue = [objectreference].RevisionMarkMode

[objectreference].RevisionMarkMode = revisionmarkmodevalue

Legal values

The data type for this property is Integer. The legal values for this property are 1 and 0.

Usage

Word Pro: RevisionMark property

{button ,AL(^H_CLICKHERE_CLASS;H_TEXTMARKER_CLASS;H_TEXT_CLASS',0)} [See list of classes](#)

(Read-only)

Data Type

[Revision](#)

Syntax

revisionmarkvalue = [objectreference].RevisionMark

Legal values

Always contains an instance of the Revision class.

Usage

Word Pro: RevisionType property

{button ,AL(^H_LAYOUT_CLASS;H_CELLGROUPLAYOUT_CLASS;H_CELLLAYOUT_CLASS;H_CONNECTEDLAYOUT_CLASS;H_COLUMNGROUPLAYOUT_CLASS;H_FOOTERLAYOUT_CLASS;H_FRAMELAYOUT_CLASS;H_FRAMEGROUPLAYOUT_CLASS;H_DROP_CAP_LAYOUT_CLASS;H_GROUPLAYOUT_CLASS;H_HEADERLAYOUT_CLASS;H_NOTELAYOUT_CLASS;H_PAGELAYOUT_CLASS;H_ROWGROUPLAYOUT_CLASS;H_ROWLAYOUT_CLASS;H_RUBY_LAYOUT_CLASS;H_SUPERTABLEGROUPLAYOUT_CLASS;H_SUPERTABLELAYOUT_CLASS;H_TABLELAYOUT_CLASS;H_ENDNOTELAYOUT_CLASS;H_FOOTNOTELAYOUT_CLASS;H_TABLEHEADINGLAYOUT_CLASS;H_TOCSUPERTABLELAYOUT_CLASS;H_REVISION_CLASS',0)} [See list of classes](#)

(Read-only) Indicates whether or not an object was inserted or deleted while in revision marking mode.

Data Type

Data type is [Integer](#), which allows the value of this property to be one of the constants listed below or its numeric equivalent (in parentheses).

Syntax

revisiontypevalue = [objectreference].RevisionType

Legal values

<u>Value</u>	<u>Effect</u>
\$LwpRevisionTypeNone (1718)	No revisions made to the object.
\$LwpRevisionTypeInsert (1719)	Inserts an object.
\$LwpRevisionTypeDelete (1720)	Deletes an object.
\$LwpRevisionTypeDontcare (1721)	Revision information for the specified object is irrelevant.

Usage

Word Pro: RevMarkCharacter property

{button ,AL('H_REVISIONDISPLAY_CLASS',0)} [See list of classes](#)

(Read-write)

Data Type

Integer

Syntax

revmarkcharactervalue = [objectreference].RevMarkCharacter

[objectreference].RevMarkCharacter = revmarkcharactervalue

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Word Pro: RevMarkPosition property

{button ,AL('H_REVISIONDISPLAY_CLASS',0)} [See list of classes](#)

(Read-write)

Data Type

Variant (Enumerated)

MarkPosition

Syntax

revmarkpositionvalue = [objectreference].RevMarkPosition

[objectreference].RevMarkPosition = revmarkpositionvalue

Legal values

\$LwpMarkPositionBothSides (585)

\$LwpMarkPositionLeft (583)

\$LwpMarkPositionRight (584)

Usage

Word Pro: RevMarkType property

{button ,AL('H_REVISIONDISPLAY_CLASS',0)} [See list of classes](#)

(Read-write)

Data Type

Variant (Enumerated)

MarkType

Syntax

revmarktypevalue = [objectreference].RevMarkType

[objectreference].RevMarkType = revmarktypevalue

Legal values

\$LwpMarkTypeBars (587)

\$LwpMarkTypeChar (588)

\$LwpMarkTypeNone (586)

Usage

Word Pro: RightAlign property

{button ,AL(^H_BULLET_CLASS',0)} [See list of classes](#)

(Read-write)

Data Type

Variant (Enumerated)

CommandState

Syntax

rightalignvalue = [objectreference].RightAlign

[objectreference].RightAlign = rightalignvalue

Legal values

\$LwpCommandStateOff (151)

\$LwpCommandStateOn (152)

\$LwpCommandStateStyle (153)

Usage

Word Pro: RightBorder property

{button ,AL(^H_BORDERLINES_CLASS;H_GUTTER_CLASS;H_BETWEENLINES_CLASS',0)} [See list of classes](#)

(Read-only) Allows you to access an object's right border object.

Data Type

[Border](#)

Syntax

rightbordervalue = [objectreference].RightBorder

Legal values

Always contains an instance of the Border class.

Usage

You can also use the AllBorders property to simultaneously access an object's BottomBorder, LeftBorder, RightBorder, and TopBorder objects.

Word Pro: RightExternalMargin property

{button ,AL('H_LAYOUT_CLASS;H_CELLGROUPLAYOUT_CLASS;H_CELLLAYOUT_CLASS;H_CONNECTEDLAYOUT_CLASS;H_COLUMNGROUPLAYOUT_CLASS;H_FOOTERLAYOUT_CLASS;H_FRAMELAYOUT_CLASS;H_FRAMEGROUPLAYOUT_CLASS;H_DROP_CAP_LAYOUT_CLASS;H_GROUPLAYOUT_CLASS;H_HEADERLAYOUT_CLASS;H_NOTELAYOUT_CLASS;H_PAGELAYOUT_CLASS;H_ROWGROUPLAYOUT_CLASS;H_ROWLAYOUT_CLASS;H_RUBY_LAYOUT_CLASS;H_SUPERTABLEGROUPLAYOUT_CLASS;H_SUPERTABLELAYOUT_CLASS;H_TABLELAYOUT_CLASS;H_ENDNOTELAYOUT_CLASS;H_FOOTNOTELAYOUT_CLASS;H_TABLEHEADINGLAYOUT_CLASS;H_TOCSUPERTABLELAYOUT_CLASS',0)} [See list of classes](#)

{button ,AL('H_RIGHTEXTERNALMARGIN_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Allows you to set the amount of margin space that is present to the right of a layout object.

Data Type

[Long](#)

Syntax

rightexternalmarginvalue = [objectreference].RightExternalMargin

[objectreference].RightExternalMargin = rightexternalmarginvalue

Legal values

Data type is Long but the unit of measurement used for this property is [Twips](#). There are 1440 Twips per inch.

Usage

This property cannot be set individually for frame layout objects within Word Pro. It is combined with all external margin values in the "Padding around border" setting on the Size & Margins panel of the InfoBox.

yWord Pro: RightContextMenu property

{button ,AL('H_APPLICATIONWINDOW_CLASS',0)} [See list of classes](#)

(Read-only) Menus obtained when a user right clicks the mouse over a context sensitive area.

Data Type

[MenuItem](#)

Syntax

rightcontextmenuvalue = [objectreference].RightContextMenu

Legal values

Always contains an instance of the MenuItem class.

Usage

Use this class to access the right click menus when no document is open.

Word Pro: RightMousePropId property

{button ,AL(^H_GRAPHICOLEBJECT_CLASS;H_GRAPHIC_CLASS;H_OLEOBJECT_CLASS',0)} [See list of classes](#)

(Read-only)

Data Type

[Integer](#)

Syntax

rightmousepropidvalue = [objectreference].RightMousePropId

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Word Pro: RightMousePropText property

{button ,AL(^H_GRAPHICOLEBJECT_CLASS;H_GRAPHIC_CLASS;H_OLEOBJECT_CLASS',0)} [See list of classes](#)

(Read-only)

Data Type

[String](#)

Syntax

rightmouseproptextvalue = [objectreference].RightMousePropText

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Word Pro: RightPage property

```
{button ,AL('H_LAYOUT_CLASS;H_CELLGROUPLAYOUT_CLASS;H_CELLLAYOUT_CLASS;H_CONNECTEDLAY  
OUT_CLASS;H_COLUMNGROUPLAYOUT_CLASS;H_FOOTERLAYOUT_CLASS;H_FRAMELAYOUT_CLASS;H  
_FRAMEGROUPLAYOUT_CLASS;H_DROP_CAPLAYOUT_CLASS;H_GROUPLAYOUT_CLASS;H_HEADERLAY  
OUT_CLASS;H_NOTELAYOUT_CLASS;H_PAGELAYOUT_CLASS;H_ROWGROUPLAYOUT_CLASS;H_ROWLA  
YOUT_CLASS;H_RUBYLAYOUT_CLASS;H_SUPERTABLEGROUPLAYOUT_CLASS;H_SUPERTABLELAYOUT  
_CLASS;H_TABLELAYOUT_CLASS;H_ENDNOTELAYOUT_CLASS;H_FOOTNOTELAYOUT_CLASS;H_TABLE  
HEADINGLAYOUT_CLASS;H_TOCSUPERTABLELAYOUT_CLASS',0)} See list of classes
```

(Read-only) The right page of a complex layout. A complex layout contains separate layouts for left and right pages.

Data Type

[PageLayout](#)

Syntax

rightpagevalue = [objectreference].RightPage

Legal values

Always contains an instance of the PageLayout class.

Usage

Word Pro: Right property

{button ,AL('H_INDENT_CLASS;H_RELATIVEINDENT_CLASS',0)} [See list of classes](#)

{button ,AL('H_RIGHT_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) The amount to indent a paragraph from the right.

Data Type

Long

Syntax

rightvalue = [objectreference].Right

[objectreference].Right = rightvalue

Legal values

Data type is Long, but the unit of measurement used for this property is Twips. There are 1440 Twips per inch.

Usage

Equivalent to clicking the Alignment tab on the InfoBox, clicking Options, and setting the "All lines from right" option in the Indent Options dialog box.

Word Pro: RowLayouts property

{button ,AL(^H_BASETABLE_CLASS;H_FOOTNOTETABLE_CLASS;H_PARALLELCOLUMNS_CLASS;H_GLOSSARY_CLASS;H_TABLEHEADING_CLASS;H_TABLE_CLASS',0)} [See list of classes](#)

(Read-only) This object provides the names of all row layout objects within a table.

Data Type

[StringCollection](#)

Syntax

rowlayoutsvalue = [objectreference].RowLayouts

Legal values

Always contains an instance of the StringCollection class.

Usage

Row layout objects exist for rows which contain non-virgin cells. A virgin cell is a cell for which a layout object has not yet been created. A cell layout object is created when the layout settings or contents of a cell are modified.

Word Pro: Rows property

{button ,AL('H_FOUNDRY_CLASS',0)} [See list of classes](#)

(Read-only) An object created from the RowLayoutCollection class. This object provides access to RowLayout objects.

Data Type

[RowLayoutCollection](#)

Syntax

rowsvalue = [objectreference].Rows

Legal values

Always contains an instance of the RowLayoutCollection class.

Usage

When accessed through the Foundry property on a Division object, the collection object in this property provides access to all the RowLayout objects contained in that Division object.

When accessed through the AppFoundry property on the WPAApplication object, this collection object provides access to all the RowLayout objects contained in the Word Pro Clipboard.

When accessed through the TempFoundry property on the WPAApplication object, this collection object provides access to all the RowLayout objects placed in TempFoundry by WordPro or a script.

When accessed through the Foundry property on the WPAApplication object, this collection object provides access to all the RowLayout objects contained in the currently active Division object.

This property is not used in the Foundry property on the TextDocument object.

For more information about collection classes, see [Overview: Word Pro LotusScript Collection Classes](#).

Word Pro: RubyLayouts property

{button ,AL('H_FOUNDRY_CLASS',0)} [See list of classes](#)

(Read-only) An object created from the RubyLayoutCollection class. This object provides access to RubyLayout objects.

Data Type

[RubyLayoutCollection](#)

Syntax

rubylayoutsvalue = [objectreference].RubyLayouts

Legal values

Always contains an instance of the RubyLayoutCollection class.

Usage

When accessed through the Foundry property on a Division object, the collection object in this property provides access to all the RubyLayout objects contained in that Division object.

When accessed through the AppFoundry property on the WPAApplication object, this collection object provides access to all the RubyLayout objects contained in the Word Pro Clipboard.

When accessed through the TempFoundry property on the WPAApplication object, this collection object provides access to all the RubyLayout objects placed in TempFoundry by WordPro or a script.

When accessed through the Foundry property on the WPAApplication object, this collection object provides access to all the RubyLayout objects contained in the currently active Division object.

This property is not used in the Foundry property on the TextDocument object.

For more information about collection classes, see [Overview: Word Pro LotusScript Collection Classes](#).

Word Pro: RunMacroOnDocEvents property

{button ,AL('H_USERINTERFACEPREFS_CLASS',0)} [See list of classes](#)

{button ,AL('H_RUNMACROONDOCEVENTS_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Determines whether scripts run on the document open event.

Data Type

[Integer](#)

Syntax

runmacroondoceventsvalue = [objectreference].RunMacroOnDocEvents

[objectreference].RunMacroOnDocEvents = runmacroondoceventsvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values. Default is True (-1).

Usage

Equivalent to the "Document open scripts" option in the "Disable" box on the General panel of the Word Pro Preferences dialog box. If this property is set to True (-1), Word Pro allows scripts to run on the document open event. If set to False (0), Word Pro does not allow scripts to run on the document open event.

Word Pro: RunMacroOnLoad property

{button ,AL('H_USERINTERFACEPREFS_CLASS',0)} [See list of classes](#)

{button ,AL('H_RUNMACROONLOAD_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Determines whether application startup scripts run.

Data Type

[Integer](#)

Syntax

runmacroonloadvalue = [objectreference].RunMacroOnLoad

[objectreference].RunMacroOnLoad = runmacroonloadvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values. Default is True (-1).

Usage

Equivalent to the "Application startup scripts" option in the "Disable" box on the General panel of the Word Pro Preferences dialog box. If this property is set to True (-1), Word Pro allows scripts to run when the application loads. If set to False (0), Word Pro does not allow scripts to run when the application loads.

Word Pro: RunOnCloseDoc property

{button ,AL(^H_AUTORUNMACRO_CLASS',0)} [See list of classes](#)

(Read-write)

Data Type

Integer

Syntax

runoncloseddocvalue = [objectreference].RunOnCloseDoc

[objectreference].RunOnCloseDoc = runoncloseddocvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Word Pro: RunOnNewDoc property

{button ,AL(^H_AUTORUNMACRO_CLASS',0)} [See list of classes](#)

(Read-write)

Data Type

Integer

Syntax

runonnewdocvalue = [objectreference].RunOnNewDoc

[objectreference].RunOnNewDoc = runonnewdocvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Word Pro: RunOnOpenDoc property

{button ,AL('H_AUTORUNMACRO_CLASS',0)} [See list of classes](#)

(Read-write)

Data Type

Integer

Syntax

runonopenDocvalue = [objectreference].RunOnOpenDoc

[objectreference].RunOnOpenDoc = runonopenDocvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Word Pro: SaveDivisions property

{button ,AL('H_HTMLOPTIONS_CLASS',0)} [See list of classes](#)

{button ,AL('H_SAVEDIVISIONS_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write)

This language element has not yet been defined. You can check for a newer Help file on the Lotus web site.

Data Type

Integer.

Syntax

savedivisionsvalue = [objectreference].SaveDivisions

[objectreference].SaveDivisions = savedivisionsvalue

Legal values

The legal values for this property are -1 and 0 but you may use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Equivalent to choosing File - Internet - HTML Export Assistant and, on the Content panel, selecting "Save divisions as separate files."

Word Pro: Saved property

{button ,AL(^H_DOCUMENT_CLASS;H_TEXTDOCUMENT_CLASS',0)} [See list of classes](#)

(Read-only)

Data Type

Integer

Syntax

savdvalue = [objectreference].Saved

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Word Pro: SavePath property

{button ,AL('H_HTMLOPTIONS_CLASS',0)} [See list of classes](#)

{button ,AL('H_SAVEPATH_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write)

This language element has not yet been defined. You can check for a newer Help file on the Lotus web site.

Data Type

Integer

Syntax

savepathvalue = [objectreference].SavePath

[objectreference].SavePath = savepathvalue

Legal values

Integer. The values are:

<u>Value</u>	<u>Effect</u>
0	Same as document.
1	Single directory for all graphics.
2	Different directory for each graphic.

Usage

Equivalent to choosing File - Internet - HTML Export Assistant and, on the Preview & Save panel, selecting an option in "Location of files when saving."

Word Pro: SaveSections property

{button ,AL('H_HTMLOPTIONS_CLASS',0)} [See list of classes](#)

{button ,AL('H_SAVESECTIONS_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write)

This language element has not yet been defined. You can check for a newer Help file on the Lotus web site.

Data Type

Integer.

Syntax

savesectionsvalue = [objectreference].SaveSections

[objectreference].SaveSections = savesectionsvalue

Legal values

The legal values for this property are -1 and 0 but you may use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Equivalent to choosing File - Internet - HTML Export Assistant and, on the Content panel, choosing "Save sections as separate files."

Word Pro: SaveSnapshot property

{button ,AL('H_PREFERENCES_CLASS',0)} [See list of classes](#)

{button ,AL('H_SAVESNAPSHOT_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Determines whether or not Word Pro saves a copy of any images in the document.

Data Type

The data type for this property is [Variant](#) which allows the value of this property to be one of the constants listed below or its numeric equivalent (in parentheses).

Syntax

savesnapshotvalue = [objectreference].SaveSnapShot

[objectreference].SaveSnapShot = savesnapshotvalue

Legal values

<u>Value</u>	<u>Effect</u>
\$LwpSnapShotNosave (1768)	Word Pro will not save a snapshot of the images.
\$LwpSnapShotSave (1769)	Word Pro will save a snapshot of the images.

Usage

Saving a copy of images in the document allows Word Pro to display them much faster.

This property is equivalent to the "Fast Graphic Display" option in the Word Pro Preferences dialog box.

Word Pro: ScaleHeight property

{button ,AL(`H_LAYOUT_CLASS;H_CELLGROUPLAYOUT_CLASS;H_CELLLAYOUT_CLASS;H_CONNECTEDLAY
OUT_CLASS;H_COLUMNGROUPLAYOUT_CLASS;H_FOOTERLAYOUT_CLASS;H_FRAMELAYOUT_CLASS;H
_FRAMEGROUPLAYOUT_CLASS;H_DROP_CAPLAYOUT_CLASS;H_GROUPLAYOUT_CLASS;H_HEADERLAY
OUT_CLASS;H_NOTELAYOUT_CLASS;H_PAGELAYOUT_CLASS;H_ROWGROUPLAYOUT_CLASS;H_ROWLA
YOUT_CLASS;H_RUBY_LAYOUT_CLASS;H_SUPERTABLEGROUPLAYOUT_CLASS;H_SUPERTABLELAYOUT
_CLASS;H_TABLELAYOUT_CLASS;H_ENDNOTELAYOUT_CLASS;H_FOOTNOTELAYOUT_CLASS;H_TABLE
HEADINGLAYOUT_CLASS;H_TOCSUPERTABLELAYOUT_CLASS',0)} [See list of classes](#)

{button ,AL(`H_SCALEHEIGHT_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) The height of a graphic object that uses custom graphic scaling.

Data Type

Long

Syntax

scaleheightvalue = [objectreference].ScaleHeight

[objectreference].ScaleHeight = scaleheightvalue

Legal values

Data type is Long but the unit of measurement used for this property is Twips. There are 1440 Twips per inch.

Usage

A graphic object is sized using the ScaleHeight property if the layout object uses custom graphic scaling. You can specify that a layout object use custom graphic scaling by setting the ScaleMode property to \$LwpScaleTypeCustom.

{button ,AL(`H_MAINTAINASPECTRATIO_PROPERTY_MEMDEF;H_SCALEWIDTH_PROPERTY_MEMDEF;H_SC
ALEPERCENTAGE_PROPERTY_MEMDEF;H_SCALEMODE_PROPERTY_MEMDEF',0)} [See related topics](#)

Word Pro: ScaleMode property

```
{button ,AL(^H_JOIN_CLASS;H_LAYOUT_CLASS;H_CELLGROUPLAYOUT_CLASS;H_CELLAYOUT_CLASS;H_CONNECTEDLAYOUT_CLASS;H_COLUMNGROUPLAYOUT_CLASS;H_FOOTERLAYOUT_CLASS;H_FRAMELAYO  
OUT_CLASS;H_FRAMEGROUPLAYOUT_CLASS;H_DROPPLAYOUT_CLASS;H_GROUPLAYOUT_CLASS;  
H_HEADERLAYOUT_CLASS;H_NOTELAYOUT_CLASS;H_PAGELAYOUT_CLASS;H_ROWGROUPLAYOUT_C  
LASS;H_ROWLAYOUT_CLASS;H_RUBYLAYOUT_CLASS;H_SUPERTABLEGROUPLAYOUT_CLASS;H_SUPE  
RTABLELAYOUT_CLASS;H_TABLELAYOUT_CLASS;H_ENDNOTELAYOUT_CLASS;H_FOOTNOTELAYOUT_C  
LASS;H_TABLEHEADINGLAYOUT_CLASS;H_TOCSUPERTABLELAYOUT_CLASS',0)} See list of classes
```

```
{button ,AL(^H_SCALEMODE_PROPERTY_EXSCRIPT',1)} See example
```

(Read-write)

[JoinScaleType]

Determines how the width and height of a join is calculated.

[Layout]

Determines how the graphic content of a layout object is scaled.

Data Type

Data type is [Variant](#), which allows the value of this property to be one of the constants listed below or its numeric equivalent (in parentheses).

Syntax

scalemodevalue = [objectreference].ScaleMode

[objectreference].ScaleMode = scalemodevalue

Legal values

[JoinScaleType]

See the explanations for scaling (scaleable) and no scaling (fixed) in the Usage section below.

<u>Value</u>	<u>Effect</u>
\$LwpJoinScaleMatchborder (2385)	See Usage.
\$LwpJoinScaleTypeNoScaling (428)	See Usage.
\$LwpJoinScaleTypeScaling (427)	See Usage.

[Layout]

<u>Value</u>	<u>Effect</u>
\$LwpScaleTypeCustom (1725)	Custom rectangle (ScaleWidth and ScaleHeight) (checks MaintainAspectRatio).
\$LwpScaleTypeFitInFrame (1723)	Fits into the frame (checks MaintainAspectRatio).
\$LwpScaleTypeOriginalSize (1722)	The graphic is not scaled.
\$LwpScaleTypePercentage (1724)	Scaled by percentage (see ScalePercentage).

Usage

[JoinScaleType]

A join object can be scaleable or fixed.

Scaleable join objects

You can set the ScaleMode property of a scaleable join object to scaling or no scaling. Setting this property to scaling causes the join object's width and height to be a function of the layout object's height and width, and also the Percentage property. (For more information, see the Percentage property.)

Setting this property to no scaling causes the width and height of a join object to be the same as the width and height properties of the join object. The width and height of the join object does not change as the layout object's width and height changes.

The width and height of a scaleable join with a no scaling setting are fixed. However, you can change the width and height properties of a join object.

Fixed join objects

A fixed join object has predefined Word Pro width and height properties. Therefore, you cannot change its width or height.

[Layout]

Equivalent to the "Graphic scaling" settings on the Misc panel of the InfoBox for certain layout objects.

```
{button ,AL(`H_MAINTAINASPECTRATIO_PROPERTY_MEMDEF;H_SCALEWIDTH_PROPERTY_MEMDEF;H_SCALEPERCENTAGE_PROPERTY_MEMDEF;H_SCALEHEIGHT_PROPERTY_MEMDEF',0)} See related topics
```

Word Pro: ScalePercentage property

```
{button ,AL('H_LAYOUT_CLASS;H_CELLGROUPLAYOUT_CLASS;H_CELLLAYOUT_CLASS;H_CONNECTEDLAY  
OUT_CLASS;H_COLUMNGROUPLAYOUT_CLASS;H_FOOTERLAYOUT_CLASS;H_FRAMELAYOUT_CLASS;H  
_FRAMEGROUPLAYOUT_CLASS;H_DROP_CAPLAYOUT_CLASS;H_GROUPLAYOUT_CLASS;H_HEADERLAY  
OUT_CLASS;H_NOTELAYOUT_CLASS;H_PAGELAYOUT_CLASS;H_ROWGROUPLAYOUT_CLASS;H_ROWLA  
YOUT_CLASS;H_RUBYLAYOUT_CLASS;H_SUPERTABLEGROUPLAYOUT_CLASS;H_SUPERTABLELAYOUT  
_CLASS;H_TABLELAYOUT_CLASS;H_ENDNOTELAYOUT_CLASS;H_FOOTNOTELAYOUT_CLASS;H_TABLE  
HEADINGLAYOUT_CLASS;H_TOCSUPERTABLELAYOUT_CLASS',0)} See list of classes
```

```
{button ,AL('H_SCALEPERCENTAGE_PROPERTY_EXSCRIPT',1)} See example
```

(Read-write) Indicates, in tenths of percentage points, the size of a layout object's graphic content in relation to its original size.

Data Type

Long

Syntax

scalepercentagevalue = [objectreference].ScalePercentage

[objectreference].ScalePercentage = scalepercentagevalue

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

A graphic object is sized using the ScalePercentage property if the layout object uses a percentage graphic scaling setting. You can specify that a layout object use percentage graphic scaling by setting the [ScaleMode](#) property to \$LwpScaleTypePercentage.

```
{button ,AL('H_MAINTAINASPECTRATIO_PROPERTY_MEMDEF;H_SCALEWIDTH_PROPERTY_MEMDEF;H_SC  
ALEHEIGHT_PROPERTY_MEMDEF;H_SCALEMODE_PROPERTY_MEMDEF',0)} See related topics
```

Word Pro: ScaleWidth property

{button ,AL('H_LAYOUT_CLASS;H_CELLGROUPLAYOUT_CLASS;H_CELLLAYOUT_CLASS;H_CONNECTEDLAYOUT_CLASS;H_COLUMNGROUPLAYOUT_CLASS;H_FOOTERLAYOUT_CLASS;H_FRAMELAYOUT_CLASS;H_FRAMEGROUPLAYOUT_CLASS;H_DROP_CAP_LAYOUT_CLASS;H_GROUPLAYOUT_CLASS;H_HEADERLAYOUT_CLASS;H_NOTELAYOUT_CLASS;H_PAGELAYOUT_CLASS;H_ROWGROUPLAYOUT_CLASS;H_ROW_LAYOUT_CLASS;H_RUBY_LAYOUT_CLASS;H_SUPER_TABLE_GROUPLAYOUT_CLASS;H_SUPER_TABLE_LAYOUT_CLASS;H_TABLE_LAYOUT_CLASS;H_ENDNOTE_LAYOUT_CLASS;H_FOOTNOTE_LAYOUT_CLASS;H_TABLE_HEADING_LAYOUT_CLASS;H_TOCSUPER_TABLE_LAYOUT_CLASS',0)} [See list of classes](#)

{button ,AL('H_SCALEWIDTH_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) The width of a graphic object that uses custom graphic scaling.

Data Type

Long

Syntax

scalewidthvalue = [objectreference].ScaleWidth

[objectreference].ScaleWidth = scalewidthvalue

Legal values

Data type is Long but the unit of measurement used for this property is Twips. There are 1440 Twips per inch.

Usage

A graphic object is sized using the ScaleWidth property if the layout object uses custom graphic scaling. You can specify that a layout object use custom graphic scaling by setting the ScaleMode property to \$LwpScaleTypeCustom.

{button ,AL('H_MAINTAINASPECTRATIO_PROPERTY_MEMDEF;H_SCALEHEIGHT_PROPERTY_MEMDEF;H_SCALEPERCENTAGE_PROPERTY_MEMDEF;H_SCALEMODE_PROPERTY_MEMDEF',0)} [See related topics](#)

Word Pro: ScreenPositionX property

{button ,AL('H_ICONBAR_CLASS',0)} [See list of classes](#)

{button ,AL('H_SCREENPOSITIONX_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-only) Indicates the X, or horizontal, position of an icon bar.

Data Type

[Long](#)

Syntax

screenpositionxvalue = [objectreference].ScreenPositionX

Legal values

Data type is Long but the unit of measurement used for this property is [Twips](#). There are 1440 Twips per inch.

Usage

Gives the coordinate of an icon bar's horizontal position in the workspace.

Word Pro: ScreenPositionY property

{button ,AL(^H_ICONBAR_CLASS',0)} [See list of classes](#)

{button ,AL(^H_SCREENPOSITIONY_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-only) Indicates (if displayed) the vertical position of an icon bar object.

Data Type

[Long](#)

Syntax

screenpositionyvalue = [objectreference].ScreenPositionY

Legal values

Data type is Long but the unit of measurement used for this property is [Twips](#). There are 1440 Twips per inch.

Usage

Gives the coordinate of the icon bar's vertical position on the workspace.

Word Pro: SearchAttributes property

{button ,AL(^H_FINDANDREPLACE_CLASS',0)} [See list of classes](#)

(Read-only)

Data Type

[Attributes](#)

Syntax

searchattributesvalue = [objectreference].SearchAttributes

Legal values

Always contains an instance of the Attributes class.

Usage

Word Pro: SearchLanguage property

{button ,AL(^H_FINDANDREPLACE_CLASS',0)} [See list of classes](#)

(Read-only)

Data Type

[Language](#)

Syntax

searchlanguagevalue = [objectreference].SearchLanguage

Legal values

Always contains an instance of the Language class.

Usage

Word Pro: SectionName property

```
{button ,AL('H_BASECONTAINER_CLASS;H_CELLCONTAINER_CLASS;H_FRAMECONTAINER_CLASS;H_NOTECONTAINER_CLASS;H_RUBYCONTAINER_CLASS;H_DROPDOWNCONTAINER_CLASS;H_PAGECONTAINER_CLASS;H_SUBPAGECONTAINER_CLASS;H_SUPERPAGECONTAINER_CLASS;H_ROWCONTAINER_CLASS;H_SUPERTABLECONTAINER_CLASS;H_TABLECONTAINER_CLASS;H_PARALLELCOLSCONTAINER_CLASS;H_TABLEONLYCONT_CLASS;H_CLICKHERE_CLASS;H_TEXTMARKER_CLASS;H_TEXT_CLASS;H_TOCSUPTABLELAYOUT_CLASS',0)} See list of classes
```

```
{button ,AL('H_SECTIONNAME_PROPERTY_EXSCRIPT',1)} See example
```

(Read-only) The internal name of the current section.

Data Type

String

Syntax

sectionnamevalue = [objectreference].SectionName

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

If the insertion point is not located within the boundaries of a named section, this property returns a null string ("").

```
{button ,AL('H_USERNAME_PROPERTY_MEMDEF;H_SECTIONUSERNAME_PROPERTY_MEMDEF',0)} See related topics
```

Word Pro: Sections property

{button ,AL('H_FOUNDRY_CLASS',0)} [See list of classes](#)

(Read-only) An object created from the SectionCollection class. This object provides access to Section objects.

Data Type

[SectionCollection](#)

Syntax

sectionsvalue = [objectreference].Sections

Legal values

Always contains an instance of the SectionCollection class.

Usage

When accessed through the Foundry property on a Division object, the collection object in this property provides access to all the Section objects contained in that Division object.

When accessed through the AppFoundry property on the WPApplication object, this collection object provides access to all the Section objects contained in the Word Pro Clipboard.

When accessed through the TempFoundry property on the WPApplication object, this collection object provides access to all the Section objects placed in TempFoundry by WordPro or a script.

When accessed through the Foundry property on the WPApplication object, this collection object provides access to all the Section objects contained in the currently active Division object.

This property is not used in the Foundry property on the TextDocument object.

For more information about collection classes, see [Overview: Word Pro LotusScript Collection Classes](#).

Word Pro: SectionTabs property

{button ,AL('H_APPLICATIONWINDOW_CLASS',0)} [See list of classes](#)

(Read-only) An instance of the SectionTabs class which manipulates divider tabs.

Data Type

[SectionTabs](#)

Syntax

sectiontabsvalue = [objectreference].SectionTabs

Legal values

Always contains an instance of the SectionTabs class.

Usage

Use this property to manipulate divider tabs when a document is not open. For example, turn the divider tabs on or off before you open a document.

Word Pro: SectionUserName property

{button ,AL('H_CLICKHERE_CLASS;H_TEXTMARKER_CLASS;H_TEXT_CLASS',0)} [See list of classes](#)

{button ,AL('H_SECTIONUSERNAME_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-only)

This language element has not yet been defined. You can check for a newer Help file on the Lotus web site.

Data Type

String

Syntax

sectionusernamevalue = [objectreference].SectionUserName

Legal values

This property is read-only. The value of this property cannot be set by a script.

Usage

{button ,AL('H_USERNAME_PROPERTY_MEMDEF;H_SECTIONNAME_PROPERTY_MEMDEF',0)} [See related topics](#)

Word Pro: Section property

{button ,AL(^H_GRAPHICOLEBJECT_CLASS;H_GRAPHIC_CLASS;H_OLEOBJECT_CLASS',0)} [See list of classes](#)

(Read-only)

Data Type

[String](#)

Syntax

sectionvalue = [objectreference].Section

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Word Pro: SelectedPages property

{button ,AL('H_PRINTSETTINGS_CLASS',0)} [See list of classes](#)

{button ,AL('H_SELECTEDPAGES_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Allows you to specify the page numbers that you want to print in a document.

Data Type

String

Syntax

selectedpagesvalue = [objectreference].SelectedPages

[objectreference].SelectedPages = selectedpagesvalue

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

You must place quotes around the page numbers or names to indicate a string expression.

Usage

You cannot use this property unless the \$LtsPrintRangeSelectedPages value is set to the PrintRange property. For information, see the [PrintRange property](#).

From the Select Pages dialog box, you can print a list of pages or whole divisions. For example, you can print pages 1-4, 7, 13-16, and 20. Or you can print divisions, such as TOC and Chapter 4. This property is equivalent to the number (s) or name (s) that you can type in List the pages box in the Select Pages dialog box. To get to the Select Pages dialog box in Word Pro, choose File - Print, select "Selected pages only, and "click "Select Pages"

Word Pro: SelectionBorderColor1 property

{button ,AL(^H_APPVIEWPREFS_CLASS',0)} [See list of classes](#)

(Read-only) Word Pro places bars and handles around any selected frame. The bars and handles are made up of three colors that give the frame a shaded effect. SelectionBorderColor1 stores the outermost color.

Data Type

[Color](#)

Syntax

selectionbordercolor1value = [objectreference].SelectionBorderColor1

Legal values

Always contains an instance of the Color class.

Usage

Word Pro: SelectionBorderColor2 property

{button ,AL(^H_APPVIEWPREFS_CLASS',0)} [See list of classes](#)

(Read-only) Word Pro places bars and handles around any selected frame. The bars and handles are made up of three colors that give the frame a shaded effect. SelectionBorderColor2 stores the middle color.

Data Type

[Color](#)

Syntax

selectionbordercolor2value = [objectreference].SelectionBorderColor2

Legal values

Always contains an instance of the Color class.

Usage

The SelectionBorderColor2 property also controls the color of the frame border when the insertion point is inside a frame.

Word Pro: SelectionBorderColor3 property

{button ,AL(^H_APPVIEWPREFS_CLASS',0)} [See list of classes](#)

(Read-only) Word Pro places bars and handles around any selected frame. The bars and handles are made up of three colors that give the frame a shaded effect. SelectionBorderColor3 stores the innermost color.

Data Type

[Color](#)

Syntax

selectionbordercolor3value = [objectreference].SelectionBorderColor3

Legal values

Always contains an instance of the Color class.

Usage

Word Pro: SelectionHidden property

{button ,AL(^H_CLICKHERE_CLASS;H_TEXTMARKER_CLASS;H_TEXT_CLASS',0)} [See list of classes](#)

(Read-only) Indicates whether or not the selected text is marked as hidden.

Data Type

Integer

Syntax

selectionhiddenvalue = [objectreference].SelectionHidden

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

If the text at either end of the selection is marked as hidden, this property is True. If text in the middle of the selection is marked as hidden, but the text at both ends is not marked as hidden, this property returns a value of False.

Word Pro: SelectionType property

{button ,AL(^H_DIVISION_CLASS;H_CLICKHERE_CLASS;H_TEXTMARKER_CLASS;H_BASSETABLE_CLASS;H_FOOTNOTETABLE_CLASS;H_PARALLELCOLUMNS_CLASS;H_GLOSSARY_CLASS;H_TABLEHEADING_CLASSES;H_TABLE_CLASS;H_TEXT_CLASS;H_TEXTDOCUMENT_CLASS',0)} [See list of classes](#)

(Read-only) Indicates what type of object is selected in the current text stream.

Data Type

Variant (Enumerated)

Syntax

selectiontypevalue = [objectreference].SelectionType

Legal values

<u>Value</u>	<u>Effect</u>
\$LwpSelectionTypeNone (1760)	Indicates that nothing is selected in the current text stream.
\$LwpSelectionTypeText (1761)	Indicates that text is selected in the current text stream.
\$LwpSelectionTypeCell (2376)	Indicates that a cell is selected in the current text stream.
\$LwpSelectionTypeFrame (2377)	Indicates that a frame is selected in the current text stream.

Usage

If more than one of these types of objects is in the selection, this property indicates which type of object is uppermost in the focus.

Word Pro: SelectTab property

{button ,AL(^H_RULER_CLASS;H_SETTABS_DIALOG_CLASS,0)} [See list of classes](#)

(Read-write)

[Ruler]

Represents the currently selected tab on the ruler. In addition, this property coordinates the tab setting between the Set Tab dialog box and the ruler.

[SetTabsDialog]

Represents the currently selected tab in the Set Tabs dialog box. In addition, this property coordinates the tab setting between the Set Tab dialog box and the ruler.

Data Type

[Integer](#)

Syntax

selecttabvalue = [objectreference].SelectTab

[objectreference].SelectTab = selecttabvalue

Legal values

The value for this property must be a positive [Integer](#).

Usage

[SetTabsDialog]

An internal Word Pro setting that coordinates the tab setting between the Set Tabs dialog box and the ruler.

Word Pro: SelectType property

```
{button ,AL('H_LAYOUT_CLASS;H_CELLGROUPLAYOUT_CLASS;H_CELLLAYOUT_CLASS;H_CONNECTEDLAYOUT_CLASS;H_COLUMNGROUPLAYOUT_CLASS;H_FOOTERLAYOUT_CLASS;H_FRAMELAYOUT_CLASS;H_FRAMEGROUPLAYOUT_CLASS;H_DROPDOWNLAYOUT_CLASS;H_GROUPLAYOUT_CLASS;H_HEADERLAYOUT_CLASS;H_NOTELAYOUT_CLASS;H_PAGELAYOUT_CLASS;H_ROWGROUPLAYOUT_CLASS;H_ROWLAYO  
UT_CLASS;H_RUBYLAYOUT_CLASS;H_SUPERTABLEGROUPLAYOUT_CLASS;H_SUPERTABLELAYOUT  
_CLASS;H_TABLELAYOUT_CLASS;H_ENDNOTELAYOUT_CLASS;H_FOOTNOTELAYOUT_CLASS;H_TABLE  
HEADINGLAYOUT_CLASS;H_TOCSUPERTABLELAYOUT_CLASS',0)} See list of classes
```

(Read-write) Allows you to control whether or not a specific type of layout is selected when the user enters that layout. For example, you can set a property to select the parent layout, the layout itself, or no layout.

Data Type

Data type is [Variant](#), which allows the value of this property to be one of the constants listed below or its numeric equivalent (in parentheses).

Syntax

selectypevalue = [objectreference].SelectType

[objectreference].SelectType = selectypevalue

Legal values

<u>Value</u>	<u>Effect</u>
\$LwpLayoutSelectNone (524)	No layout is selected when the layout is clicked; the currently selected layout or the default layout opens when the layout is clicked.
\$LwpLayoutSelect (525)	The layout itself is selected.
\$LwpLayoutSelectParent (2050)	The parent of the layout is selected when you click on one of the layouts that comprises a NoteLayout.

Usage

Word Pro: SequenceName property

{button ,AL('H_FRAMECAPTIONOPTIONS_CLASS',0)} [See list of classes](#)

{button ,AL('H_SEQUENCENAME_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write)

This language element has not yet been defined. You can check for a newer Help file on the Lotus web site.

Data Type

String.

Syntax

sequencenamevalue = [objectreference].SequenceName

[objectreference].SequenceName = sequencenamevalue

Legal values

String.

Usage

Word Pro: SequenceNumber property

{button ,AL('H_FRAMECAPTIONOPTIONS_CLASS',0)} [See list of classes](#)

{button ,AL('H_SEQUENCENUMBER_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write)

This language element has not yet been defined. You can check for a newer Help file on the Lotus web site.

Data Type

Integer.

Syntax

sequencenumbervalue = [objectreference].SequenceNumber

[objectreference].SequenceNumber = sequencenumbervalue

Legal values

Integer.

Usage

Word Pro: ServerFormat property

{button ,AL('H_GRAPHICOLEBJECT_CLASS;H_GRAPHIC_CLASS;H_OLEOBJECT_CLASS',0)} [See list of classes](#)

(Read-write)

Data Type

[String](#)

Syntax

serverformatvalue = [objectreference].ServerFormat

[objectreference].ServerFormat = serverformatvalue

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Word Pro: SetContextOfBar property

{button ,AL('H_ICONBAR_CLASS',0)} [See list of classes](#)

(Read-write) Allows you to query or specify the context in which an icon bar will be displayed when you are working in a specific context. Word Pro has several different contexts (always, in text, in a frame, in columns, in a table or table cell, in a drawing). You can specify in which of these contexts you want the bar displayed.

Data Type

Integer

Syntax

[objectreference].SetContextOfBar = setcontextofbarvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

You can set the display of a specific icon bar set. Equivalent to the value found in the "Bar can be displayed when context is" box in the SmartIcons Setup dialog box.

Word Pro: SetTabsDialog property

{button ,AL('H_APPLICATIONWINDOW_CLASS',0)} [See list of classes](#)

(Read-only) An instance of the SetTabsDialog class which deals with the dialog box used to set tabs in a document.

Data Type

[SetTabsDialog](#)

Syntax

settabsdialogvalue = [objectreference].SetTabsDialog

Legal values

Always contains an instance of the SetTabsDialog class.

Usage

Use this property to manipulate the Set Tabs dialog box in conjunction with Ruler objects. You can use this property to bring up the Set Tabs dialog box and select a tab.

Word Pro: Shape property

{button ,AL('cī½_BACKGROUND_CLASS;H_CHARTPLOTWALL_CLASS;H_CHARTDATAPOINTS_CLASS;H_CHARTLEGENDENTRIES_CLASS;H_CHARTPIETITLES_CLASS;H_CHARTMAJORGRIDLINES_CLASS;H_CHARTPIES_CLASS;H_CHARTPIES_CLASS;H_CHARTPIESLICEGROUPS_CLASS;H_CHARTPIESLICES_CLASS;H_CHARTPLOT_CLASS;H_CHARTSERIESCOLLECTION_CLASS;H_CHARTTABLESERIESCOLLECTION_CLASS;H_CHARTTEXTENTRIES_CLASS;H_CHARTTEXTLABELS_CLASS',0)} [See list of classes](#)

(Read-write)

Data Type

Integer

Syntax

[objectreference].Shape = shapevalue

shapevalue = [objectreference].Shape

Legal values

Usage

Word Pro: ShowBubbleHelp property

{button ,AL('H_USERINTERFACEPREFS_CLASS',0)} [See list of classes](#)

{button ,AL('H_SHOWBUBBLEHELP_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Determines whether bubble help is displayed for icons and InfoBox tabs.

Data Type

[Integer](#)

Syntax

showbubblehelpvalue = [objectreference].ShowBubbleHelp

[objectreference].ShowBubbleHelp = showbubblehelpvalue

Legal values

Data type is Integer. The legal values for this property are 1 and 0.

Usage

Equivalent to the "Bubble help" option in the "General usage" box on the Enable panel of the Word Pro Preferences dialog box. If this property is set to 1, Word Pro displays bubble help. If set to 0, Word Pro does not display bubble help.

Word Pro: ShowDivisionTabs property

{button ,AL('H_DOCCONTROL_CLASS',0)} [See list of classes](#)

{button ,AL('H_SHOWDIVISIONTABS_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Indicates whether or not all division tabs, including hidden division tabs, are shown in a document.

Data Type

[Integer](#)

Syntax

showdivisiontabsvalue = [objectreference].ShowDivisionTabs

[objectreference].ShowDivisionTabs = showdivisiontabsvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Word Pro can contain hidden tabs. For example, Word Pro always places scripts in a division, but the script division tab will not display unless you set the ShowDivisionTabs property.

Equivalent to choosing File - TeamSecurity and selecting "Display all division tabs in document" on the Other Protection panel.

{button ,AL('H_SHOWTABS_PROPERTY_MEMDEF',0)} [See related topics](#)

Word Pro: ShowExportWarningMessages property

{button ,AL('H_USERINTERFACEPREFS_CLASS','0)} [See list of classes](#)

{button ,AL('H_SHOWEXPORTWARNINGMESSAGES_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write)

This language element has not yet been defined. You can check for a newer Help file on the Lotus web site.

Data Type

Integer

Syntax

showexportwarningmessagesvalue = [objectreference].ShowExportWarningMessages

[objectreference].ShowExportWarningMessages = showexportwarningmessagesvalue

Legal values

The legal values for this property are -1 and 0 but you may use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Word Pro: ShowFileNew property

{button ,AL('H_USERINTERFACEPREFS_CLASS',0)} [See list of classes](#)

{button ,AL('H_SHOWFILENEW_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Determines whether the New Document dialog box displays when creating a new document.

Data Type

[Integer](#)

Syntax

showfilenewvalue = [objectreference].ShowFileNew

[objectreference].ShowFileNew = showfilenewvalue

Legal values

Data type is Integer.

Usage

Equivalent to the "Show File New dialog box" option on the General panel of the Word Pro Preferences dialog box. If this property is set to 1, Word Pro displays the File New dialog box. If set to 0, Word Pro does not display the File New dialog box.

Note This property affects new document creation only with the "Create a new document" SmartIcon. Choosing File - New will present the New Document dialog box regardless of this property value.

Word Pro: ShowGraphicPreview property

{button ,AL(^H_USERINTERFACEPREFS_CLASS,0)} [See list of classes](#)

(Read-write) Determines whether a graphic preview is displayed in the Import Picture dialog box.

Data Type

Integer

Syntax

showgraphicpreviewvalue = [objectreference].ShowGraphicPreview

[objectreference].ShowGraphicPreview = showgraphicpreviewvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values. Default is True (-1). However, this property always retains the last value selected by the user.

Usage

Equivalent to the "Preview" box on the Import Picture dialog box. If this property is set to True (-1), Word Pro displays the graphic preview. If set to False (0), Word Pro does not display the graphic preview.

Word Pro: ShowHeaderFooterBar property

{button ,AL('H_USERINTERFACEPREFS_CLASS','0')} [See list of classes](#)

{button ,AL('H_SHOWHEADERFOOTERBAR_PROPERTY_EXSCRIPT',1')} [See example](#)

(Read-write)

This language element has not yet been defined. You can check for a newer Help file on the Lotus web site.

Data Type

Integer

Syntax

showheaderfooterbarvalue = [objectreference].ShowHeaderFooterBar

[objectreference].ShowHeaderFooterBar = showheaderfooterbarvalue

Legal values

The legal values for this property are -1 and 0 but you may use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Word Pro: ShowHiddenText property

{button ,AL('H_DIVISIONOPTIONS_CLASS',0)} [See list of classes](#)

{button ,AL('H_SHOWHIDDENTEXT_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Specifies whether Word Pro will display hidden text (-1) or not display hidden text (0).

Data Type

[Integer](#)

Syntax

showhiddentextvalue = [objectreference].ShowHiddenText

[objectreference].ShowHiddenText = showhiddentextvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Equivalent to "Show hidden text" on the Other Protection panel of the TeamSecurity dialog box.

{button ,AL('H_HIDDENMODE_PROPERTY_MEMDEF',0)} [See related topics](#)

Word Pro: ShowInContext property

{button ,AL('H_ICONBAR_CLASS',0)} [See list of classes](#)

{button ,AL('H_SHOWINCONTEXT_PROPERTY_EXSCRIPT',1)} [See example](#)

(WriteOnly) Allows you to enable or disable the display of an icon bar object. Acts as an on/off switch and, if selected, displays a specific set of icons whenever you are working in a specific part of a document. If turned on, the bar will display in its specified context; if turned off, the bar will never display.

Data Type

[Integer](#)

Syntax

[objectreference].ShowInContext = showincontextvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Used when you want to see an icon bar set in a specific context. Equivalent to the option set in the "Bar is enabled to display during its context" box in the SmartIcons Setup dialog box. You can turn it on (displays a specific set whenever you are working in a specific context) or off (turns off the display).

For example, if you turn off the Comment Tools icon bar and then use View - Show/Hide - Comment Tools to redisplay this bar, Comment Tools would then be checked (to display) in the SmartIcons Setup dialog box.

Word Pro: ShowMailDisabled property

{button ,AL('H_USERINTERFACEPREFS_CLASS',0)} [See list of classes](#)

{button ,AL('H_SHOWMAILDISABLED_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Determines whether the mail notification icon is displayed.

Data Type

[Integer](#)

Syntax

showmaildisabledvalue = [objectreference].ShowMailDisabled

[objectreference].ShowMailDisabled = showmaildisabledvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values. Default is True (-1).

Usage

Equivalent to the "New mail indicator" option in the "Disable field" on the General panel of the Word Pro Preferences dialog box. If this property is set to True (-1), Word Pro does not display the new mail icon in the status bar. If set to False (0), Word Pro displays the new mail icon in the status bar.

Word Pro: FootnoteContSep class

The separator between continued footnotes.

Base Classes

BaseObject\FooterSepOpt

Derived Classes

None.

Contained by

FootnoteOptions in the FootnoteContSep Property

Usage

Consists of properties that define separator options for footnotes that continue across multiple pages. In Word Pro, this class is represented on the Separators panel of the Footnote and Endnote Options dialog box, when the "Separator line for" option is set to "Continued footnotes."

Word Pro: FootnoteLayoutCollection class

A collection of footnote layouts objects in the foundry of a specific division.

Base Classes

BaseObject\BaseCollection

Derived Classes

None.

Contained by

Foundry in the FootnoteLayouts Property

Usage

Use this collection to access any of the footnote layout objects in the foundry of a specific division.

Word Pro: FootnoteLayout class

The layout for a footnote object.

Base Classes

BaseObject\Layout\TableLayout\EndnoteLayout

Derived Classes

None.

Contained by

Usage

Word Pro: FootnoteNumbering class

Refers to the numbering order for footnote objects in a division.

Base Classes

BaseObject\FootnoteNumOpt

Derived Classes

None.

Contained by

FootnoteOptions in the FootnoteNumbering Property

Usage

Word Pro: FootnoteNumOpt class

The numbering options for a footnote in a division.

Base Classes

BaseObject

Derived Classes

[EndnoteDivisionGroupNum](#)

[EndnoteDivisionNum](#)

[EndnoteDocNum](#)

[FootnoteNumbering](#)

Contained by

Usage

Word Pro: FootnoteOptions class

Refers to the footnote options displayed in the Footnote dialog box.

Base Classes

BaseObject

Derived Classes

None.

Contained by

Division in the FootnoteOptions Property

TextDocument in the FootnoteOptions Property

Usage

Word Pro: FootnoteSeparator class

Refers to the line that separates the footnote from the body text in the document.

Base Classes

BaseObject\FooterSepOpt

Derived Classes

None.

Contained by

FootnoteOptions in the FootnoteSeparator Property

Usage

Consists of properties that define separator options for footnotes that are contained within a single page. In Word Pro, this class is represented on the Separators panel of the Footnote and Endnote Options dialog box, when the "Separator line for" option is set to "Regular footnotes."

Word Pro: FootnoteSepOpt class

Refers to the footnote separator options displayed on the Separators panel in the Footnote and Endnote Options dialog box.

Base Classes

BaseObject

Derived Classes

FootnoteContSep

FootnoteSeparator

Contained by**Usage**

Provides a basis for derived classes which are used to access footnote separator options for regular footnotes and continued footnotes. A footnote separator is the line that separates the footnote from the body text.

Word Pro: FootnoteTable class

A table that contains footnote objects.

Base Classes

BaseObject\Content\BaseTable

Derived Classes

None.

Contained by**Usage**

Word Pro: FormatCheckPref class

Refers to the format check options displayed in the Format Check Options dialog box, accessed from the Format Check bar.

Base Classes

BaseObject

Derived Classes

None.

Contained by

WPApplication in the FormatCheckPreferences Property

Usage

You can check your document for consistent use of spacing between sentences, correct bulleted lists, and the appearance of acronyms in a paragraph. The format check also replaces incorrect characters and common typing mistakes.

Word Pro: Foundry class

A place where Word Pro creates, stores, and provides access to other Word Pro objects. Word Pro maintains several Foundry objects and each one has a special use as described below under Usage. However, in most cases, you use the Division.Foundry object to access objects in a document and the WPAApplication.TempFoundry object to create, store, and move objects from one Division.Foundry to another.

Base Classes

BaseObject

Derived Classes

None.

Contained by

[Division](#) in the [Foundry](#) Property

[WPAApplication](#) in the [TempFoundry](#) Property

[WPAApplication](#) in the [Foundry](#) Property

[WPAApplication](#) in the [AppFoundry](#) Property

Usage

WPAApplication.AppFoundry

AppFoundry is a property on the WPAApplication object (always stored in the CurrentApplication variable). It contains a Foundry object which Word Pro uses as the Clipboard. This is the same Clipboard used when you copy or cut items in a Word Pro document. When you cut or copy, Word Pro takes all the objects from your selection and places them in their respective collection objects in the Foundry object, stored in the AppFoundry property.

For example, if you select some text and a table, and choose Edit - Copy, Word Pro places all the objects that comprise that text and table into their respective collection objects in AppFoundry. This means that all the Layout objects are stored in the corresponding layout collection objects. All CharacterStyle objects are stored in the CharacterStyleCollection object. All CellEngine objects are stored in the CellCollection object. Text objects are stored in the TextCollection object. When you choose Edit - Paste, all of these objects are reassembled in their original form and displayed in the document at the insertion point.

Because Word Pro uses the Foundry object in AppFoundry as its Clipboard, you must exercise caution when working with AppFoundry. Any objects you place in AppFoundry are included in the next Paste operation. Any objects you remove from AppFoundry are excluded from the next Paste operation, and may adversely affect your ability to paste from the Clipboard.

You can get an object from AppFoundry and store it in a variable using the following statement:

```
myobject = CurrentApplication.AppFoundry.collectionpropertyname(itemreference)
```

In this statement, *myobject* is the variable in which you want to store the object; CurrentApplication is a global variable that always contains the WPAApplication object; *collectionpropertyname* is the name of the property that contains the collection object where the object you want is stored; *itemreference* is the index that specifies the object you want. For more information about collection classes, see [Overview: Word Pro LotusScript Collection Classes](#).

Note While you may retrieve objects from AppFoundry, you should not use LotusScript to place objects in the AppFoundry collections. This can interfere with normal user operations, such as Cut and Copy. When creating and storing your own Word Pro objects, use the Foundry object in the TempFoundry property.

WPAApplication.TempFoundry

TempFoundry is a property on WPAApplication. TempFoundry contains a Foundry object which Word Pro uses to temporarily store objects that are part of a drag and drop operation. You can use TempFoundry in much the same way. You can use the collection objects in TempFoundry as a staging area for any Word Pro LotusScript objects you create and manipulate.

For example, when you want to move an object or objects from one document to another, you can store those objects temporarily in the TempFoundry collection objects. The TempFoundry property is always available, regardless of which document is active, so you always have access to the contents of its collections. This makes it an ideal place for temporarily storing items that you want to use or move.

Note You must [clear](#) TempFoundry after each use. Any objects left in any of TempFoundry's collections can reappear during Drag and Drop and other operations, and result in unpredictable behavior.

Division.Foundry

In addition to AppFoundry and TempFoundry, Word Pro maintains one Foundry object for each Division object. These Division foundries are stored in the Foundry property on each Division object. The Foundry object in a division provides access to all the objects in that division, including layouts, text, graphics, markers, tables, footnotes, and so on. As seen in the AppFoundry example above, you can access all the objects in a division through the appropriate collection in the Division foundry.

You can access objects in any division's Foundry property by going through the division collection object stored in the Divisions property on WPAApplication, as follows:

```
myobject = CurrentApplication.Divisions(DivisionName).Foundry.collectionpropertyname(itemreference)
```

In this example, *DivisionName* is the name of the division as seen in the Division tab; *collectionpropertyname* is the name of the Foundry object property in which the collection object you seek is stored; *itemreference* is a name or reference to a specific object in the collection.

WPAApplication.Foundry

The Foundry property in WordPro provides a shortcut to the currently active division's Foundry object. The Foundry object in WordPro.Foundry changes as the focus changes from one Division object to another.

For example, if you have a document with one division named Overview and another division named Summary, the contents of the Foundry property on WordPro change as you move the focus from Overview to Summary. While the focus is on the Overview division, this property contains the Foundry object for the Division object named Overview. When the focus changes to the Summary division, this property also changes to contain the Foundry object for the Division object named Summary.

To access a collection object in the currently active division's Foundry, you can use the following statement:

```
myobject = CurrentApplication.Foundry.collectionpropertyname(itemreference)
```

TextDocument.Foundry

The Foundry object stored in TextDocument.Foundry is not used by Word Pro. Do not use this Foundry object in your scripts.

Word Pro: FrameContainer class

The container object for frames. This object only exists for one frame at a time, and only when there is a frame with in the focus. When a FrameContainer object is present, it is stored in the Frame property on the WPApplication object.

Base Classes

BaseObject\BaseContainer

Derived Classes

[NoteContainer](#)

[RubyContainer](#)

Contained by

[WPApplication](#) in the [Frame](#) Property

Usage

The primary use for a FrameContainer object is to provide quick and easy access to the FrameLayout object for the currently active frame. A FrameContainer object always represents the frame that currently has the focus. Therefore, if you assign a FrameContainer object to a variable, you can use that variable to access the currently active frame. However, you must remember that the frame referenced by the variable changes as the focus moves from one frame to another. This is because the variable references the FrameContainer object, and the FrameContainer object always represents the frame that has the focus. If there is no frame with in the focus, there is no FrameContainer object. Therefore, a variable that stores a FrameContainer object has a null value, whenever the focus does not contain a frame. There is never more than one FrameContainer object at any time.

For more information about container objects, see [BaseContainer](#).

Word Pro: FrameGroupLayout class

The layout for a frame group.

Base Classes

BaseObject\Layout\FrameLayout

Derived Classes

None.

Contained by

Usage

When multiple frames are selected, the combined layout is a FrameGroupLayout object.

Word Pro: FrameLayoutCollection class

A collection of frame layout objects in the foundry of a specific division.

Base Classes

BaseObject\BaseCollection

Derived Classes

None.

Contained by

Foundry in the Frames Property

Foundry in the FrameStyles Property

Usage

Use this collection to access any of the frame layout objects in the foundry of a specific division.

Word Pro: FrameLayout class

A frame layout for a frame object. This class inherits most of its members from the Layout class.

Base Classes

BaseObject\Layout

Derived Classes

FrameGroupLayout

Contained by

Usage

The FrameLayout class provides you with a way to access and modify the format and appearance of FrameLayout objects within your document.

Since the FrameLayout class is derived from the Layout class, FrameLayout objects can be stored within properties of the Layout type. For example, the Layout property with in the FrameContainer class is of the Layout type. However, this property often stores objects of the FrameLayout type. The Layout property is implemented in this way so that objects of other derived layout class types can be stored there as well. The Layout property with in the FrameContainer class, for example, may also contain objects of the NoteLayout type.

FrameLayout objects within a division are stored together in a collection. You can use the collection to access all FrameLayout objects in the collection, or you can reference a particular FrameLayout object in the collection. For example, by using the FrameLayouts collection, you can modify each FrameLayout object in the collection to be of a certain height. For more information on how to work with collections, see [Overview: Word Pro LotusScript Collection Classes](#).

At many locations within your document, multiple layouts are available. For example, your cursor may be within a frame within a page. In this case, the frame and the page both have associated layout objects. These layout objects may be combined with other objects into related groups, known as containers. For more information on containers and their associated layouts, see the help topic titled [Word Pro: BaseContainer class](#).

Word Pro: GlossaryCollection class

A collection of glossary objects in the foundry of a specific division.

Base Classes

BaseObject\BaseCollection

Derived Classes

None.

Contained by

Foundry in the Glossarys Property

Usage

Use this collection to access any of the glossary objects in the foundry of a specific division.

Word Pro: Glossary class

A glossary object within a Word Pro glossary file.

Base Classes

BaseObject\Content\BaseTable\ParallelColumns

Derived Classes

None.

Contained by**Usage**

A glossary file stores frequently used text, tables, frames, and so on, which can be inserted into any document. These frequently used items are contained in a Glossary object that is stored in a Word Pro glossary file.

Word Pro: Grammar class

A grammatical proofing tool that analyzes a document for possible errors, and supplies suggestions and examples for incorrect sentences.

Base Classes

BaseObject

Derived Classes

None.

Contained by

UserInterfacePrefs in the GrammarOptions Property

Usage

You can proofread and edit a document for grammar, style, and mechanics. You can also display document and readability statistics for every document you check. Equivalent to choosing Edit - Check Grammar, and then using the default grammar options or customizing grammar options for proofing the document.

The Grammar class uses many Apply properties, some of which are only found in specific language versions of Word Pro.

Word Pro: GraphicCollection class

A collection of graphic objects in the foundry of a specific division.

Base Classes

BaseObject\BaseCollection

Derived Classes

None.

Contained by

Foundry in the Graphics Property

Usage

Use this collection to access any of the graphic objects in the foundry of a specific division.

Word Pro: GraphicOleObjectCollection class

A collection of graphic and OLE objects in the foundry of a specific division.

Note OLE is not supported under OS/2.

Base Classes

BaseObject\BaseCollection

Derived Classes

None.

Contained by**Usage**

Use this collection to access any of the graphic and OLE objects in the foundry of a specific division.

Word Pro: GraphicOleObject class

An OLE graphic object in a document; the virtual base class for a graphic and OLE object.

Note OLE is not supported under OS/2.

Base Classes

BaseObject\Content

Derived Classes

Graphic

OleObject

Contained by

WPApplication in the GraphicOleObject Property

Usage

Word Pro: Graphic class

A graphic object in a document.

Base Classes

BaseObject\Content\GraphicOleObject

Derived Classes

None.

Contained by

WPAApplication in the Graphic Property

Usage

Word Pro: GroupLayoutCollection class

A collection of group layout objects in the foundry of a specific division.

Base Classes

BaseObject\BaseCollection

Derived Classes

None.

Contained by

Foundry in the Groups Property

Usage

Use this collection to access any of the group layout objects in the foundry of a specific division.

Word Pro: GroupLayout class

The layout for a group of objects.

Base Classes

BaseObjectLayout

Derived Classes

None.

Contained by

Usage

Word Pro: Gutter class

Controls the appearance of a gutter line. A gutter line is the border which appears in the center of the gap between text columns.

Base Classes

BaseObject\BorderLines

Derived Classes

None.

Contained by

Layout in the Gutter Property

Usage

The properties provided by the Gutter class are equivalent to the "Line style," "Line width," and "Line color" settings on the Columns panel of the InfoBox for certain layout objects.

Word Pro: HeaderLayoutCollection class

A collection of header layout objects in the foundry of a specific division.

Base Classes

BaseObject\BaseCollection

Derived Classes

None.

Contained by

Foundry in the Headers Property

Usage

Use this collection to access any of the header layout objects in the foundry of a specific division.

Word Pro: HeaderLayout class

The layout for a header object.

Base Classes

BaseObjectLayout

Derived Classes

None.

Contained by

Usage

Word Pro: HyphenationOptions class

Hyphenation options display on the Options panel of the Document Properties dialog box.

Base Classes

BaseObject

Derived Classes

None.

Contained by

DivisionOptions in the HyphenationOptions Property

Usage

Word Pro: IconBarCollection class

A collection of icon bar objects in the IconBarManager class.

Base Classes

BaseObject\BaseCollection

Derived Classes

None.

Contained by

IconBarManager in the IconBars Property

Usage

This collection of IconBar objects are indexed by their icon bar name.

Word Pro: IconBarManager class

A tool for managing icon bar objects in the application. Keeps and manages the list of icon bar objects in the document. You must go through the IconBarManager before using the IconBar property.

Base Classes

BaseObjectWindow

Derived Classes

None.

Contained by

ApplicationWindow in the IconBarManager Property

Usage

Used in conjunction with IconBar and IconBarCollection objects. You can use the IconBarManager to select, find, add, or remove icon bar objects. Word Pro keeps a list of icon bar sets in the IconBarCollection object.

Word Pro: IconBar class

A bar containing a set of icons that represent shortcuts for Word Pro functions, commands, and scripts.

Base Classes

BaseObject

Derived Classes

None.

Contained by

Usage

Use the properties of the IconBar class to set or retrieve information about icon bars. Information such as position on screen, height, and width is available. You can also work with icons contained in an icon bar, performing actions such as adding icons to an icon bar.

Word Pro: Indent class

The indentation of text from the right or left margins.

Base Classes

BaseObject

Derived Classes

RelativeIndent

Contained by

ClickHere in the Indent Property

ParagraphStyle in the Indent Property

Text in the Indent Property

TextMarker in the Indent Property

Usage

Word Pro: IndexSection class

A section that contains the index for a document.

Base Classes

BaseObject\Section

Derived Classes

None.

Contained by**Usage**

Word Pro: Join class

The rectangular bounding area that connects lines surrounding a frame, page, or table layout.

Base Classes

BaseObject

Derived Classes

None.

Contained by

Layout in the Join Property

Usage

A join connects line styles at each corner of a page, frame, or table. Page, frame, and table layouts all contain join and line objects. Each line object has a set of join objects it can use. For a list of join objects and the corresponding line objects, see the JoinType property.

Word Pro: Kinsoku class

The page layout object for the Asian version of Word Pro.

Base Classes

BaseObject

Derived Classes

None.

Contained by

[ClickHere](#) in the [Kinsoku](#) Property

[ParagraphStyle](#) in the [Kinsoku](#) Property

[Text](#) in the [Kinsoku](#) Property

[TextMarker](#) in the [Kinsoku](#) Property

Usage

Word Pro: Language class

Base Classes

BaseObject

Derived Classes

None.

Contained by

[CharacterStyle](#) in the [Language](#) Property

[ClickHere](#) in the [Language](#) Property

[DivisionOptions](#) in the [Language](#) Property

[FindAndReplace](#) in the [ReplaceLanguage](#) Property

[FindAndReplace](#) in the [SearchLanguage](#) Property

[ParagraphStyle](#) in the [Language](#) Property

[Text](#) in the [Language](#) Property

[TextMarker](#) in the [Language](#) Property

Usage

Word Pro: LayoutCollection class

A collection of layout objects in the foundry of a specific division.

Base Classes

BaseObject\BaseCollection

Derived Classes

None.

Contained by

Foundry in the Layouts Property

Layout in the ChildLayouts Property

Usage

Use this collection to access any of the layout objects in the foundry of a specific division.

Word Pro: LayoutOverride class

Allows you to add items to a DivisionInfo object's layout override list.

Base Classes

BaseObject

Derived Classes

None.

Contained by

DivisionInfo in the LayoutOverride Property

Usage

Layout objects, such as headers and footers, are usually stored as children of a page layout. A layout override list allows you to remove layout objects from the page layout heirarchy in order to control their placement. For example, you can place a frame on a DivisionInfo object's layout override list so that it appears on a specific page, regardless of the page layout that is in effect.

Word Pro: Layout class

A class which defines properties and methods that are common to all Word Pro layout objects. An explanation of layout objects is provided below under Usage. This information applies to all layout objects in Word Pro, but each layout object may exhibit minor differences. These differences are noted in the documentation of each specific layout class.

Base Classes

[BaseObject](#)

Derived Classes

[CellGroupLayout](#)

[CellLayout](#)

[ConnectedLayout](#)

[EndnoteLayout](#)

[FooterLayout](#)

[FootnoteLayout](#)

[FrameGroupLayout](#)

[FrameLayout](#)

[GroupLayout](#)

[HeaderLayout](#)

[NoteLayout](#)

[PageLayout](#)

[RowLayout](#)

[RubyLayout](#)

[SuperTableGroupLayout](#)

[SuperTableLayout](#)

[TableHeadingLayout](#)

[TableLayout](#)

[TOCSuperTableLayout](#)

Contained by

[BaseContainer](#) in the [Layout](#) Property

[BaseTable](#) in the [Layout](#) Property

[BaseTable](#) in the [CurrentColumn](#) Property

[Layout](#) in the [Footer](#) Property

[Layout](#) in the [Style](#) Property

[Layout](#) in the [RightPage](#) Property

[Layout](#) in the [Header](#) Property

[Layout](#) in the [LeftPage](#) Property

[Marker](#) in the [Layout](#) Property

[WPApplication](#) in the [Layout](#) Property

[WPApplication](#) in the [CurrentColumn](#) Property

Usage

The Layout class also provides the foundation of formatting attributes for a number of derived classes. An object created from Layout or any of these derived layout classes is known as a layout object. A layout object gets its properties, methods, and events from its derived layout class. Very few objects in Word Pro actually use Layout as their data type.

For example, a frame is one type of layout object. It gets its properties, methods, and events from the FrameLayout class. Other layout objects include: Cells (CellLayout), Endnotes (EndnoteLayout), Footers (FooterLayout), Footnotes

(FootnoteLayout), Headers (Header Layout), Comment Notes (NoteLayout), Pages (PageLayout), Rows (RowLayout), and Tables (TableLayout). These derived classes allow you to access and modify physical appearance attributes of these layout objects.

Default layouts and the Style property

Each time you create one of these layout objects, Word Pro gets that object's property values from a default layout object. A default layout object is represented by a style in Word Pro. For example, when you look at the Style panel in the InfoBox, you see a Default Table style for table objects, a Default Frame style for frame objects, and so on. A layout object's default property values are always accessible through the Style property of that layout object.

Layout objects and collections

Layout objects that are created from the same derived layout class are stored together in collections. You can use these collections to access all layout objects in the collection, or you can reference a particular layout object in the collection. For example, by using the FrameLayouts collection, you can modify all FrameLayout objects in the collection to be of a certain height. For more information on working with collections, see [Overview: Word Pro LotusScript Collection Classes](#).

Layout objects and containers

At many locations within your document, the current focus includes multiple layouts. For example, your cursor may be in a cell in a table on a page. In this case, the cell, the table, and the page all have associated layout objects. These layout objects may be combined with other objects into related groups, known as containers. For more information on containers and their associated layouts, see the Help topic titled [Word Pro: BaseContainer class](#).

Word Pro: LineNumberOptions class

The line number options displayed in the Line Numbering dialog box on the Options panel of the Document Properties dialog box.

Base Classes

BaseObject

Derived Classes

None.

Contained by

Division in the LineNumberOptions Property

TextDocument in the LineNumberOptions Property

Usage

Word Pro: LWPTimer class

Allows you to schedule a specific event every x number of seconds.

Base Classes

BaseObject

Derived Classes

None.

Contained by**Usage**

Word Pro: Macro class

An object that automates tasks in the application.

Base Classes

BaseObject

Derived Classes

None.

Contained by

ApplicationWindow in the Macro Property

Usage

Word Pro: MailRouting class

Directs the distribution of e-mail messages in Word Pro. Each document has an internal data structure to which the MailRouting Class points.

Base Classes

BaseObject

Derived Classes

None

Contained by

TextDocument in the MailRouting Property

Usage

Contains the name of the route and the names of people in the route. The Word Pro document must be open for MailRouting to function. Values are 0 when starting a new route and 1 when editing a route.

Word Pro: MarkerCollection class

A collection of marker objects in the foundry of a specific division.

Base Classes

BaseObject\BaseCollection

Derived Classes

None.

Contained by

Foundry in the Markers Property

Usage

Use this collection to access any of the marker objects in the foundry of a specific division.

Word Pro: Marker class

Hidden objects used to attach some data or functionality to a specific location in the document.

Base Classes

BaseObject

Derived Classes

[ClickHere](#)

[PowerField](#)

[RubyMarker](#)

[TableMarker](#)

[TextMarker](#)

Contained by

Usage

Word Pro: MenuItemCollection class

A collection of menu item objects in the MenuItem class.

Base Classes

BaseObject\BaseCollection

Derived Classes

None.

Contained by

MenuItem in the Items Property

Usage

Use this collection to access any of the menu item objects in the MenuItem class.

Word Pro: MenuItem class

Any menu or item on a menu.

Base Classes

BaseObject

Derived Classes

None.

Contained by

ApplicationWindow in the LwpMenuBar Property

ApplicationWindow in the RightMouseMenus Property

ApplicationWindow in the FreeMenus Property

Usage

Allows you to access Word Pro menus and create custom menus. Custom menu items can execute scripts or call predefined Word Pro menu commands. For more information on creating menu items and assigning functionality to those menu items, see the Action property and the NewItem method.

MenuItem objects can be nested within other MenuItem objects. For example, Word Pro's File menu is a MenuItem object. The File menu contains other MenuItem objects, such as Open, Close, and Save. In this case, the File menu is the parent MenuItem object, and the MenuItem objects that appear within the File menu are child MenuItem objects.

Word Pro: MergeOptions class

The merge options displayed in the Merge Assistant dialog box.

Base Classes

BaseObject

Derived Classes

None.

Contained by

TextDocument in the MergeOptions Property

Usage

Word Pro: NoteContainer class

The container object for notes. This object only exists for one note at a time, and only when there is a note with in the focus.

Base Classes

BaseObject\BaseContainer\FrameContainer

Derived Classes

None.

Contained by

Not contained in a property of any object.

Usage

The primary use for a NoteContainer object is to provide quick and easy access to the NoteLayout object for the currently active note. A NoteContainer object always represents the note that currently has the focus. Therefore, if you assign a NoteContainer object to a variable, you can use that variable to access the currently active note. However, you must remember that the note referenced by the variable changes as the focus moves from one note to another. This is because the variable references the NoteContainer object, and the NoteContainer object always represents the note that has the focus. If there is no note with in the focus, there is no NoteContainer object. Therefore, a variable that stores a NoteContainer object has a null value whenever the focus does not contain a note. There is never more than one NoteContainer object at any time.

For more information about container objects, see [BaseContainer](#).

Word Pro: NoteLayoutCollection class

A collection of note layout objects in the foundry of a specific division.

Base Classes

BaseObject\BaseCollection

Derived Classes

None.

Contained by

Foundry in the NoteLayouts Property

Usage

Use this collection to access any of the note layout objects in the foundry of a specific division.

Word Pro: NoteLayout class

The layout for a note object.

Base Classes

BaseObjectLayout

Derived Classes

None.

Contained by

Usage

Word Pro: Numbering class

Numbering pages in a document, or numbering lines of text or blank lines on a page in a document.

Base Classes

BaseObject

Derived Classes

None.

Contained by

[ClickHere](#) in the [Numbering](#) Property

[ParagraphStyle](#) in the [Numbering](#) Property

[Text](#) in the [Numbering](#) Property

[TextMarker](#) in the [Numbering](#) Property

Usage

Word Pro: NumericFormatSubset class

Allows you to change the formatting of numeric values within table cells.

Base Classes

BaseObject

Derived Classes

None.

Contained by

NumericFormat in the AnyNumber Property

NumericFormat in the Negative Property

NumericFormat in the Zero Property

Usage

Word Pro: NumericFormat class

The format of numbers in a table cell object.

Base Classes

BaseObject

Derived Classes

None.

Contained by

Layout in the NumericFormat Property

Usage

The NumericFormat class allows you to format numeric values within table cells. You can modify the way that Word Pro represents values by applying scientific notation, currency formatting, fixed decimal formatting, and so on, in the FormatType property. Three additional categories are available for formatting numeric values within table cells. The Negative property allows you to format negative values; the Zero property allows you to format zero values; the AnyNumber property allows you to modify the format of any other number.

Word Pro: OleObjectCollection class

A collection of Ole objects in the foundry of a specific division.

Base Classes

BaseObject\BaseCollection

Derived Classes

None.

Contained by

Foundry in the OleObjects Property

Usage

Use this collection to access any of the Ole objects in the foundry of a specific division.

Word Pro: OleObject class

An OLE object with in the application.

Base Classes

BaseObject\Content\GraphicOleObject

Derived Classes

None.

Contained by

WPApplication in the OleObject Property

Usage

Word Pro: Options class

The options displayed in a dialog box in a division or a document.

Base Classes

BaseObject

Derived Classes

None.

Contained by

Division in the DocOptions Property

TextDocument in the Options Property

TextDocument in the DocOptions Property

Usage

Word Pro: OutlineSeqCollection class

A collection of OutlineStyleSequence objects. The scope of this collection is usually limited to a single Division object.

Base Classes

BaseObject\BaseCollection

Derived Classes

None.

Contained by

Foundry in the OutlineStyleSequences Property

Usage

Word Pro: OutlineSeqItemCollection class

A collection of OutSeqItem objects. The scope of this collection is usually limited to a single TextDocument object.

Base Classes

BaseObject\BaseCollection

Derived Classes

None.

Contained by

OutlineStyleSequence in the OutlineSeqItems Property

Usage

```
'Example: GetProfileString method
' This example prints the last 6 open files names to the LotusScript Output
' panel.
' RUNTIME DEPENDENCIES: You must have a document open for this script to work.

Key = "LastOpen1"
Section = "WordProUser"
Defstring0 = ""
IniFileType0 = $LwpIniUserPrefs
WhichIniLocation0 = ""
IniName0 = "lwpuser.ini"
For x = 1 To 6
    Key = "LastOpen" & x
    Print .GetProfileString(Section, Key, Defstring0, IniFileType0,
WhichIniLocation0,IniName0)
Next
```

```
'Example: GetStandardButtonId method
' This example simulates clicking on the font status bar button.
' RUNTIME DEPENDENCIES: You must have a document open for this script to work.

Dim StatBar As StatusBar
Dim ButtonId as Integer

Set StatBar = .ApplicationWindow.StatusBar
Forall Button In StatBar.StatusBarButtons
    ButtonId = Button.GetButtonId
    If (ButtonId = StatBar.GetStandardButtonId($LwpStandButtFontButton)) Then
        Button.SimulateButtonClick
    End If
End Forall
```

'Example: GetTabAlignChar method

' This example inserts a numeric tab 4 inches from the left margin,
'with the period being used as the alignment character.

'After the tab is set, the value of the alignment character is
'printed to the LotusScript output panel.

```
.Text.TabRack.InsertOne 5760, $LwpTabTypeNumeric , $LwpTabLeaderNone,  
$LwpTabRelativeLeft , Asc("."))
```

```
Align = .Text.TabRack.GetTabAlignChar(0)
```

```
Print "Align on = " & Align & " (the character " & Chr(Align) & ")"
```

'Example: GetTabLeaderType method

' This example inserts a numeric tab 4 inches from the left margin,

'with the period being used as the alignment character.

'After the tab is set, the value of the leader type is

'printed to the LotusScript output panel.

```
.Text.TabRack.InsertOne 5760, $LwpTabTypeNumeric , $LwpTabLeaderDot,  
$LwpTabRelativeLeft , Asc("."))
```

```
Leader = .Text.TabRack.GetTabLeaderType(0)
```

```
Print "Leader = " & Leader
```

'Example: GetTabPosition method

' This example inserts a numeric tab 4 inches from the left margin,

'with the period being used as the alignment character.

'After the tab is set, the value of the tab's position is

'printed to the LotusScript output panel.

```
.Text.TabRack.InsertOne 5760, $LwpTabTypeNumeric , $LwpTabLeaderNone,  
$LwpTabRelativeLeft , Asc("."))
```

```
Position = .Text.TabRack.GetTabPosition(0)
```

```
Print "The tab's position is " & Position & " or " & position/1440 & " inches."
```

```
'Example: GetTabRelativeType method
' This example inserts a numeric tab 4 inches from the left margin,
'with the period being used as the alignment character.
'After the tab is set, the relative position is
'printed to the LotusScript output panel.

.Text.TabRack.InsertOne 5760, $LwpTabTypeNumeric , $LwpTabLeaderNone,
$LwpTabRelativeLeft , Asc("."))
RelType = .Text.TabRack.GetTabRelativeType(0)
Print "Relative Type = " & RelType
```

'Example: GetTabType method

' This example inserts a numeric tab 4 inches from the left margin,
'with the period being used as the alignment character.

'After the tab is set, the type of tab is

'printed to the LotusScript output panel.

```
.Text.TabRack.InsertOne 5760, $LwpTabTypeNumeric , $LwpTabLeaderNone,  
$LwpTabRelativeLeft , Asc("."))
```

```
TabType = .Text.TabRack.GetTabType(0)
```

```
Print "Type = " & TabType
```



```
'Example: GetText method
' This example prints the current word, sentence and paragraph as well as
' each word of the current paragraph to the Script Editor Output panel.
' RUNTIME DEPENDENCIES: You must have a document open with the cursor
' positioned on a line with text for this script to work.

Print .Text.GetText($LwpGetObjectWord, False)
Print .Text.GetText($LwpGetObjectSentence, False)
Print .Text.GetText($LwpGetObjectParagraph, False)
While Not .Text.AtEndOfParagraph
    Print .Text.GetText($LwpGetObjectWord,True)
Wend
```

```
'Example: GetUniqueName method
' This example creates a bookmark using a unique name determined by
' the GetUniqueName method.
' RUNTIME DEPENDENCIES: You must have a document open for this script to work.
```

```
Dim MarkerName as String      ' the hexadecimal name of the marker
Dim NewBookMarkName as String ' the user name for the bookmark
MarkerName = .Mark($LwpMarkerTypeBookmark)
NewBookMarkName = .Division.BookMarkManager.GetUniqueName
.Division.BookmarkManager.AddBookmark NewBookMarkName, MarkerName
Print NewBookMarkName ' display new name on LotusScript output panel.
```

```
'Example: GetWordMisspelled method
' This example inserts a sentence of text into the current document. Each word
' starting with the first is then selected and checked for misspelling.
' RUNTIME DEPENDENCIES: You must have a document open for this script to work.

.Type "This is a Sintence of text."
.Text.MoveToStart $LwpLocationTypeLine

Do
    .SelectWord
    Stat = .Text.GetWordMisspelled
    If Stat = False Then
        MsgBox "Mispelled " & .Text.GetText($LwpGetObjectTypeInfoWord, False)
    End If
    NextWord = .Text.Forward ($LwpNavigateObjectTypeWord, 1)
Loop Until (.Text.AtBeginningOfParagraph = True) Or (NextWord = False)
```

'Example: GIFAnimationOn property
'This example asks the users if GIF images should be animated within Word Pro,
'and then sets the appropriate option.

```
stat = MessageBox ("Should animated GIF files display animated?", 36, "Example  
Script")
```

```
If stat = 6 Then ' user said yes  
    .Preferences.GIFAnimationOn = True  
Else  
    .Preferences.GIFAnimationOn = False  
End If
```

```
'Example: Glossarize method
'This example opens the glossary file and adds a glossary entry

.Type "This is some text for the new glossary item"
.SelectSentence
.InternalCopy True 'This is needed since Glossarize works off what is in the Temporary
Foundry
.ApplicationWindow.UserInterfacePrefs.OpenDocsVisible = False
.GlossaryOpen "GLOSSARY.GLS", "Lotus Word Pro"
.ApplicationWindow.UserInterfacePrefs.OpenDocsVisible = True

Forall Gloss In .Division.Foundry.Glossarys
    GlossItem$ = "SampleEntry"
    Gloss.Glossarize(GlossItem$)
    Exit Forall
End Forall
.SaveGlossary "GLOSSARY.GLS", ""
.Documents("GLOSSARY.GLS").Close
```

```
'Example: GlossaryDataFileName property  
'This example lets the user change the glossary file name found in user preferences.  
'If the user changes the name, the new name is saved.
```

```
GlossName = .ApplicationWindow.UserInterfacePrefs.GlossaryDataFileName  
NewName = Inputbox ("Type a new glossary file name:", "Example Script", GlossName)  
If NewName <> "" Then ' user didn't cancel  
    .ApplicationWindow.UserInterfacePrefs.GlossaryDataFileName = NewName  
End If
```

```
'Example: GlossaryDataFiles property
'This example displays a message box with a list of glossary data files.

AllNames = ""
Forall GlossName In .ApplicationWindow.UserInterfacePrefs.GlossaryDataFiles
    allnames = AllNames & glossname & ", "
End Forall
MessageBox "The glossary filenames are: " & AllNames, MB_OK, "Example Script"
```

```
'Example: GlossaryDataPaths property
'This example displays a message box with the names of the glossary paths.

AllPaths = ""
Forall GlossPath In .ApplicationWindow.UserInterfacePrefs.GlossaryDataPaths
    AllPaths = AllPaths & GlossPath & ", "
End Forall
MessageBox "The glossary pathnames are: " & AllPaths, MB_OK, "Example Script"
```


'Example: GlossaryInsert method

'This example creates a glossary entry, then inserts it in the document.

```
.Type "This is a glossary entry.[shiftHome]"  
.CreateGlossaryEntry "glossary.gls", "Gloss Entry"  
.Type "[End][Enter][Enter]"  
.GlossaryInsert "glossary.gls", "Gloss Entry"
```

```
'Example: GlossaryOpen method
' This example opens the glossary file named GLOSSARY.GLS and prints all of
' the glossary entries to the Lotus Script Output panel.
' RUNTIME DEPENDENCIES: You must have a document open for this script to work.

.ApplicationWindow.UserInterfacePrefs.OpenDocsVisible = False
.ApplicationWindow.UserInterfacePrefs.OpenReadOnly = True
GlossPath = .ApplicationWindow.UserInterfacePrefs.GlossaryPath
.GlossaryOpen GlossPath & "\GLOSSARY.GLS", "Lotus Word Pro"
.ApplicationWindow.UserInterfacePrefs.OpenDocsVisible = True
.ApplicationWindow.UserInterfacePrefs.OpenReadOnly = False
Forall Gloss In .Division.Foundry.Glossarys
    Count% = Gloss.NumRows
    For Item% = 1 To (Count% - 1)
        GlossItem$ = Gloss.EnumerateTerm(Item%)
        GlossText$ = Gloss.ExtractText(GlossItem$)
        Print GlossText$
    Next
End Forall
.Documents("GLOSSARY.GLS").Close
```

```
'Example: GlossaryPath property
'This example lets the user change the glossary path found in user preferences.
'If the user changes the number, the new path is saved.

GPath = .ApplicationWindow.UserInterfacePrefs.GlossaryPath
NewPath = Inputbox ("Type a new glossary path:", "Example Script", GPath)
If NewPath <> "" Then ' user didn't cancel
    .ApplicationWindow.UserInterfacePrefs.InsertPath $LwpSetGlossaryPath, NewPath
End If
```

'Example: GoToBookmark method
' This example first creates a bookmark in the current document, then moves
' the cursor out of the bookmark by splitting the paragraph, and then displays
' a message.
' When you click OK, the cursor goes back to the bookmark.
' RUNTIME DEPENDENCIES: You must have a document open for this script to work.

```
Dim MarkerName as String  
MarkerName = .Mark($LwpMarkerTypeBookmark)  
.Division.BookmarkManager.AddBookmark "ExampleBookmark", MarkerName  
.Text.SplitParagraph  
MessageBox "Click OK to go to the bookmark.",MB_OK,"Example Script"
```

```
.GoToBookmark("ExampleBookmark")
```

'Example: GoToContainer method

' This example selects the first frame in the current division.

' RUNTIME DEPENDENCIES: You must have a document open which contains a frame
' for this script to work.

.Page.GoToContainer \$LwpGoToLocationFrame

```
'Example: GoToDivision method
'This example displays a message box asking the user for the name of
'a division to Go To. The script then goes to the division,
'or reports an error if the division cannot be found.

GoToName = Inputbox ("Type the name of the division you want to Go To:")
stat = .GoToDivision (GoToName)
If stat = 0 Then ' the method failed
    MsgBox "Couldn't find the division", MB_OK, "Example Script"
End If
```

```
'Example: GoToLayout method
' This example creates a table with 5 rows and 5 columns into the current
' document. The cursor is then positioned to row 2, column 3.
' RUNTIME DEPENDENCIES: You must have a document open for this script to work.
```

```
Dim MyTable As Table
.CreateTable False, "Default Table", 5,5
Set MyTable = .Table
' rows & cols are zero based
MyTable.CellLayout(1,2).GotoLayout
```

'Example: GotoNextParallelColumn method

' This example creates a parallel column and navigate to the second column.

' RUNTIME DEPENDENCIES: You must have a document open for this script to work.

.CreateParallelColumns 3, \$LtsAlignmentHorizCenter

.GotoNextParallelColumn

'Example: GoToObject method

' This example moves the insertion point to document header.

' RUNTIME DEPENDENCIES: You must have a document open for this script to work.

.GoToObject ("Header", 1)

'Example: GoToPage method

' This example goes to the second page of the current document.

' RUNTIME DEPENDENCIES: You must have a document open with two or more pages
' for this script to work.

.GoToPage 2

```
'Example: GoToSection method
' This example inserts two sections and then goes to each one while selecting
' the sections' contents.
' RUNTIME DEPENDENCIES: You must have a document open for this script to work.

.InsertSection "Default Page", True, True, $LwpStartTypeNextpage, False, True
.InsertSection "Default Page", True, True, $LwpStartTypeNextpage, False, True
Forall Section In .Division.Foundry.Sections
    Section.GoToSection
    .SelectSection
End Forall
```

```
'Example: GoToTableCell method
' This example creates a table with 5 rows and 5 columns into the current document.
The cursor is then advanced to
' the next cell.
' RUNTIME DEPENDENCIES: You must have a document open for this script to work.
```

```
Dim MyTable As Table
.CreateTable False, "Default Table", 5,5
Set MyTable = .Table
MyTable.GoToTableCell $LwpCellObjectType,True
```

'Example: GoTo method

' This examples illustrates how to validate the contents of a Click Here
' Block. After the Click Here Block loses focus, the ExitClickHere event is
' called. If more than 15 characters have been entered, a message box is
' displayed and the Click Here Block is then re-selected.
' RUNTIME DEPENDENCIES: You must have a document open and a Click Here Block
' for this script to work.

```
Sub Exitclickhere(Source As Clickhere, Clickherename As String)
    Dim Temp as String
    Const MaximumLen = 15
    Temp = Source.GetMarkedText
    If Len(Temp) > MaximumLen Then
        Messagebox "Please Enter less than 15 characters."
        Source.GoTo(True)
    End If
End Sub
```

'Example: GrammarFormalityLevel property

'This example sets the grammar checking formality level to "Formal" for the current document.

.ApplicationWindow.UserInterfacePrefs.GrammarOptions.**GrammarFormalityLevel** = 2

'Example: GrammarOptions property

'This example sets grammar options in the current document

'for grammatical style.

.ApplicationWindow.UserInterfacePrefs.**GrammarOptions**.MaximumWordsinaSentence = 35

.ApplicationWindow.UserInterfacePrefs.**GrammarOptions**.SpacesBetweenSentences = 1

.ApplicationWindow.UserInterfacePrefs.**GrammarOptions**.MaxIdenticalConsecSentOpens = 4

.ApplicationWindow.UserInterfacePrefs.**GrammarOptions**.MaxIdenticalSentOpensWithin10 = 5

.ApplicationWindow.UserInterfacePrefs.**GrammarOptions**.GrammarProofLevel = 0

.ApplicationWindow.UserInterfacePrefs.**GrammarOptions**.GrammarFormalityLevel = 2

'Example: GrammarProofLevel property

'This example sets the grammar checking proof level to a full proofing in
'the current document.

.ApplicationWindow.UserInterfacePrefs.GrammarOptions.**GrammarProofLevel** = 0


```
'Example: GraphicImports property
'This example lists all the graphic import types in a message box.

AllFilters = ""
Forall GraphicFilt In .ApplicationWindow.Filter.GraphicImports
    AllFilters = AllFilters & GraphicFilt & ", "
End Forall
MessageBox "The graphic filter names are: " & AllFilters, MB_OK, "Example Script"
```

```
'Example: Green property  
'This example lets the user specify a level of green for text, then sets the desired  
level.
```

```
Dim GreenLevel As Integer  
Pct = Inputbox ("What percent green should this text be?", "Example Script", "100")  
If Pct = "" Then Exit Sub  
GreenLevel = 255*Pct/100  
.Text.Font.FontColor.Red = 0  
.Text.Font.FontColor.Blue = 0  
.Text.Font.FontColor.Green = GreenLevel  
.Text.Font.FontColor.Override = $LwpColorOverrideRgb
```

'Example: Greeting property
'This example lets the user change the greeting set up in TeamSecurity.
'If the user changes the greeting, the new greeting is saved.
' If you don't have access to the TeamSecurity dialog box, this will not let you set a new greeting.

```
Greeting = .ActiveDocument.DocControl.Greeting  
NewGreeting = Inputbox ("Type a new greeting:", "Example Script", Greeting)  
If NewGreeting <> "" Then ' user didn't cancel  
    .ActiveDocument.DocControl.Greeting = NewGreeting  
End If
```

'Example: GridDistance property

'This example creates a frame, then displays a grid with lines spaced one
'inch apart. The contents of the frame is then set to snap to the grid.

.CreateFrame

.Frame.Layout.GridType = \$LwpGridTypeLines

.Frame.Layout.**GridDistance** = 1440 ' 1440 twips = 1 inch

.Frame.Layout.IsSnapTo = True

'Example: GridType property

'This example creates a frame, then displays a grid with lines spaced one
'inch apart. The contents of the frame is then set to snap to the grid.

.CreateFrame

.Frame.Layout.**GridType** = \$LwpGridTypeLines

.Frame.Layout.GridDistance = 1440

.Frame.Layout.IsSnapTo = True

'Example: GroupDivision method

'This example creates a new division in the current document. The new
'division and the original are then grouped.

.ActiveDocWindow.WinViewPrefs.IsViewSectionTabs = True

.ApplicationWindow.SectionTabs.AddNewSectionTabs

.GroupDivision

'Example: Gutter property

'This example creates newspaper columns in the current document, and
'places a double line between the columns.

```
.Page.Layout.RightPage.NumCols = 2
```

```
.Page.Layout.RightPage.Gutter.LinePlacement = &Hf
```

```
.Page.Layout.RightPage.Gutter.AllBorders.Pattern = $LtsBorderPatternDouble
```

```
.Page.Layout.RightPage.Gutter.AllBorders.WidthInTwips = 80 ' 4 pt. line
```

```
'Example: HasNamedProperty method
' This example checks for the existence of a named property, 'ExampleProp' in the
active document.
' If the property does not exist, the example creates it and assigns it a value.
' The value is then printed to the LotusScript Output panel.
' RUNTIME DEPENDENCIES: You must have a document open for this script to work.

Dim stat as Integer

stat = .ActiveDocument.HasNamedProperty("ExampleProp")
If stat = False Then
    .ActiveDocument.SetNamedProperty "ExampleProp", "Here is some data."
End If
Print .ActiveDocument.GetNamedProperty ("ExampleProp")
```



```
'Example: HasTabs property
'This example reports if the paragraph style used by the text at the
'insertion point has a tab ruler.

If .Text.ParagraphStyle.HasTabs = True then
    MessageBox .Text.ParagraphStyleName & " has a tab ruler", MB_OK, "Example Script"
Else
    MessageBox .Text.ParagraphStyleName & " does not have a tab ruler", MB_OK,
"Example Script"
End If
```

'Example: Header property

'This example creates a new document, and then changes the header margins.

.NewDocument

.Layout.**Header**.IsMarginSameAsParent = False

.Layout.**Header**.MarginLeft = 2880 ' 2 inches

.Layout.**Header**.MarginRight = 720 ' 1/2 inch

'Example: Heading property

'This example assigns an outline level of 1 to the Heading 1 paragraph style

'and makes it a heading style.

'It assigns an outline level of 2 to the Heading 2 paragraph style,

'and makes it not a heading style.

```
.Division.Foundry.ParagraphStyles("Heading 1").DocumentLevel = 1
```

'Make this a heading style

```
.Division.Foundry.ParagraphStyles("Heading 1").Heading = 1
```

'Assign the Default Outline numbering sequence, and set the numbering position to 1.

```
.Division.Foundry.ParagraphStyles("Heading 1").Bullet.Name = "Default Outline"
```

```
.Division.Foundry.ParagraphStyles("Heading 1").NumberingPosition = 1
```

```
.Division.Foundry.ParagraphStyles("Heading 2").DocumentLevel = 2
```

'Don't make this a heading style

```
.Division.Foundry.ParagraphStyles("Heading 2").Heading = 0
```

```
.Division.Foundry.ParagraphStyles("Heading 2").Bullet.Name = "Default Outline"
```

```
.Division.Foundry.ParagraphStyles("Heading 2").NumberingPosition = 2
```

```
'Example: Height property
Dim CR As String*1
Dim IcnPallet As String
Dim MsgStr As String
Dim IcnMgr As IconBarManager

IcnPallet = "Comment Tools"
CR = Chr(10)
Set IcnMgr = .ApplicationWindow.IconBarManager

With IcnMgr.IconBars(IcnPallet)
    MsgStr = "Height = " & .Height & CR
    MsgStr = MsgStr & "IconBarPositionState = " & .IconBarPositionState & CR
    MsgStr = MsgStr & "PositionType = " & .PositionType & CR
    MsgStr = MsgStr & "ScreenPositionX = " & .ScreenPositionX & CR
    MsgStr = MsgStr & "ScreenPositionY = " & .ScreenPositionY
    MessageBox MsgStr, 64, "Script Example - " & .Name
End With
```

'Example: HelpText property

'This example creates a Click Here block, and sets the Help text for the block.

Dim ClickHereName as String

ClickHereName = .InsertClickHere()

'First enable the Help text

.Division.Foundry.ClickHeres(ClickHereName).**UsesHelp** = True

'And set he Help text contents

.Division.Foundry.ClickHeres(ClickHereName).**HelpText** = "Click here to type Text"

'Example: Help method

' This example displays the Word Pro 97 LotusScript object model help.

.Help "C:\Lotus\WordPro\Wp0n71en.hlp"

```
'Example: HiddenMode property
'This example sets the hidden attribute, then types some text.
'The option to show hidden text is then deselected, causing the text to be hidden.
.Text.Attributes.HiddenMode = True ' put on the Hidden attribute.
.Type "Now is the time for all good men to come to the aid of their party."
.Text.Attributes.HiddenMode = False
CurrentDocument.DivisionOptions.ShowHiddenText = False ' don't display hidden text
MessageBox "Click OK to show hidden text", MB_OK, "Example Script"
CurrentDocument.DivisionOptions.ShowHiddenText = True
```

'Example: HideIconBars method

' This example hides all SmartIcon bars the redisplays them after the message
' box is closed.

.ApplicationWindow.IconBarManager.HideIconBars

MessageBox "Click OK to show SmartIcon bars.",MB_OK,"Example Script"

.ApplicationWindow.IconBarManager.ShowIconBars


```
'Example: HideIconBar method
' This example displays and then hides the "Comment Tools" SmartIcon bar
' RUNTIME DEPENDENCIES: You must have a document open for this script to work.

Dim IcnPallet As String
Dim MsgStr As String
Dim IcnMgr As IconBarManager

IcnPallet = "Comment Tools"
Set IcnMgr = .ApplicationWindow.IconBarManager

' Set the context and show the bar
IcnMgr.IconBars(IcnPallet).ShowInContext = True
IcnMgr.IconBars(IcnPallet).Show

MsgStr = "|" & IcnPallet
MsgStr = MsgStr & "|" pallet is now displayed, click OK to hide this pallet"|
MessageBox MsgStr, 48, "Example Script"

' Reset the context and hide the bar.
IcnMgr.IconBars(IcnPallet).ShowInContext = False

IcnMgr.IconBars(IcnPallet).HideIconBar
```

```
'Example: HideStatusBar method
' This example hides the status bar if it is visible, and shows it if it is
' hidden.
If .ApplicationWindow.StatusBar.Visible = True Then
    .ApplicationWindow.StatusBar.HideStatusBar
Else
    .ApplicationWindow.StatusBar.ShowStatusBar
End If
```

'Example: HighLightMode property

'This example enables Highlight mode, then types some text.

.Text.Attributes.**HighlightMode** = True

.Type "Now is the time for all good men to come to the aid of their party."

.Text.Attributes.**HighlightMode** = False

'Example: HighlightToggle method

' This example toggles the Review & Comment tools highlighter on or off.

' RUNTIME DEPENDENCIES: You must have a document open for this script to work.

.HighlightToggle

'Example: HiLiteColor property

'This example sets the color for highlighting and comment notes for the
'current editor to green.

```
.ActiveDocument.EditorManager.CurrentEditor.HiLiteColor.Red = 0  
.ActiveDocument.EditorManager.CurrentEditor.HiLiteColor.Blue = 0  
.ActiveDocument.EditorManager.CurrentEditor.HiLiteColor.Green = 255  
.ActiveDocument.EditorManager.CurrentEditor.HiLiteColor.Override =  
$LwpColorOverrideRgb
```

```
'Example: HorizontalSplitWindow property
' create a new document, then split the document's window horizontally
  .newdocument
  ' set the option for a vertical window
  .ApplicationWindow.UserInterfacePrefs.HorizontalSplitWindow = true
  ' the NewWindow method actually executes the split
  .NewWindow
```

```
'Example: HourGlass method
' This example displays an hourglass mouse pointer while counting to 50000
' after which the original pointer is restored.

.HourGlass(True)
For i = 1 to 50000
Next
.HourGlass(False)
```

'Example: HyphenateLastWordInColumnOrPage property

'This example asks the user whether to hyphenate the last word on a page or in a column, and then sets the appropriate option for the current division.

```
stat = MessageBox ("Do you want to hyphenate the last word?", 36, "Example Script")
If stat = 6 Then ' user said yes
    .Division.DivisionOptions.HyphenationOptions.HyphenateLastWordInColumnOrPage =
True
Else
    .Division.DivisionOptions.HyphenationOptions.HyphenateLastWordInColumnOrPage =
False
End If
```


'Example: HyphenateLastWordInPara property

'This example asks the user whether to hyphenate the last word in a paragraph, and then sets the appropriate option for the current division.

```
stat = MessageBox ("Do you want to hyphenate the last word?", 36, "Example Script")
If stat = 6 Then ' user said yes
    .Division.DivisionOptions.HyphenationOptions.HyphenateLastWordInPara = True
Else
    .Division.DivisionOptions.HyphenationOptions.HyphenateLastWordInPara = False
End If
```

```
'Example: HyphenationOptions property
'This example sets hyphenation options for a new document.

.NewDocument
For i = 1 To 20 ' put some text in to demonstrate
    .type "testing hyphenation "
Next
' turn on auto hyphenation
.ActiveDocument.DivisionOptions.HyphenationOptions.AutoHyphenate = True
.Text.Attributes.NoHyphenate = False ' turn on local hyphenation
MessageBox "Click OK to disable hyphenation", MB_OK, "Example Script"
.ActiveDocument.DivisionOptions.HyphenationOptions.AutoHyphenate = False
```

'Example: HyphZoneAfter property

'This example sets options for hyphenation so that a minimum of six
'characters must be before the hyphen in a hyphenated word, and at least
'two characters must be carried to the following line.

```
.Division.DivisionOptions.HyphenationOptions.HyphZoneBefore = 6
```

```
.Division.DivisionOptions.HyphenationOptions.HyphZoneAfter = 2
```

'Example: HyphZoneBefore property

'This example sets options for hyphenation so that a minimum of six
'characters must be before the hyphen in a hyphenated word, and at least
'two characters must be carried to the following line.

```
.Division.DivisionOptions.HyphenationOptions.HyphZoneBefore = 6
```

```
.Division.DivisionOptions.HyphenationOptions.HyphZoneAfter = 2
```

```
'Example: IconBarPositionState property
Dim CR As String*1
Dim IcnPallet As String
Dim MsgStr As String
Dim IcnMgr As IconBarManager

IcnPallet = "Comment Tools"
CR = Chr(10)
Set IcnMgr = .ApplicationWindow.IconBarManager

With IcnMgr.IconBars(IcnPallet)
    MsgStr = "Height = " & .Height & CR
    MsgStr = MsgStr & "IconBarPositionState = " & .IconBarPositionState & CR
    MsgStr = MsgStr & "PositionType = " & .PositionType & CR
    MsgStr = MsgStr & "ScreenPositionX = " & .ScreenPositionX & CR
    MsgStr = MsgStr & "ScreenPositionY = " & .ScreenPositionY
    MessageBox MsgStr, 64, "Script Example - " & .Name
End With
```

```
'Example: IconBarSets property
'This example prints the name of all installed SmartIcons sets to the
'LotusScript output panel.

For i = 0 To .ApplicationWindow.IconBarManager.IconBarSets.Count -1
    Print.ApplicationWindow.IconBarManager.IconBarSets.item(i)
Next
```

'Example: IconBars property
'This example displays the names of all the SmartIcons sets currently
'displayed. It then accesses a SmartIcon set which is not currently
'displayed, and prints its name.

```
Forall Bars In .ApplicationWindow.IconBarManager.IconBars  
    Print Bars.Name  
End Forall
```

```
Dim MyBar As IconBar  
Set MyBar = .ApplicationWindow.IconBarManager.IconBars("Table cell")  
Print "The current bar is " & MyBar.IconSetName
```

'Example: GraphicType property

'This example asks the user whether to export graphics to HTML as JPEG images.

'If so, the option is set to JPEG, if not, the option is set to PNG.

```
stat = MessageBox ("Do you want to export graphics as JPEG?", 36, "Example Script")
```

```
If stat = 6 Then ' user said yes, set type to 0 for JPEG images
```

```
    .ApplicationWindow.UserInterfacePrefs.HTMLOptions.GraphicType = 0
```

```
Else ' user said no, set type to 2 for PNG images
```

```
    .ApplicationWindow.UserInterfacePrefs.HTMLOptions.GraphicType = 2
```

```
End If
```



```
'Example: HTMLOptions property
'This example asks the user whether to import HTML files as source code
'rather than HTML text, then sets the appropriate option.

stat = MessageBox ("Do you want to import HTML source code?", 36, "Example Script")
If stat = 6 Then ' user said yes
    .ApplicationWindow.UserInterfacePrefs.HTMLOptions.ImportAsSource = True
Else
    .ApplicationWindow.UserInterfacePrefs.HTMLOptions.ImportAsSource = False
End If
```

'Example: IconSize property

'This example asks the user whether to display large icons, and then sets the appropriate option.

```
stat = MessageBox ("Do you want to display large SmartIcons?", 36, "Example Script")
```

```
If stat = 6 Then ' user said yes
```

```
    .ApplicationWindow.IconBarManager.IconSize = $LwpIconSizeSupervga
```

```
Else
```

```
    .ApplicationWindow.IconBarManager.IconSize = $LwpIconSizeVga
```

```
End If
```

'Example: IgnoreSoftHyphens property

'This example asks the user whether to hyphenate at soft hyphens, and then sets the appropriate option.

```
stat = MessageBox ("Do you want to use soft hyphens?", 36, "Example Script")
If stat = 6 Then ' user said yes
    .Division.DivisionOptions.HyphenationOptions.IgnoreSoftHyphens = False
Else
    .Division.DivisionOptions.HyphenationOptions.IgnoreSoftHyphens = True
End If
```

```
'Example: ImageProcess method
'This example script imports a bitmap image, then displays the Image Processing
'dialog box.

' import a bmp file from the Word Pro directory
GPath = .path & "\helpbutn.bmp"
.ImportGraphic GPath, ".bmp", False, False, "Default Graphic/OLE"
MessageBox "Click OK to display the Image Processing dialog box.", MB_OK, "Example
Script"
.OleObject.ImageProcess
```

```
'Example: ImportAsSource property
'This example asks the user whether to import HTML files as source code
'rather than HTML text, then sets the appropriate option.

stat = MessageBox ("Do you want to import HTML source code?", 36, "Example Script")
If stat = 6 Then ' user said yes
    .ApplicationWindow.UserInterfacePrefs.HTMLOptions.ImportAsSource = True
Else
    .ApplicationWindow.UserInterfacePrefs.HTMLOptions.ImportAsSource = False
End If
```

```
'Example: ImportGraphic method
' This example inserts the 'Turtle' Word Pro Drawing graphic into the current
' document. The graphic is placed in a frame based upon the 'Default
' Graphic/OLE' frame style.
' RUNTIME DEPENDENCIES: You must have a document open and have installed the
' the Word Pro clipart for this script to work.
```

```
Dim FileName as String
FileName = .ApplicationWindow.UserInterfacePrefs.GraphicPath & "\Turtle.SDW"
```

```
.ImportGraphic FileName, ".SDW", False, False, "Default Graphic/OLE"
```

Note OLE is not supported under OS/2.

```
'Example: IncludeHeader property
'This example asks the user whether to include the document's footer in the
'HTML file, then sets the appropriate option.

stat = MessageBox ("Do you want to export the document's footer?", 36, "Example
Script")
If stat = 6 Then   ' user said yes
    .ApplicationWindow.UserInterfacePrefs.HTMLOptions.IncludeFooter = True
Else
    .ApplicationWindow.UserInterfacePrefs.HTMLOptions.IncludeFooter = False
End If
```

```
'Example: IncludeHeader property
'This example asks the user whether to include the document's header in the
'HTML file, then sets the appropriate option.

stat = MessageBox ("Do you want to export the document's header?", 36, "Example
Script")
If stat = 6 Then ' user said yes
    .ApplicationWindow.UserInterfacePrefs.HTMLOptions.IncludeHeader = True
Else
    .ApplicationWindow.UserInterfacePrefs.HTMLOptions.IncludeHeader = False
End If
```


'Example: IncludeInitialsInNotes property

'This example asks the user whether to display the user's initials in

'comment notes, and then sets the appropriate option.

```
stat = MessageBox ("Do you want to view initials in comment notes?", 36, "Example  
Script")
```

```
If stat = 6 Then ' user said yes
```

```
    .ActiveDocument.DocOptions.IncludeInitialsInNotes = True
```

```
Else
```

```
    .ActiveDocument.DocOptions.IncludeInitialsInNotes = False
```

```
End If
```

'Example: IndentFromLeft property

'This example sets options for the footnote separator line, including
'indenting the line 1 inch from the left margin.

.Division.FootnoteOptions.FootnoteSeparator.UseSeparatorLine = True

.Division.FootnoteOptions.FootnoteSeparator.IsFixedLength = False

.Division.FootnoteOptions.FootnoteSeparator.CustomLength = 9360

.Division.FootnoteOptions.FootnoteSeparator.**IndentFromLeft** = 1440

.Division.FootnoteOptions.FootnoteSeparator.SpaceAbove = 40

.Division.FootnoteOptions.FootnoteSeparator.SpaceBelow = 60

```
'Example: Indent property
'This example types some text, and then adjusts the indent options for the paragraph.

.NewDocument
For i = 1 To 20
    .type "Indention test "
Next
.Text.Indent.All = 1440 ' indent everything one inch
.Text.Indent.First = 720 ' indent the first line another 1/2 inch
.Text.Indent.Rest = -720 ' pull the 'rest of text' indent back 1/2 inch
.Text.Indent.Right = 720 ' and right indent everything 1/2 inch.
MessageBox "Click OK to revert to the style indents.", MB_OK, "Example Script"
.Text.Indent.RevertToStyle
```

```
'Example: InitFindAndReplace method
' This example inserts three identical sentences into the current document,
' clears the FindAndReplace settings, sets the FindString to "cat" and then
' displays a message box. When you click OK, the script finds the first 'cat'.
' RUNTIME DEPENDENCIES: You must have a document open for this script to work.

Dim SentenceCount As Integer
For SentenceCount = 1 To 3
    .Text.InsertText "The sleep-deprived fox jumped over the gravity-challenged cat."
    .Text.SplitParagraph
Next
.Application.ResetFindAndReplace
.Application.FindAndReplace.FindString = "cat"
.InitFindAndReplace True
MessageBox "Click OK to find the word 'cat'.",MB_OK,"Example Script"
.Find
```

'Example: InsertBreak method

' This example inserts three lines of text into the current document.

' Each line is part of the same paragraph.

' RUNTIME DEPENDENCIES: You must have a document open for this script to work.

.Text.InsertText "First"

.Text.InsertBreak \$LwpBreakTypeWord

.Text.InsertText "line of text on paragraph 1."

.Text.InsertBreak \$LwpBreakTypeLine

.Text.InsertText "Second line of text on paragraph 1."

.Text.InsertBreak \$LwpBreakTypePage

.Text.InsertText "Third line of text on paragraph 1."

```
'Example: InsertBullet method
' This example selects a bullet style in the WingDings font, and then creates several
lines of a bulleted list..
' RUNTIME DEPENDENCIES: You must have a document open for this script to work.

.InsertBullet "Wingdings", "1"
For i = 1 To 4
    .Type "This is a bulleted list item.[ENTER]"
Next
```

```
'Example: InsertClickHere method
' This example inserts a ClickHere block in the current document and assigns
' the return value (the ClickHere ID) to a variable (NewClickHereId).
' The script then stores the new ClickHere in a variable (NewClickHere) and
' inserts some text in it.
' The script then prints the names and values for all ClickHeres in the
' current division to the Script Editor Output panel.
' RUNTIME DEPENDENCIES: You must have a document open for this script to work.

Dim NewClickHereId as String
Dim NewClickHere as ClickHere

NewClickHereId = .InsertClickHere()

Set NewClickHere = .Division.Foundry.ClickHeres(NewClickHereId)
NewClickHere.Prompt.InsertText "Click Here to type text "
NewClickHere.InsertText "Some text for the click here."

Forall Clicks in .Division.Foundry.ClickHeres
    Print Clicks.Name &" -- " & Clicks.GetText($LWPGetObjectParagraph, False)
End Forall
```

'Example: InsertColumnBreak method

' This example inserts a column break in the current document.

' RUNTIME DEPENDENCIES: You must have a document open for this script to work.

.InsertColumnBreak


```
'Example: InsertDate method
'Insert the current date using a pre-set formula
  .InsertDate "Now() %Da"
'Insert the current time using a pre-set formula
  .InsertDate "Now() %T1"
'Insert the current date and time using a custom formula for mixed case letters
  .InsertDate "Now() %FLWeekday, M/D/YY, HH:mm ampm"
'Insert the current date and time using a custom formula for uppercase letters
  .InsertDate "Now() %FCWeekday, M/D/YY, HH:mm ampm"
'Insert the current date at the cursor location. Note the use of the short system date
format.
  .InsertDate "Now() %FLSystemShortDate"
'Insert the current time at the cursor location. Note the use of the system time
format.
  .InsertDate "Now() %FLSystemTime"
```

'Example: InsertDocInfo method

'

' This example inserts the filename, date created, file size and author name
' predefined document fields, as well as an end-user field named
' 'CustomField1'.

' RUNTIME DEPENDENCIES: You must have a document open and have created
' an end-user field named 'CustomField1' for this script to work.

.InsertDocInfo \$LwpDocVarFilename

.InsertDocInfo \$LwpDocVarDatecreated

.InsertDocInfo \$LwpDocVarDocsize

.InsertDocInfo \$LwpDocVarCreatedby

.InsertDocInfo \$LwpDocVarField, "CustomField1"

```
'Example: InsertDocument method
' This example uses an input box to get a file name from you and then inserts
' the contents of that document into the currently active document.
' The default file name is "README95.LWP".
' RUNTIME DEPENDENCIES: You must have a document open for this script to work.
```

```
Dim FileName as String
FileName = InputBox$ ("Enter a filename to insert, e.g., README95.LWP:", "Example
Script", "")
```

```
.InsertDocument FileName, "", "", False, True
```

```
'Example: InsertField method
' This example inserts several power fields into the current document.
' Each field is on a new line and is preceded by its description.
' RUNTIME DEPENDENCIES: You must have a document open for this script to work.

.Text.InsertText "Create date: "
.InsertField "CreateDate %DB"
.Text.SplitParagraph

.Text.InsertText "Edit date: "
.InsertField "EditDate %DB"
.Text.SplitParagraph

.Text.InsertText "Editing time: "
.InsertField "TotalEditingTime"
.Text.SplitParagraph

.Text.InsertText "Number of words: "
.InsertField "NumWords"
```

'Example: InsertFootnote method

' This example inserts a footnote into the current document. The footnote

' is placed at the bottom of the page and is numbered 1.

' RUNTIME DEPENDENCIES: You must have a document open for this script to work.

.Division.FootnoteOptions.FootnoteNumbering.StartingNumber = 1

.Division.FootnoteOptions.FootnoteNumbering.ResetWhen = LwpResetOptionEachPage

.InsertFootnote \$LwpFnTypeAtBottomOfPage

'Example: InsertHardSpace method

' This example inserts 2 words seperated by a hard space into the current
' document.

' RUNTIME DEPENDENCIES: You must have a document open for this script to work.

.Text.InsertText "Some"

.Text.InsertHardSpace

.Text.InsertText "text."

```
'Example: InsertHyperlink method
'This example inserts text to use as a Hyperlink,
'selects part of that text, and then creates the link
.Type "Visit our Web page at www.lotus.com"
.Type "[SHIFTLeft][SHIFTLeft][SHIFTLeft][SHIFTLeft][SHIFTLeft]"
.Type "[CTRLSHIFLeft][CTRLSHIFLeft][CTRLSHIFLeft]"
.InsertHyperlink "http://www.lotus.com", "", "", "", ""
.Type "[Esc]"

'Move the cursor down
.Type "[CTRLDown][Enter][Enter][Up]"

'This example inserts the same Hyperlink, but uses the
'LinkedText parameter to specify which text should be linked.
.Type "Visit our web page at "
.InsertHyperlink "http://www.lotus.com", "", "", "", "www.lotus.com"
```

'Example: InsertIndex method

' This example Inserts an index at the end of the current document in a

' separate division based on the default SmartMaster.

' RUNTIME DEPENDENCIES: You must have a document open for this script to work.

.InsertIndex

'Example: InsertionMode property
'This example enabled Insert mode, types some text, then enables overtype.
'The next typed text overwrites the first.

```
.Preferences.InsertionMode = True  
.Type "This test is inserted[Home]"  
.Preferences.InsertionMode = False  
.Type "This text is overstruck."
```

```
'Example: InsertNote method  
' This example inserts a comment note, adds some text to it and then deselects the  
note.  
' RUNTIME DEPENDENCIES: You must have a document open for this script to work.
```

```
.InsertNote  
.Text.InsertText "Some text for the note."  
.Type("[ESC]") ' close the Note
```

'Example: InsertNumber method

' This example insert the roman numeral 5 into the current document.

' RUNTIME DEPENDENCIES: You must have a document open for this script to work.

.Text.InsertNumber \$LwpNumberingStyleUppercaseroman , 0, 0, 5

```
'Example: InsertOleDivision method
' This example insert an OLE division based on the bitmap graphic file named
' TEST.BMP
' RUNTIME DEPENDENCIES: You must have a document open and a file named
' TEST.BMP located in the C:\ directory for this script to work.
```

```
Dim FilePath As String
Dim ClassID As String
```

```
ClassID = "{0003000A-0000-0000-C000-000000000046}"
FilePath = "C:\TEST.BMP"
.InsertOleDivision $LwpDivLocInsertAfterCurrentdiv,
$LwpOleActionCreateembedded,ClassID,FilePath
```

Note OLE is not supported under OS/2.

'Example: InsertOne method

' This example inserts a right align tab with leader dots one inch from the
' right margin.

' RUNTIME DEPENDENCIES: You must have a document open for this script to work.

.Text.TabRack.InsertOne 1444, \$LwpTabTypeRight , \$LwpTabLeaderDot ,
\$LwpTabRelativeRight , 32

'Example: InsertPageBreak method

' This example inserts a page break in the current document.

' RUNTIME DEPENDENCIES: You must have a document open for this script to work.

.InsertPageBreak

```
'Example: InsertPageLayout method
' This example inserts a new page layout based on the current page layout.
' RUNTIME DEPENDENCIES: You must have a document open for this script to work.

Dim StyleName as String
StyleName = .Page.Layout.Style.Name

.InsertPageLayout StyleName , 0, 0, 0
```

```
'Example: InsertPageNumber method
' This example inserts a page number on the current document in the
' following format: "Page 1 - Division Name".
' RUNTIME DEPENDENCIES: You must have a document open for this script to work.

' must include this script to do bitwise operations on constants.
%INCLUDE "WPBITMSK.LSS"

.InsertPageNumber $LwpNumberingStyleBasic, "Page ", " - ", 0, 1,
LwpPageNumberFlagsIncludeDivName or LwpPageNumberFlagsResetondivision
```


'Example: InsertPath method

' This example changes Word Pro's default document path.

' RUNTIME DEPENDENCIES: You must have a document open for this script to work.

.Application.ApplicationWindow.UserInterfacePrefs.InsertPath \$LwpSetDocumentsPath,
"C:\dev\lotus"

'Example: InsertRowOrColumn method

' This example creates a parallel column table with 3 columns. A new row is
' inserted.

' RUNTIME DEPENDENCIES: You must have a document open for this script to work.

.CreateParallelColumns 3, \$LtsAlignmentHorizCenter

.BaseTable.InsertRowOrColumn \$LwpTableInsTypeRow, True, 1

'Example: InsertSection method

' This example inserts a new section in the active division of the current
' document.

' RUNTIME DEPENDENCIES: You must have a document open for this script to work.
.InsertSection "Default Page", True, True, \$LwpStartTypeNextpage, False, True

```
'Example: InsertTab method  
' This example inserts 5 consecutive tabs.  
' RUNTIME DEPENDENCIES: You must have a document open for this script to work.
```

```
rc = .Text.InsertTab(5)
```

'Example: InsertText method

' This example inserts some text into the current document.

' RUNTIME DEPENDENCIES: You must have a document open for this script to work.

.Text.InsertText "Some text for the document."

'Example: InsertTOC method

' This example inserts a Table of Contents (TOC) at beginning of the active
' document in a separate division based on the default SmartMaster.
' RUNTIME DEPENDENCIES: You must have a document open for this script to work.

.InsertTOC \$LwpGenerateAcrossEntireDoc, \$LwpTOCPlacementBeginofdoc, True

'Example: InternalCopy method

' This example copies the selected text, creates a new line and then pastes
' the copied text on the new line.

' RUNTIME DEPENDENCIES: You must have a document open and text selected for
' this script to work.

.Text.InternalCopy(True)

.Text.MoveToEnd \$LwpLocationTypeLine

.Text.SplitParagraph

.Text.InternalPaste \$LwpFoundryTypeTemporary

'Example: InternalCut method

' This example cuts the selected text, creates a new line and then pastes
' the copied text on the new line.
' RUNTIME DEPENDENCIES: You must have a document open and text selected for
' this script to work.

```
.Text.InternalCut(True)  
.Text.MoveToEnd $LwpLocationTypeLine  
.Text.SplitParagraph  
.Text.InternalPaste $LwpFoundryTypeTemporary
```


'Example: InternalPaste method
' This example copies the selected text, creates a new line and then pastes
' the copied text on the new line.
' RUNTIME DEPENDENCIES: You must have a document open and text selected for
' this script to work.

```
.Text.InternalCopy(True)  
.Text.MoveToEnd $LwpLocationTypeLine  
.Text.SplitParagraph  
.Text.InternalPaste $LwpFoundryTypeTemporary
```

```

'Example: InvalidateButton method
' This example creates a new button to the status bar and then adds text to
' the button. The STATUSBARBUTTONOVERRIDE TEXT is then bound to the
' SetTheButtonText subroutine to set the button text during needs repainting.
' RUNTIME DEPENDENCIES: You must have a document open for this script to work.

Dim ButtonName As String
Dim NewButton As StatusBarButton
With .ApplicationWindow.StatusBar
    ButtonName = .CreateNewbutton (0, 0, 100, LwpButtonTypeText )
'create the new button
    With .StatusBarButtons(ButtonName)
'
        .SetOverrideText("New Button...")
        .InvalidateButton
        On Event STATUSBARBUTTONOVERRIDE TEXT From .StatusBarButtons(ButtonName) Call
SetTheButtonText
    End With
        .InvalidateWholeBar ' Force the bar to repaint
End With
End Sub

Sub SetTheButtonText (Source As StatusBarButton, ButtonName As String)
    'Add the the button text each time the status bar needs repainting.
    Source.SetOverrideText("New Button...")
End Sub

```

```

'Example: InvalidateWholeBar method
' This example creates a new button to the status bar and then adds text to
' the button. The STATUSBARBUTTONOVERRIDE TEXT is then bound to the
' SetTheButtonText subroutine to set the button text during needs repainting.
' RUNTIME DEPENDENCIES: You must have a document open for this script to work.

Dim ButtonName As String
Dim NewButton As StatusBarButton
With .ApplicationWindow.StatusBar
    ButtonName = .CreateNewbutton (0, 0, 100, LwpButtonTypeText )
'create the new button
    With .StatusBarButtons(ButtonName)
'        .SetOverRideText("New Button...")
        .InvalidateButton
        On Event STATUSBARBUTTONOVERRIDE TEXT From .StatusBarButtons(ButtonName) Call
SetTheButtonText
    End With
    .InvalidateWholeBar ' Force the bar to repaint
End With
End Sub

Sub SetTheButtonText (Source As StatusBarButton, ButtonName As String)
    'Add the the button text each time the status bar needs repainting.
    Source.SetOverRideText("New Button...")
End Sub

```

'Example: Invalidate method

' This example creates a bookmark based on the current marker name and then
' deletes the marker and repaints the active document to clear its bookmark
' identifier.
' RUNTIME DEPENDENCIES: You must have a document open for this script to work.

```
Dim MarkerName As String
```

```
MarkerName = .Mark($LwpMarkerTypeBookmark)
```

```
.Division.BookmarkManager.AddBookmark "ExampleBookmark", MarkerName
```

```
MessageBox "Click OK to delete the bookmark.",MB_OK,"Example Script"
```

```
rc =.Division.Foundry.Markers.Item(MarkerName).DeleteMarker()
```

```
rc = .ActiveDocWindow.Invalidate()
```

'Example: IsAsciiCRLF property

'This example asks the user how to treat return symbols in ASCII import and
'export, and then sets the appropriate option.

```
stat = MessageBox ("Do ASCII text files have carriage returns after each line?", 36,  
"Example Script")
```

```
If stat = 6 Then ' user said yes
```

```
    .ApplicationWindow.Filter.IsAsciiCRLF = True
```

```
Else
```

```
    .ApplicationWindow.Filter.IsAsciiCRLF = False
```

```
End If
```

```
'Example: IsAsciiKeepStyle property
'This example asks the user whether style names are included in ASCII
'text files, and then sets the appropriate option.

stat = MessageBox ("Are style names included in ASCII files?", 36, "Example Script")
If stat = 6 Then ' user said yes
    .ApplicationWindow.Filter.IsAsciiKeepStyle = 1
Else
    .ApplicationWindow.Filter.IsAsciiKeepStyle = 0
End If
```

```
'Example: IsAutoGrow property
'This example creates a table and asks the user whether the table rows should
'automatically adjust their height to the contents.
'It then sets the appropriate option.

.CreateTable
stat = MessageBox ("Do you want use automatic row sizing?", 36, "Example Script")
If stat = 6 Then ' user said yes
    .Table.IsAutoGrow = True
Else
    .Table.IsAutoGrow = False
End If
```

```
'Example: IsColumnBreakAfter property
'This example sets options for column breaks.
.Layout.NumCols = 3 ' go into a newspaper column format
.Type "This is some text in the column.[Enter]"
.Text.Breaks.IsColumnBreakBefore = True ' put a column break before the next paragraph
.Type "There is a column break before this paragraph.[Enter]"
.Text.Breaks.IsColumnBreakBefore = False
.Text.Breaks.IsColumnBreakAfter = True ' put a column break after the next paragraph
.Type "There is a column break after this paragraph.[Enter]"
.Text.Breaks.IsColumnBreakAfter = False
.Type "This is text in the third column."
```



```
'Example: IsColumnBreakBefore property
'This example sets options for column breaks.
.Layout.NumCols = 3 ' go into a newspaper column format
.Type "This is some text in the column.[Enter]"
.Text.Breaks.IsColumnBreakBefore = True ' put a column break before the next paragraph
.Type "There is a column break before this paragraph.[Enter]"
.Text.Breaks.IsColumnBreakBefore = False
.Text.Breaks.IsColumnBreakAfter = True ' put a column break after the next paragraph
.Type "There is a column break after this paragraph.[Enter]"
.Text.Breaks.IsColumnBreakAfter = False
.Type "This is text in the third column."
```

'Example: IsComplex property

'This example sets up an alternating (complex) page layout, then changes
'the left margin of the left page, and mirrors the change to the right page.

.Page.Layout.**IsComplex** = True

.Page.Layout.LeftPage.MarginLeft = 2880

.Page.Layout.LeftPage.MirrorPage

Word Pro: HidelconBars method

{button ,AL('H_ICONBARMANAGER_CLASS',0)} [See list of classes](#)

{button ,AL('H_HIDEICONBARS_METHOD_EXSCRIPT',1)} [See example](#)

Temporarily closes/hides all SmartIcons bars that are currently showing, until the context changes.

Syntax

[objectreference].HidelconBars()

Parameters**Return value**

The return value for this method will always be -1 or 0. When testing the return value, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

The user interface for closing icon bar objects is found in the list that displays when you click on the file drawer adjacent to an icon bar object.

Word Pro: HideStatusBar method

{button ,AL('H_STATUSBAR_CLASS',0)} [See list of classes](#)

{button ,AL('H_HIDESTATUSBAR_METHOD_EXSCRIPT',1)} [See example](#)

Hides the status bar.

Syntax

[objectreference].HideStatusBar()

Parameters**Return value**

The return values for this method will always be -1 and 0. When testing the return value, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Word Pro: HighlightToggle method

{button ,AL('H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL('H_HIGHLIGHTTOGGLE_METHOD_EXSCRIPT',1)} [See example](#)

Turns the Review & Comment tools highlighter on or off. Equivalent to clicking the Highlighter icon on the Review & Comment tools icon bar.

Syntax

[objectreference].HighlightToggle()

Parameters

None.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Word Pro: Hit method

{button ,AL(^H_DIVISION_CLASS;H_TEXTDOCUMENT_CLASS',0)} [See list of classes](#)

Syntax

[objectreference].Hit()

Parameters

Return value

Usage

Word Pro: HourGlass method

{button ,AL('H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL('H_HOURLASS_METHOD_EXSCRIPT',1)} [See example](#)

Allows you to show or hide the Windows hourglass cursor.

Syntax

[objectreference].HourGlass(Show)

Parameters

Show

A Numeric expression which allows you to specify whether or not you want the hourglass cursor to show. Data type is Integer. The legal values for this parameter are -1 and 0 but you may use the LotusScript constants True (-1) and False (0) instead of the integer values. Optional parameter. There is no default value for this parameter. A True value shows the hourglass.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Typically, you would use this method to hide the hourglass while your script is running.

Word Pro: ImportGraphic method

{button ,AL('H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL('H_IMPORTGRAPHIC_METHOD_EXSCRIPT',1)} [See example](#)

Imports a graphic into the current document. Unlike an OLE object, you cannot use the imported graphic to launch the application that created the graphic or edit the graphic in place. However, the graphic is updated if you change the original file and resave the Word Pro file that contains the imported graphic.

Note OLE is not supported under OS/2.

Syntax

[objectreference].ImportGraphic(FilePath, FileFormat, Link, ScratchOutFrame, [FrameStyle])

Parameters

FilePath

A String expression which specifies the directory path and name of the file which is the source of the imported graphic.

FileFormat

A String expression which specifies the file format for the graphic you are importing. The string expression for each file format is unique and registered with the operating system. The values listed in the table below were valid at the time of publication.

<u>If you are importing this type of graphic...</u>	<u>Use this value for the FileFormat parameter...</u>
---	---

Bitmap	.bmp
CGM	.cgm
Corel Draw	.cdr
Encapsulated Postscript	.eps
Freelance Graphics	.drw
GIF	.gif
HPGL	.plt
JPEG	.jpg
Kodak Photo CD	.pcd
Lotus PIC	.pic
PCX	.pcx
TIFF	.tif
Windows Metafile	.wmf
Word Perfect Graphic	.wpg
Word Pro Draw	.sdw
Word Pro Equation	.tex

Link

An Integer value of -1 or 0 indicating whether the imported graphic receives updates from the original (-1) or remains independent of the original (0). You can use the LotusScript constants of True (-1) and False (0) as the value for this parameter.

ScratchOutFrame

An Integer value, which indicates whether you want to draw the new graphic frame by hand or let Word Pro draw the frame based on a frame style. If you want to draw the frame yourself, use a True value (-1) for this parameter. If you want Word Pro to draw the frame based on an existing style, use a False value (0).

FrameStyle

A String expression, which specifies the frame style you want to use for the imported graphic's frame. Optional parameter. If the imported graphic is an equation and you do not specify a frame style, Word Pro uses the default equation frame style. All other imported graphics are placed in the default GraphicOle frame style, unless you specify another frame style using this parameter.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively. For example, if the graphic specified in the FilePath parameter does not exist, Word Pro returns 0.

Usage

Word Pro: ImportPicture method

{button ,AL('H_GRAPHICOLEBJECT_CLASS;H_GRAPHIC_CLASS;H_OLEOBJECT_CLASS',0)} [See list of classes](#)

Imports a graphic picture into the current document.

Syntax

[objectreference].ImportPicture(Path, FileFormat, Link)

Parameters

Path

Data type is String.

FileFormat

Data type is String.

Link

Data type is Integer. The legal values for this parameter are -1 and 0 but you may use the LotusScript constants True (-1) and False (0).

Return value

The return value for this method will always be -1 or 0. When testing the return value, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Word Pro: IndexAll method

{button ,AL('H_CLICKHERE_CLASS;H_TEXTMARKER_CLASS;H_TEXT_CLASS',0)} [See list of classes](#)

Syntax

[objectreference].IndexAll(Formula, ViceVersaFormula, MarkerName)

Parameters

Formula

Data type is String.

ViceVersaFormula

Data type is String.

MarkerName

Data type is String.

Return value

Integer as Boolean.

Usage

Word Pro: InitFindAndReplace method

{button ,AL('H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL('H_INITFINDANDREPLACE_METHOD_EXSCRIPT',1)} [See example](#)

Initializes the Find & Replace utility by clearing the registers which track the number of finds and replacements. This method also sets the options for the Find & Replace function.

Syntax

[objectreference].InitFindAndReplace(UseUserSettings)

Parameters

UseUserSettings

An Integer expression which allows you to choose between the options which were last set by the user (True) and the standard default options (False). Data type is Integer. The legal values for this parameter are -1 and 0 but you may use the LotusScript constants True (-1) and False (0) instead of the integer values. Optional parameter. Default is False (0).

Return value

None.

Usage

The default Find & Replace options in Word Pro are the options that appear the first time you perform a Find & Replace during any Word Pro session. If you change these options to perform a find or replace, Word Pro sees your new options as "user settings." Your user settings remain in effect until you change them again, at which time, your new settings take the place of your previous settings. These user settings are discarded each time you exit Word Pro.

If you provide a True value for the UseUserSettings parameter, you are telling Word Pro to use the last options you set for Find & Replace. A False value tells Word Pro to use the default Find & Replace settings, regardless of the options you may choose during the current session of Word Pro.

Word Pro: InsertBreak method

{button ,AL('H_CLICKHERE_CLASS;H_TEXTMARKER_CLASS;H_TEXT_CLASS',0)} [See list of classes](#)

{button ,AL('H_INSERTBREAK_METHOD_EXSCRIPT',1)} [See example](#)

Inserts a break in a ClickHere block, a TextMarker, or Text object.

Syntax

[objectreference].InsertBreak(BreakType)

Parameters

BreakType

Use this parameter to specify the type of break you want to insert. Data type is Variant, which allows the value of this parameter to be one of the constants listed below or its numeric equivalent (in parentheses). There is no default value.

<u>Value</u>	<u>Effect</u>
\$LwpBreakTypeColumn (73)	Inserts a column break in a multi-column page layout. Equivalent to choosing Text - Insert Other - Column Break.
\$LwpBreakTypeLine (74)	Inserts a soft line break. Equivalent to pressing [CTRL+SHIFT+RETURN].
\$LwpBreakTypePage (72)	Inserts a hard page break. Equivalent to pressing [CTRL+RETURN].
\$LwpBreakTypeWord (75)	Inserts a soft hyphen. Equivalent to pressing [CTRL+HYPHEN].

Return value

The return value for this method will always be -1 or 0. When testing the return value, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Word Pro: InsertBullet method

{button ,AL('H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL('H_INSERTBULLET_METHOD_EXSCRIPT',1)} [See example](#)

Starts a bulleted list with the character specified in the FontName and BulletChar parameters.

Syntax

[objectreference].InsertBullet(FontName, BulletChar)

Parameters

FontName

A String expression representing the name of the font from which you are getting the bullet character.

BulletChar

A String expression specifying the character used as the bullet.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Calling this method is equivalent to choosing a "Bullet style" on the Bullet and number panel of the Text InfoBox.

Word Pro: InsertClickHere method

{button ,AL(^H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL(^H_INSERTCLICKHERE_METHOD_EXSCRIPT',1)} [See example](#)

Inserts a ClickHere block in a document.

Syntax

[objectreference].InsertClickHere()

Parameters

None.

Return value

A String representing the name of the ClickHere object which was inserted.

Usage

Word Pro: InsertColumnBreak method

{button ,AL(^H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL(^H_INSERTCOLUMNBREAK_METHOD_EXSCRIPT',1)} [See example](#)

Inserts a column break in a document.

Syntax

[objectreference].InsertColumnBreak()

Parameters

None.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

A column break breaks a page, a parallel column, or a table across a page.

Word Pro: InsertDate method

{button ,AL('H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL('H_INSERTDATE_METHOD_EXSCRIPT',1)} [See example](#)

Inserts a date at the insertion point. You can specify the date format with the Date parameter. Equivalent to choosing Text - Insert Other - Date/Time.

Syntax

[objectreference].InsertDate(Date)

Parameters

Date

A String expression representing the date formula. A date formula specifies which date you want to insert as well as the format for the date. You can use one of the preset date or time formulas (the ones beginning with %D or %T), or you can create a custom formula (beginning with %FL for mixed case or %FC for all uppercase).

The table below contains four examples of date formulas. Note how the pre-defined formulas are shorter and easier to use, but the custom formulas give you precise control over every aspect of the date.

<u>This formula...</u>	<u>...yields this result</u>
"CreateDate %Da"	1/8/97
"CreateDate %T3"	09:01AM
"CreateDate %FLWeekday, M/D/YY, HH:mm ampm"	Saturday, 1/8/97, 09:01 AM
"CreateDate %FCWeekday, M/D/YY, HH:mm ampm"	SATURDAY, 1/8/97, 09:01 AM

Note Only one of the preset date formulas uses leading zeros. To get leading zeros on dates, you must create your own formula, using the "%FL" or "%FC."

All date formulas must begin with a reference to a specific date. You can specify one of the three dates listed below. Each date value includes both date and time information.

<u>This date...</u>	<u>...yields this:</u>
Now()	The current system date and time.
CreateDate	The date and time the document was first saved.
EditDate	The date and time the document was last saved.

After you specify a date, specify a formula to control how the date and time information displays. You can specify only one formula. If you want to display both the date and time information, you must use a custom formula. When you use a custom date formula (%FC or %FL), you can include as many or as few of the custom elements as you like, in any order you like, and with any additional text or punctuation you want.

Note All custom formula elements are case-sensitive. Using a different case than that shown in the table will yield unpredictable results.

All sample dates and times in the table below reflect the date Saturday, January 8, 1997 at 9:01:05 AM.

<u>Pre-set Formula</u>	<u>Equivalent Custom Formula</u>	<u>Sample</u>
%Da	%FLM/D/YY	1/8/97
%Db	Month D, YYYY	January 8, 1997
%DB	Month D, YYYY	JANUARY 8, 1997
%Dc	D Month YYYY	8 January 1997
%DC	D Month YYYY	8 JANUARY 1997
%Dd	Weekday, Month D, YYYY	Saturday, January 8, 1997
%DD	Weekday, Month D, YYYY	SATURDAY, JANUARY 8, 1997
%De	Month D	January 8
%DE	Month D	JANUARY 8
%Df	Weekday, Month D	Saturday, January 8
%DF	Weekday, Month D	SATURDAY, JANUARY 8

%Dg	M/D	1/8
%Dh	M/D/YYYY	1/8/1997
%Di	D. Month	8. January
%DI	D. Month	8. JANUARY
%Dj	D. Month YYYY	8. January 1997
%DJ	D. Month YYYY	8. JANUARY 1997
%Dk	YYYY Month D	1997 January 8
%DK	YYYY Month D	1997 JANUARY 8
%DI	Month, YYYY	January, 1997
%DL	Month, YYYY	JANUARY, 1997
%Dm	DD/MM/YYYY	08/01/1997
%T1	HH:mm	09:01 (leading zero on hour)
%T2	H:mm ampm	9:01AM
%T3	HH:mm ampm	09:01AM (leading zero on hour)
%T4	(none)	9:01A
%T5	(none)	09:01A (leading zero on hour)
%T6	H:mm ampm	9:01am
%T7	HH:mm ampm	09:01am (leading zero on hour)
%T8	(none)	9:01a
%T9	(none)	09:01a (leading zero on hour)

Custom formula element

	<u>Description</u>	<u>Sample</u>
%FL	Begins a mixed case date formula (after a specific date, such as CreateDate).	Saturday, January 1, 1997
%FC	Begins an uppercase date formula (after a specific date, such as CreateDate).	SATURDAY, JANUARY 1, 1997
SystemShortDate	Returns the date in the Short Date format which is specified in the Date panel of the Regional Settings control panel.	<i>(See the Date panel in your Regional Settings control panel)</i>
SystemLongDate	Returns the date in the Long Date format which is specified in the Date panel of the Regional Settings control panel.	<i>(See the Date panel in your Regional Settings control panel)</i>
ISODate1	Returns the date in the ISODate1 format.	1997/01/08
ISODate2	Returns the date in the ISODate2 format.	1997/01/08 09:01:05
Month	Returns the entire month name.	January
Mn	Returns an abbreviated month name.	Jan
Weekday	Returns the entire weekday name.	Saturday

Wday	Returns an abbreviated weekday name.	Sat
D	Returns the day digit.	8
DD	Returns the day digit with leading zero.	08
M	Returns the month digit.	1
MM	Returns the month digit with leading zero.	01
YY	Returns the abbreviated year.	97
YYYY	Returns the entire year.	1997
H	Returns the hour.	9
HH	Returns the hour with leading zero.	09
m	Returns the minutes.	1
mm	Returns the minutes with leading zero.	01
S	Returns the seconds.	5
SS	Returns the seconds with leading zero.	05
ampm	Uses the 12-hour clock and appends the AM/PM indicator.	AM

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Word Pro: InsertDocInfo method

```
{button ,AL(^H_WPAPPLICATION_CLASS;H_CLICKHERE_CLASS;H_TEXTMARKER_CLASS;H_TEXT_CLASS',0)}
```

[See list of classes](#)

```
{button ,AL(^H_INSERTDOCINFO_METHOD_EXSCRIPT',1)} See example
```

Inserts a document information field at the insertion point in the currently active document. Document information fields are listed on the Fields panel of the Document Properties dialog box (choose File - Document Properties - Document). To insert a field which you created, use \$LwpDocVarField for the Type parameter and the name of your field for the FieldName parameter.

Syntax

```
[objectreference].InsertDocInfo(Type,[FieldName])
```

Parameters

Type

The type of document information field you want to insert. Data type is Variant, which allows the value of this parameter to be one of the constants listed below or its numeric equivalent (in parentheses). There is no default value.

<u>Constant</u>	<u>Type of field inserted</u>
\$LwpDocVarAllversionnames (210)	The names of the different versions of the currently active document.
\$LwpDocVarCreatedby (2046)	The name of the user who created the currently active document.
\$LwpDocVarDatecreated (196)	The date the currently active document was created.
\$LwpDocVarDatelastrevison (197)	The date the currently active document was last revised.
\$LwpDocVarDescription (195)	The contents of the "Description" field for the currently active document.
\$LwpDocVarDivisionname (203)	The name of the active division in the currently active document.
\$LwpDocVarDoccategory (212)	The value of the "Document category" field.
\$LwpDocVarDocsize (202)	The size of the currently active document.
\$LwpDocVarField (191)	Use with the FieldName parameter to insert a custom DocField which you create.
\$LwpDocVarFilename (192)	The name of the currently active document.
\$LwpDocVarKeywords (215)	The keywords listed in the "Keywords" field for the currently active document.
\$LwpDocVarLasteditor (2047)	The initials of the last user who edited the currently active document.
\$LwpDocVarNumchars (201)	The number of characters stored in the currently active document.
\$LwpDocVarNumpages (199)	The number of pages in the currently active document.
\$LwpDocVarNumversions (209)	The number of versions of the currently active document.
\$LwpDocVarNumwords (200)	The number of words in the currently active document.
\$LwpDocVarOthereditors (2048)	The initials of all the users who edited the currently active document.
\$LwpDocVarOtherversioneditors (207)	The other editors for the currently active version of this document.
\$LwpDocVarPath (193)	The location of the currently active document.
\$LwpDocVarSectionname (204)	The name of currently active section. If the insertion

	point is not in a named section, there is no result.
\$LwpDocVarStylesheet (194)	The name of the SmartMaster used for the currently active document.
\$LwpDocVarTotaledittime (198)	The total amount of time that the currently active document has been open for editing.
\$LwpDocVarVersioncreatedate (206)	The date on which this version of the currently active document was created.
\$LwpDocVarVersioncreatedby (205)	The name of the user who created the currently active version of the active document.
\$LwpDocVarVersionlasteditdate (213)	The date on which the currently active version was last edited.
\$LwpDocVarVersionlasteditedby (214)	The name of the user who last edited the currently active version of the active document.
\$LwpDocVarVersionname (208)	The name of the currently active version of the active document.
\$LwpDocVarVersionnumrevisions (2049)	The number of revisions made to the currently active version of the active document.
\$LwpDocVarVersionremarks (211)	The contents of the "Version Remarks" field for the currently active document.

FieldName

An optional String expression representing the name of the custom DocField that you want to insert. To insert a custom DocField, you must use \$LwpDocVarField as the value for the Type parameter.

Return value

The return value for this method will always be -1 or 0. When testing the return value, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Word Pro: InsertDocument method

{button ,AL(^H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL(^H_INSERTDOCUMENT_METHOD_EXSCRIPT',1)} [See example](#)

Inserts a document in the Word Pro application.

Syntax

[objectreference].InsertDocument([FilePath,] [FileType,] [Password,] [AddToLastFileOpenList,] [Restore])

Parameters

FilePath

An optional String expression which specifies the name and location of the document you want to insert.

FileType

An optional String expression representing the file type of the document you want to insert.

Password

An optional String expression representing the password of the document you want to insert.

AddToLastFileOpenList

Allows you to add the inserted document to the list of recently opened files. This list appears in the File menu in Word Pro. Data type is Integer. The legal values for this parameter are -1 and 0 but you may use the LotusScript constants True (-1) and False (0) instead of the integer values. Optional parameter. Default is False.

Restore

Allows you to restore the original position of the insertion point. A True value places your insertion point at the beginning of the inserted document. A False leaves the insertion point at end of the inserted document. Data type is Integer. The legal values for this parameter are -1 and 0 but you may use the LotusScript constants True (-1) and False (0) instead of the integer values. Optional parameter. Default is True.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Word Pro: InsertField method

{button ,AL('H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL('H_INSERTFIELD_METHOD_EXSCRIPT',1)} [See example](#)

Inserts a Power Field in a document.

Syntax

[objectreference].InsertField(Formula, [AutoRun], [Lock], [AutoNew], [pfType])

Parameters

Formula

A String expression representing the Power Field instructions.

AutoRun

An Integer expression which allows you to specify whether or not the power field will run automatically or only when updated. Data type is Integer. The legal values for this parameter are -1 and 0 but you may use the LotusScript constants True (-1) and False (0) instead of the integer values. Optional parameter. Default is False (0), which means the power field must be updated manually.

Lock

An Integer expression which allows you to specify whether or not the power field value is locked. Data type is Integer. The legal values for this parameter are -1 and 0 but you may use the LotusScript constants True (-1) and False (0) instead of the integer values. Optional parameter. Default is False (0), which means the power field will not be locked.

AutoNew

An Integer expression which allows you to specify whether or not the power field value is converted to static text. This option allows you to insert a power field as part of a SmartMaster, so that each time a document is created from that SmartMaster, the value of the power field is converted to static text within the new document. Data type is Integer. The legal values for this parameter are -1 and 0 but you may use the LotusScript constants True (-1) and False (0) instead of the integer values. Optional parameter. Default is False (0), which means the power field remains a live element within your document.

PfType

Not used in Word Pro 97 or Word Pro for OS/2 Warp 4.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Word Pro: InsertFootnote method

{button ,AL('H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL('H_INSERTFOOTNOTE_METHOD_EXSCRIPT',1)} [See example](#)

Inserts a footnote at the insertion point in the currently active document.

Syntax

[objectreference].InsertFootnote([FootnoteType])

Parameters

FootnoteType

Use one of the values listed below to specify which type of footnote you want to insert. Data type is Variant, which allows the value of this parameter to be one of the string values listed below or its numeric equivalent (in parentheses). There is no default value.

<u>Value</u>	<u>Effect</u>
\$LwpFnTypeAnyposition (289)	The footnote flows with the footnote anchor.
\$LwpFnTypeAtBottomOfPage (290)	The footnote is at the bottom of the page.
\$LwpFnTypeAtEndOfDiv (293)	The footnote is at the end of the division which contains the footnote anchor.
\$LwpFnTypeAtEndOfDivisionSepDiv (294)	The footnote is in a separate division at the end of the division containing the footnote anchor.
\$LwpFnTypeAtEndOfDivisionGroup (295)	The footnote is at the end of the division group containing the footnote anchor.
\$LwpFnTypeAtEndOfDivGroupSepDiv (296)	The footnote is in a separate division at the end of the division group containing the footnote anchor.
\$LwpFnTypeAtEndOfDoc (291)	The footnote is at the end of the document containing the footnote anchor.
\$LwpFnTypeAtEndOfDocSepDiv (292)	The footnote is in a separate division at the end of the document containing the footnote anchor.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Word Pro: InsertHardSpace method

{button ,AL('H_CLICKHERE_CLASS;H_TEXTMARKER_CLASS;H_TEXT_CLASS',0)} [See list of classes](#)

{button ,AL('H_INSERTHARDSPACE_METHOD_EXSCRIPT',1)} [See example](#)

Inserts a non-breaking space at the insertion point.

Syntax

[objectreference].InsertHardSpace()

Parameters

None.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Word Pro: InsertHyperlink method

{button ,AL('H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL('H_INSERTHYPERLINK_METHOD_EXSCRIPT',1)} [See example](#)

Inserts a Hyperlink at the insertion point using the specified attributes.

Syntax

[objectreference].InsertHyperlink(URL, Anchor, ALTag, MarkerName[, LinkedText])

Parameters

URL

A string value which specifies the destination of the Hyperlink. This string is usually a URL, but could also be any other valid HTML link code such as and IP address, a "mailto:" command, or an FTP address.

Anchor

ALTag

MarkerName

LinkedText

A string value which specifies the text you want to appear in the document. This is the "clickable" text seen in the browser. This parameter is optional. If you do not provide a value for this parameter, Word Pro uses the current selection as the linked text.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Word Pro: InsertIndex method

{button ,AL('H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL('H_INSERTINDEX_METHOD_EXSCRIPT',1)} [See example](#)

Inserts an index in the currently active document. By default, this method inserts an index at the end of the document in a separate division, based on the default SmartMaster (default.mwp). The default index derives its entries from the entire document.

Syntax

[objectreference].InsertIndex([IndexGeneration,] [IndexLocation,] [UseSeparateDivision,] [SmartMasterName])

Parameters

IndexGeneration

Specifies the scope of the new index in terms of where the index will look for its index entries. Data type is Variant, which allows the value of this parameter to be one of the string values listed below or its numeric equivalent (in parentheses). Default is \$LwpGenerateAcrossEntireDoc, which generates an index for the entire document.

<u>Value</u>	<u>Effect</u>
\$LwpGenerateAcrossCurrentDiv (414)	Generates an index using entries found in the current division.
\$LwpGenerateAcrossCurrentSect (415)	Generates an index using entries found in the current section.
\$LwpGenerateAcrossEntireDoc (412)	Generates an index using all entries in the currently active document.
\$LwpGenerateAcrossGroupedDivs (413)	Generates an index using entries found in the currently active group of divisions.
\$LwpGenerateAcrossSelectedText (416)	Generates an index using entries found in the current selection.

IndexLocation

Data type is Variant, which allows the value of this parameter to be one of the string values listed below or its numeric equivalent (in parentheses). Default is \$LwpIndexLocationEndofdoc, which places the new index at the end of the document.

<u>Value</u>	<u>Effect</u>
\$LwpIndexLocationEndofdivision (418)	Places the new index at the end of the currently active division
\$LwpIndexLocationEndofdoc (417)	Places the new index at the end of the currently active document.
\$LwpIndexLocationEndofgroup (419)	Places the new index at the end of the currently active group of divisions.
\$LwpIndexLocationEndofsection (421)	Places the new index at the end of the currently active section.
\$LwpIndexLocationInsertionpoint (420)	Places the new index at the insertion point.

UseSeparateDivision

A Numeric expression which allows you to place the new index in a separate division. Data type is Integer. The legal values for this parameter are -1 and 0 but you may use the LotusScript constants True (-1) and False (0) instead of the integer values. Optional parameter. Default is True, which places the new index in its own division.

SmartMasterName

An optional String expression representing the name of the SmartMaster used for the index division.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Word Pro: InsertLink method

{button ,AL('H_DDELINKMANAGER_CLASS',0)} [See list of classes](#)

Inserts a Dde link into a document.

Syntax

[objectreference].InsertLink(MarkerName, Server, Topic, Item)

Parameters

MarkerName

A String expression representing the internal name of the marker designating the link. You must create the marker before using this method. Required parameter.

Server

A String expression representing the executable name of the server to which you want to link.

Topic

A String expression representing the name of the drive, directory, and name of the file that contains the data or the object name of the data.

Item

A String expression representing the name for the data, such as a range of cells, a named spreadsheet range, or a bookmark name.

Return value

The return value for this method will always be -1 or 0. When testing the return value, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

This method always uses a rich text format. Update data only value is always False (0). If you want to insert a link, the [AddDdeLink method](#) is recommended instead of this method, because the additional parameters found in the AddDdeLink method give you more control.

Word Pro: InsertMarker method

{button ,AL(^H_MARKER_CLASS;H_POWERFIELD_CLASS;H_CLICKHERE_CLASS;H_TEXTMARKER_CLASS;H_RUBYMARKER_CLASS;H_TABLEMARKER_CLASS',0)} [See list of classes](#)

Inserts the specified content object into a marker.

Syntax

[objectreference].InsertMarker(MarkerName)

Parameters

MarkerName

A String expression representing the name of the content object you want to insert.

Return value

The return value for this method will always be -1 or 0. When testing the return value, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Word Pro: InsertNote method

{button ,AL('H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL('H_INSERTNOTE_METHOD_EXSCRIPT',1)} [See example](#)

Inserts a note at the insertion point. The focus is left inside the note so you can start typing. Equivalent to choosing Create - Comment Note.

Syntax

[objectreference].InsertNote()

Parameters

None.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Word Pro: InsertNumber method

{button ,AL(^H_CLICKHERE_CLASS;H_TEXTMARKER_CLASS;H_TEXT_CLASS',0)} [See list of classes](#)

{button ,AL(^H_INSERTNUMBER_METHOD_EXSCRIPT',1)} [See example](#)

Inserts a placeholder for a paragraph's number and allows you to set the numbering style used when displaying that number.

Note A paragraph must be assigned to a numbering sequence in order for Word Pro to assign a paragraph number. Therefore, if you use this method in a paragraph which is not part of a numbering sequence, the value displayed in the placeholder is zero (0).

Syntax

[objectreference].InsertNumber(NumberingStyle, Char, Level, [Start])

Parameters

NumberingStyle

Use this parameter to specify a numbering style for the paragraph number. Data type is Variant, which allows the value of this parameter to be one of the constants listed below or its numeric equivalent (in parentheses). There is no default value.

<u>Value</u>	<u>Effect</u>
\$LwpNumberingStyleBasic (1588)	Displays the paragraph number in the Arabic style (1, 2, 3, 4, 5, and so on).
\$LwpNumberingStyleLeadingzero (2233)	Displays the paragraph number in the Arabic style with leading zeros in the tens place (01, 02, 03, 04, 05, and so on).
\$LwpNumberingStyleLowercaseletters (1590)	Displays the paragraph number as a lowercase letter (a, b, c, d, and so on).
\$LwpNumberingStyleUppercaseletters (1589)	Displays the paragraph number as an uppercase letter (A, B, C, D, and so on).
\$LwpNumberingStyleLowercaseroman (1592)	Displays the paragraph number as a lowercase Roman numeral (i, ii, iii, iv, and so on).
\$LwpNumberingStyleUppercaseroman (1591)	Displays the paragraph number as an uppercase Roman numeral (I, II, III, IV, and so on).
\$LwpNumberingStyleChar (1593)	Displays X of the characters you specify in the Char parameter, where X = paragraph number. For example, if you specify asterisk (ANSI code 42) as the character and the paragraph number is 4, the placeholder you insert with this method displays the paragraph number as four asterisks (****).

Char

The value of this parameter should be Zero (0), unless you use \$LwpNumberingStyleChar as the value for the NumberingStyle parameter. In that case, use this parameter to specify the numeric code for the character which you want to use when displaying the paragraph number. The numeric code you use should be derived from the character set defined by your operating system.

For example, if you are using the US English version of Word Pro and Windows 95, you should get your code from the ANSI 1252 code page. The character you specify in this parameter is displayed X number of times in the inserted placeholder, where X is equal to the paragraph number. If you specify asterisk (ANSI code 42) as the character and the paragraph number is 4, the placeholder you insert with this method displays the paragraph number as four

asterisks (****).

Level

An Integer indicating the position from which you want to pull the paragraph number. Because most numbering styles display only one number for each paragraph, the value of this parameter is usually 1. However, some numbering styles use multiple digits with a decimal notation, such as 1.1.1, 1.1.2, 1.1.3. In this kind of numbering style, the Level parameter enables you to specify which digit of the paragraph number you want to display in the placeholder.

For example, the paragraphs below belong to a multi-digit numbering style called Engineering Numbering (Type 3).

1.2.3 In this paragraph a Level value of 1 returns 1, 2 returns 2, and 3 returns 3.

3.2.1 In this paragraph a Level value of 1 returns 3, 2 returns 2, and 3 returns 1.

Note You can assign this Engineering Numbering style from the Custom Numbering dialog box. Select one or more paragraphs, choose Text - Bullets & Numbers and click the Custom button under the "Number style" option. In the Custom Numbering dialog box, click the Outline tab and specify "Engineering Numbering" as the Outline numbering sequence option. Finally, specify the third list item as the "Position and numbering type" option and click OK.

Start

This parameter allows you to change the starting (or base) value of the paragraph counter used in calculating the value for the placeholder. The paragraph counter usually has a base value of zero (0) and counts upward in increments of 1, so the first paragraph is numbered 1, the second paragraph is numbered 2, and so on. But you can change this base value by specifying a different integer for this parameter. Keep in mind that any change in the base value affects only the way the paragraph number is displayed in this placeholder. The actual paragraph number and any other placeholders you insert are unaffected by this change.

For example, if you insert a placeholder in a paragraph where the paragraph number is 3, that placeholder uses 3 as the paragraph number and displays 3 in the numbering style you specified, such as 3, 03, c, C, iii, III, ###. However, if you insert another placeholder and change the start value to 2, Word Pro counts the paragraphs starting at 2, resulting in a paragraph number of 5 which is displayed in the style you specify, such as 5, 05, e, E, v, V, #####.

In essence, if you specify a value other than 0 for this parameter, that value is added to the paragraph number and the resulting sum is displayed in the numbering style you specify. If you change the Start value and the resulting value is less than 0, no paragraph number is displayed.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

As stated above, this method inserts a placeholder for the paragraph number. This placeholder acts as a type of window for seeing the paragraph number. The paragraph number itself is not affected by how many of these placeholders you insert. Therefore, if you insert several of these placeholders in the same paragraph, they all display the same number, though they may display that number using different numbering styles or different starting values.

For example, all three paragraphs below share the same style: Numbered list. The Numbered list style was assigned to an outline numbering sequence, so the numbering is turned on and the actual paragraph numbers are displayed to the left of each paragraph. The InsertNumber method was used to insert seven placeholders at the end of each paragraph. Each placeholder was assigned a different numbering style but all the placeholders have a Start value of zero(0).

1. This is the first paragraph with the Numbered list style: 1, 01, a, A, i, I, #
2. This is the second paragraph with the Numbered list style: 2, 02, b, B, ii, II, ##
3. This is the third paragraph with the Numbered list style: 3, 03, c, C, iii, III, ###

The code used to insert the seven placeholders looks like this:

```
.Text.InsertNumber 1588, 0, 1  
.Text.InsertNumber 2233, 0, 1  
.Text.InsertNumber 1590, 0, 1  
.Text.InsertNumber 1589, 0, 1  
.Text.InsertNumber 1592, 0, 1  
.Text.InsertNumber 1591, 0, 1  
.Text.InsertNumber 1593, 35, 1
```

The difference in the numbering styles was achieved by the different values in the NumberingStyle parameter. The

difference in the paragraph number displayed was caused by the number of paragraphs which shared the Numbered list style.

Now look at the same three paragraphs below. Again they all share the same Numbered list style and each paragraph contains 7 placeholders.

1. This is the first paragraph with the Numbered list style: 1, 02, c, D, v, VI, #####
2. This is the second paragraph with the Numbered list style: 2, 03, d, D, vi, VII, #####
3. This is the third paragraph with the Numbered list style: 3, 04, e, F, vii, VIII, #####

The code used to insert the seven placeholders looks like this:

```
.Text.InsertNumber 1588, 0, 1, 0  
.Text.InsertNumber 2233, 0, 1, 1  
.Text.InsertNumber 1590, 0, 1, 2  
.Text.InsertNumber 1589, 0, 1, 3  
.Text.InsertNumber 1592, 0, 1, 4  
.Text.InsertNumber 1591, 0, 1, 5  
.Text.InsertNumber 1593, 35, 1, 6
```

Notice that the Start parameter was used to change the way the paragraph number displays. but the paragraph number itself remains the same.

Word Pro: InsertOleDivision method

{button ,AL('H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL('H_INSERTOLEDIVISION_METHOD_EXSCRIPT',1)} [See example](#)

Inserts an OLE division in a Word Pro document. Equivalent to choosing Create - Division and clicking Create OLE Division.

Note This method is not available in OS/2

Syntax

[objectreference].InsertOleDivision(DivisionLocation, OleAction, ClassName, Path[, Parent][,Neighbor])

Parameters

DivisionLocation

Specifies where the OLE division is inserted. Data type is Variant, so you can use either one of the string values or its numeric equivalent (in parentheses) as the value of this parameter.

<u>Value</u>	<u>Effect</u>
\$LwpDivLoInsertAfterCurrentdiv (185)	Inserts the OLE division before the current division.
\$LwpDivLoInsertAtInsertionPt (186)	Inserts the OLE division at the insertion point, splitting the current division.
\$LwpDivLoInsertBeforeCurrentdiv (184)	Inserts the OLE division after the current division.

OleAction

Allows you to specify how you want to create the OLE division contents. Data type is Variant, so you can use one of the string expressions listed below or its numeric equivalent (in parentheses) as the value of this parameter.

<u>Value</u>	<u>Effect</u>
\$LwpOleActionClipboardembedded (1604)	Creates an embedded OLE object using the current contents of the Clipboard.
\$LwpOleActionClipboardlink (1605)	Creates a linked OLE object using the current contents of the Clipboard.
\$LwpOleActionCreateembedded (1603)	Creates an embedded OLE object using the file you specify in the Path parameter.
\$LwpOleActionCreatelink (1602)	Creates a linked OLE object using the file you specify in the Path parameter.
\$LwpOleActionCreatenew (1601)	Creates a new OLE object using the server application you specify in the ClassName parameter.

ClassName

A String expression which specifies the type of OLE object you are creating. You must provide this information so Word Pro knows how to create the OLE object. The type of object is expressed as the ClassID or ProgID for the application which creates that type of object. (The application used to create an OLE object is often referred to as the server application.) For example, a Lotus Freelance Presentation has a ClassID of {CF746000-94FB-101B-8C12-02608C454BFF} and a ProgID of FLW3Presentation.

Here are the server application IDs for other SmartSuite application objects:

1-2-3 Worksheet

ClassID = {00045295-0000-0000-C000-000000000046}

ProgID = 123Worksheet

Launches 1-2-3 and opens an untitled worksheet.

Lotus Approach Report

ClassID = {00028703-0000-0000-C000-000000000046}

ProgID = ApproachReport

Launches Approach and prompts you to select an existing database from which to create the report. Once the

database is open, the Report Assistant opens and waits for you to create the report.

Lotus Approach Application

ClassID = {00028701-0000-0000-C000-000000000046}

ProgID = ApproachApplication

Launches Approach and prompts you to select an existing database.

Lotus Freelance Presentation

ClassID = {CF746000-94FB-101B-8C12-02608C454BFF}

ProgID = FLW3Presentation

Launches Freelance and prompts you with the New Presentation dialog box.

Lotus Freelance Drawing

ClassID = {CF746001-94FB-101B-8C12-02608C454BFF}

ProgID = FLW3Drawing

Launches Freelance and opens a new presentation with one blank page.

Lotus ScreenCam Movie 2.1

ClassID = {00041920-0000-0000-C000-000000000046}

ProgID = ScreenCamMovie2

Launches ScreenCam 2.1 and displays the ScreenCam control panel for you to start a recording.

You can find the ClassIDs and ProgIDs for other server applications in the Windows Registry for Windows 3.1 and Windows 95.

Path

A String expression which specifies the name and path of the source file you want to use for the OLE division. Use this parameter when you use \$LwpOleActionCreateembedded (1603) or \$LwpOleActionCreatelink (1602) as the value for the OleAction parameter. If you use any other value for OleAction, you must use a Null string ("") as the value for Path.

Parent

A String expression which allows you to specify the name of the division which you want to be a parent for the OLE division. If you include this parameter, Word Pro inserts the OLE division as a child division to the division you name here.

Neighbor

A String expression which allows you to specify the name of the division which you want to be a neighbor for the OLE division. If you include this parameter, Word Pro inserts the OLE division next to the division you name here.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

An OLE division is seen by Word Pro as a part of the Word Pro document. However, when you click the division tab for an OLE division, Word Pro launches the server application for the division's source file.

Word Pro: InsertOne method

{button ,AL(^H_TABRACK_CLASS',0)} [See list of classes](#)

{button ,AL(^H_INSERTONE_METHOD_EXSCRIPT',1)} [See example](#)

Inserts a tab in a document and allows you to indicate the following tab properties: position, tab type, tab leader, and relative tab type.

Syntax

[objectreference].InsertOne(Position, TabType, LeaderType, RelativeType, AlignChar)

Parameters

Position

Data type is Long. Position of the tab measured in Twips.

TabType

Data type is Variant; type is one of the four types listed below. You can use the string or its code as the value of this parameter.

<u>Value</u>	<u>Effect</u>
\$LwpTabTypeCenter (1864)	Inserts a center tab.
\$LwpTabTypeLeft (1863)	Inserts a left tab.
\$LwpTabTypeNumeric (1866)	Inserts a numeric tab.
\$LwpTabTypeRight (1865)	Inserts a right tab.

LeaderType

Data type is Variant; type is one of the three types listed below, or none. You can use the string or its code as the value of this parameter.

<u>Value</u>	<u>Effect</u>
\$LwpTabLeaderDot (1857)	Inserts a dot leader.
\$LwpTabLeaderHyphen (1856)	Inserts a hyphen leader.
\$LwpTabLeaderLine (1858)	Inserts a line leader.
\$LwpTabLeaderNone (1855)	Does not insert a leader.

RelativeType

Data type is Variant; type is one of the three types listed below. You can use the string or its code as the value of this parameter.

<u>Value</u>	<u>Effect</u>
\$LwpTabRelativeCenter (1862)	Inserts a relative center tab.
\$LwpTabRelativeLeft (1860)	Inserts a relative left tab.
\$LwpTabRelativeRight (1861)	Inserts a relative right tab.

AlignChar

A numeric expression representing the tab alignment character. You must use an Integer as the numeric expression. Use the ASC function to get the numeric value of a character. For example, to use a period as an alignment character, use ASC(".").

Return value

Usage

Word Pro: InsertPageBreak method

{button ,AL(^H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL(^H_INSERTPAGEBREAK_METHOD_EXSCRIPT',1)} [See example](#)

Inserts a page break in the document.

Syntax

[objectreference].InsertPageBreak()

Parameters

None.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Word Pro: InsertPageLayout method

{button ,AL('H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL('H_INSERTPAGELAYOUT_METHOD_EXSCRIPT',1)} [See example](#)

Inserts a page layout at the insertion point. Use the LayoutStyleName parameter to specify the page style and the StartType parameter to specify where the new page begins.

Syntax

[objectreference].InsertPageLayout(LayoutStyleName, [UsePrevHeaderText,] [UsePrevFooterText,] [StartType])

Parameters

LayoutStyleName

A String expression representing the name of the page style you want to use for the new page layout.

UsePrevHeaderText

A Numeric expression indicating whether or not you want the new page to use previous header text. Data type is Integer. The legal values for this parameter are -1 and 0 but you may use the LotusScript constants True (-1) and False (0) instead of the integer values. Optional parameter. Default is False, which does not use the header text from the previous page layout.

UsePrevFooterText

A Numeric expression indicating whether or not you want the new page to use previous footer text. Data type is Integer. The legal values for this parameter are -1 and 0 but you may use the LotusScript constants True (-1) and False (0) instead of the integer values. Optional parameter. Default is False, which does not use the footer text from the previous page layout.

StartType

Specifies where you want to start the new page layout. Data type is Variant, which allows the value of this parameter to be one of the string values listed below or its numeric equivalent (in parentheses). Default is \$LwpStartTypeThispage.

<u>Value</u>	<u>Effect</u>
\$LwpStartTypeNextevenpage (1827)	Starts the new page layout on the next even-numbered page.
\$LwpStartTypeNextoddpag (1826)	Starts the new page layout on the next odd-numbered page.
\$LwpStartTypeNextpage (1824)	Starts the new page layout on the next page.
\$LwpStartTypeThispage (1825)	Starts the new page layout on the current page.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Word Pro: InsertPageNumber method

{button ,AL('H_WPAPPLICATION_CLASS;H_CLICKHERE_CLASS;H_TEXTMARKER_CLASS;H_TEXT_CLASS',0)}

[See list of classes](#)

{button ,AL('H_INSERTPAGENUMBER_METHOD_EXSCRIPT',1)} [See example](#)

Inserts a page number at the insertion point, using the numbering style and accompanying text you specify.

Syntax

When called from the WPAApplication object:

[objectreference].InsertPageNumber([NumberingStyle,] [BeforeText,] [AfterText,] [StartingNumber,] [StartOnPage,] [Flags])

When called from a Text object:

[objectreference].InsertPageNumber(NumberingStyle, BeforeText, AfterText, StartingNumber, StartOnPage, Flags)

Parameters

NumberingStyle

Specifies the numbering style for the page number you are inserting. Data type is Variant, which allows the value of this parameter to be one of the constants listed below or its numeric equivalent (in parentheses). There is no default value when called from a Text, TextMarker, or ClickHere object. When called from the WPAApplication object, this is an optional parameter with a default of \$LwpNumberingStyleBasic.

<u>Value</u>	<u>Effect</u>
\$LwpNumberingStyleBasic (1588)	Displays the page number in Arabic style (1, 2, 3, 4, 5, and so on).
\$LwpNumberingStyleLeadingzero (2233)	Displays the page number in Arabic style with leading zeros in the tens place (01, 02, 03, 04, 05, and so on).
\$LwpNumberingStyleLowercaseletters (1590)	Displays the page number as a lowercase letter (a, b, c, d, and so on).
\$LwpNumberingStyleUppercaseletters (1589)	Displays the page number as an uppercase letter (A, B, C, D, and so on).
\$LwpNumberingStyleLowercaseroman (1592)	Displays the page number as a lowercase Roman numeral (i, ii, iii, iv, and so on).
\$LwpNumberingStyleUppercaseroman (1591)	Displays the page number as an uppercase Roman numeral (I, II, III, IV, and so on).
\$LwpNumberingStyleChar (1593)	Displays X of the characters you specify in the Char parameter where X = page number. For example, if you specify asterisk (ANSI code 42) as the character and the page number is 4, the placeholder you insert with this method displays the page number as four asterisks (****).

BeforeText

The text you want Word Pro to place before the page number. Data type is String. When called from the WPAApplication object, this is an optional parameter.

AfterText

The text you want Word Pro to place after the page number. Data type is String. When called from the WPAApplication object, this is an optional parameter.

StartingNumber

An Integer which specifies the starting page number. For example, if you use 5 as the value for this parameter, Word Pro uses 5 as the first page number, regardless of the page on which you place the first page number. When called from the WPAApplication object, this is an optional parameter with a default of 0.

StartOnPage

An Integer which specifies the page on which the starting page number appears. For example, if you use 1 as the value for this parameter, Word Pro places the starting page number (specified in StartingNumber parameter) on the first page in your document. When called from the WPAApplication object, this is an optional parameter with a default value of 1.

Flags

Data type is Variant, which allows the value of this parameter to be one of the constants listed below or its hexadecimal equivalent (in parentheses). You can combine these values when you want Word Pro to combine the features listed below. Use the OR operator to combine values. When called from the WPAApplication object, this is an optional parameter with a default of LwpPageNumberFlagsDefault.

Value	Effect
LwpPageNumberFlagsDefault (&H0)	Starts counting pages from the page designated in StartOnPage.
LwpPageNumberFlagsIncludebefore (&H4)	Counts all pages in the entire document.
LwpPageNumberFlagsIncludedivname (&H2)	Inserts the division name with the page number.
LwpPageNumberFlagsIncludeseaname (&H1)	Inserts the section name with the page number.
LwpPageNumberFlagsResetondivision (&H20)	Resets the page numbers at the beginning of each division.
LwpPageNumberFlagsResetonsection (&H10)	Resets the page numbers at the beginning of each section.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Call this method from the WPAApplication object when you want to insert page numbers into the currently active document. If you accept all the default values for the parameters, Word Pro starts the page numbers with 1 on the first page of your document and no text before or after the page number. The page numbers continue throughout the document.

Call this method from the Text object when you want to insert page numbers into a specific Text object in a specific place.

This method appears on TextMarker and ClickHere objects because these objects inherit the method from the Marker class. Do not call this method from a TextMarker or ClickHere object.

Word Pro: InsertPath method

{button ,AL('H_USERINTERFACEPREFS_CLASS',0)} [See list of classes](#)

{button ,AL('H_INSERTPATH_METHOD_EXSCRIPT',1)} [See example](#)

Inserts path or file information in the Word Pro Preferences dialog box.

Syntax

[objectreference].InsertPath(PathSelection, Path, [Path2],[Path3],[Path4],[Path5])

Parameters

PathSelection

Indicates which path or file you are setting. Data type is Variant. The value of this parameter must be one of the strings below or its numeric equivalent (in parentheses).

<u>Value</u>	<u>Effect</u>
\$LwpSetDocumentsPath (2081)	Inserts path information into the "Documents" field on the Locations panel of the Word Pro Preferences dialog box.
\$LwpSetSmartmasterPath (2082)	Inserts path information into the "SmartMaster" field on the Locations panel of the Word Pro Preferences dialog box.
\$LwpSetScriptPath (2083)	Inserts path information into the "Scripts" field on the Locations panel of the Word Pro Preferences dialog box.
\$LwpSetSmarticonPath (2084)	Inserts path information into the "SmartIcons" field on the Locations panel of the Word Pro Preferences dialog box.
\$LwpSetBackupPath (2085)	Inserts path information into the "Backups" field on the Locations panel of the Word Pro Preferences dialog box.
\$LwpSetUserdictPath (2086)	Inserts path information into the "User dictionaries" field on the Locations panel of the Word Pro Preferences dialog box.
\$LwpSetUserdictFile (2087)	Inserts filename information into the "Default user dictionary(s)" field on the Default files panel of the Word Pro Preferences dialog box.
\$LwpSetGlossaryPath (2088)	Inserts path information into the "Glossaries" field on the Locations panel of the Word Pro Preferences dialog box.
\$LwpSetGlossaryFile (2089)	Inserts file name information into the "Default glossary file(s)" field on the Default files panel of the Word Pro Preferences dialog box.
\$LwpSetGraphicPath (2544)	Inserts path information into the "Graphics" field on the Locations panel of the Word Pro Preferences dialog box.

Path

A String expression representing the file or path information to be included in the specified field.

Path2

An optional String expression representing additional file or path information to be included in the specified field.

Path3

An optional String expression representing additional file or path information to be included in the specified field.

Path4

An optional String expression representing additional file or path information to be included in the specified field.

Path5

An optional String expression representing additional file or path information to be included in the specified field.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Word Pro: InsertRowOrColumn method

{button ,AL(`H_BASetable_CLASS;H_FOOTNOTetable_CLASS;H_PARALLELCOLUMNS_CLASS;H_GLOSSARY_CLASS;H_TABLEHEADING_CLASS;H_TABLE_CLASS',0)} [See list of classes](#)

{button ,AL(`H_INSERTROWORCOLUMN_METHOD_EXSCRIPT',1)} [See example](#)

Inserts a new row or column into a table.

Syntax

[objectreference].InsertRowOrColumn(TableInsType, InsertAfter, NumToInsert, [Position,] [CopyCellStyle])

Parameters

TableInsType

The value of this Variant parameter must be one of the strings below or its numeric equivalent.

<u>Value</u>	<u>Effect</u>
\$LwpTableInsTypeRow (1875)	Indicates that the method should insert rows.
\$LwpTableInsTypeColumn (1876)	Indicates that the method should insert columns.

InsertAfter

A Boolean expression that controls whether the new row or column is inserted after the row or column specified in the Position parameter. If no value is specified in the Position parameter, the rows or columns are inserted in relation to the current selection.

NumToInsert

An Integer value which specifies the number of rows or columns to insert.

Position

An optional Integer value which specifies the ID of the row or column next to which new items should be inserted. If the table is in the current context, the default parameter value is the current row or column ID. If you are accessing a table that is not in the current context, the Position parameter will default to the last row or column ID.

CopyCellStyle

An optional Boolean parameter which indicates whether or not to copy the cell style from the row or column specified in the Position parameter. If no value is specified for the Position parameter and you are accessing the table in the current context, the cell style of the row or column in the current selection is used. If you are accessing a table that is not in the current context, the cell style of the last row or column of the table is used. The default value of this parameter is True.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

If your cursor is within a table, you can access that table in the current context by referring to the Table property of the WPAApplication class.

If your cursor is not within a table, you can access the tables in the current division by accessing the tableonlycollection object of the division's foundry. For example, use the following code to insert a new row to the end of all tables in a division:

```
Forall mtab in .Division.Foundry.Tables
    mtab.InsertRowOrColumn $LwpTableInsTypeRow, True, 1
End Forall
```

Word Pro: InsertRuby method

{button ,AL(^H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

This method is implemented only in the Asian-language versions of Word Pro.

Word Pro: InsertSection method

{button ,AL('H_WPAPPLICATION_CLASS;H_SECTIONTABS_CLASS',0)} [See list of classes](#)

{button ,AL('H_INSERTSECTION_METHOD_EXSCRIPT',1)} [See example](#)

Inserts a new section marker at the insertion point in the currently active division of a document. When called from a SectionTabs object, this method accepts no parameters.

Syntax

From the WPAApplication object:

[objectreference].InsertSection([StyleName,][UsePrevHeaderText,] [UsePrevFooterText,] [StartType,] [IsCreateIndex,] [ShowTab])

From a SectionTabs object:

[objectreference].InsertSection()

Parameters

StyleName

A String expression which specifies the name of the page style you want to use for the new section. Optional parameter. If you do not provide a page style name, Word Pro uses the page style on the currently active page.

UsePrevHeaderText

The legal values for this optional Integer parameter are -1 and 0 but you may use the LotusScript constants True (-1) and False (0).

A value of True indicates that the method should use the header text from the previous section.

A value of False indicates that the method should start a new header for the inserted section.

The default value for this parameter is False.

UsePrevFooterText

The legal values for this optional Integer parameter are -1 and 0 but you may use the LotusScript constants True (-1) and False (0).

A value of True indicates that the method should use the footer text from the previous section.

A value of False indicates that the method should start a new footer for the inserted section.

The default value for this parameter is False.

StartType

An optional parameter which allows you to specify how the new section begins. Default is \$LwpStartTypeThispage.

Data type is Variant, which allows you to use one of the string values below or its numeric equivalent (in parentheses).

<u>Value</u>	<u>Effect</u>
\$LwpStartTypeNextevenpage (1827)	Starts the new section on the next even-numbered page.
\$LwpStartTypeNextoddpag (1826)	Starts the new section on the next odd-numbered page.
\$LwpStartTypeNextpage (1824)	Starts the new section on the next page.
\$LwpStartTypeThispage (1825)	Starts the new section on the currently active page.

IsCreateIndex

Allows you to indicate whether the new section is an index section or a normal section. Data type is Integer. The legal values for this parameter are -1 and 0 but you may use the LotusScript constants True (-1) and False (0). Optional parameter. Default is False.

ShowTab

Allows you to show or hide the new section's tab. A value of True shows the tab, while a value of False hides the tab.

Data type is Integer. The legal values for this parameter are -1 and 0 but you may use the LotusScript constants True (-1) and False (0). Optional parameter. Default is True.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Word Pro: InsertSpecialTab method

{button ,AL('H_CLICKHERE_CLASS;H_TEXTMARKER_CLASS;H_TEXT_CLASS',0)} [See list of classes](#)

Inserts a tab mark with the alignment, leader, and other attributes you specify. The attributes for the tab are local only, meaning they have no effect on the tab settings in the horizontal ruler or the paragraph style.

Syntax

[objectreference].InsertSpecialTab([p1,] [TabType,] [LeaderType,] [RelativeType,] [AlignChar])

Parameters

Position

Specifies the position of the tab in Twips. There are 1440 Twips per inch. Data type is Long.

TabType

Specifies the type of tab alignment you want the tab mark to use. This, in turn, determines how the text which follows the tab mark is aligned. Data type is Variant, which allows the value of this parameter to be one of the constants listed below or its numeric equivalent (in parentheses). There is no default value, unless you call this method from WPAApplication, in which case default is \$LwpTabTypeLeft.

<u>Value</u>	<u>Effect</u>
\$LwpTabTypeLeft (1863)	Inserts a left-aligned tab.
\$LwpTabTypeRight (1865)	Inserts a right-aligned tab.
\$LwpTabTypeCenter (1864)	Inserts a centered tab.
\$LwpTabTypeNumeric (1866)	Inserts a decimal-aligned tab. If you want the text to align to a character other than a decimal, specify that character in the AlignChar parameter.

LeaderType

Specifies the type of leader used by the tab you are inserting. Data type is Variant, which allows the value of this parameter to be one of the constants listed below or its numeric equivalent (in parentheses). There is no default value, unless you call this method from WPAApplication, in which case default is \$LwpTabLeaderNone.

<u>Value</u>	<u>Effect</u>
\$LwpTabLeaderDot (1857)	Uses periods as the leader.
\$LwpTabLeaderHyphen (1856)	Uses hyphens as the leader.
\$LwpTabLeaderLine (1858)	Uses a line as the leader.
\$LwpTabLeaderNone (1855)	No leader.

RelativeType

Specifies the alignment of the tab mark in relation to the page. The default value places the tab mark at the left edge of the page, plus the number of Twips you specify in the Position parameter. Data type is Variant, which allows the value of this parameter to be one of the constants listed below or its numeric equivalent (in parentheses). There is no default value, unless you call this method from WPAApplication, in which case default is \$LwpTabTypeLeft.

<u>Value</u>	<u>Effect</u>
\$LwpTabRelativeLeft (1860)	Aligns the tab mark with the left edge of the page and then offsets it by the number of Twips

	specified in the Position parameter.
\$LwpTabRelativeRight (1861)	Aligns the tab mark with the right edge of the page and then offsets it by the number of Twips specified in the Position parameter.
\$LwpTabRelativeCenter (1862)	Aligns the tab mark with the center of the page and then offsets it by the number of Twips specified in the Position parameter.

AlignChar

An Integer expression which specifies the code for an ASCII character. If you specify a character in this parameter and that character is present in the text which follows this tab mark, the text aligns with the specified character at the tab mark. Optional parameter.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Word Pro: InsertTab method

{button ,AL('H_CLICKHERE_CLASS;H_TEXTMARKER_CLASS;H_TEXT_CLASS',0)} [See list of classes](#)

{button ,AL('H_INSERTTAB_METHOD_EXSCRIPT',1)} [See example](#)

Inserts the specified number of tab marks at the insertion point.

Syntax

[objectreference].InsertTab([count])

Parameters

Count

An Integer expression which specifies the number of tab marks you want to insert. Optional parameter. Default is 1.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Word Pro: InsertText method

{button ,AL('H_CLICKHERE_CLASS;H_TEXTMARKER_CLASS;H_TEXT_CLASS',0)} [See list of classes](#)

{button ,AL('H_INSERTTEXT_METHOD_EXSCRIPT',1)} [See example](#)

Inserts the specified text at the insertion point.

Syntax

[objectreference].InsertText(Text, [Split,] [TextType,])

Parameters

Text

A String expression which represents the text you want to insert in the document.

Split

An Integer which allows you to specify whether or not you want Word Pro to create a new paragraph at the end of the inserted text. Data type is Integer. The legal values for this parameter are -1 and 0, but you may use the LotusScript constants True (-1) and False (0) instead of the integer values. Optional parameter. Default is False (0), which means Word Pro will not add a paragraph mark at the end of the inserted text.

TextType

An optional parameter which allows you to specify the keyboard layout or ANSI code page from which the inserted text should be drawn. Value of this parameter should always be \$LwpTextTypeRoman when writing scripts for US English users. Data type is Variant, which allows the value of this parameter to be one of the constants listed below or its numeric equivalent (in parentheses). Default value is \$LwpTextTypeRoman.

<u>Value</u>	<u>Effect</u>
\$LwpTextTypeRoman (1937)	Use when writing for US English users.
\$LwpTextTypeUnicode (1936)	Use when writing scripts for systems which use Unicode.
\$LwpTextTypeRawUnicode (1940)	Inserts the text as Unicode unless the text can be converted to the native Windows code page.
\$LwpTextTypeHkatakana (1939)	Use for Asian-language versions of Word Pro; represents single-byte Japanese text.
\$LwpTextTypeKanji (1938)	Use for Asian-language versions of Word Pro; represents double-byte kanji Japanese text.
\$LwpTextTypeNative (2014)	Use if you want Word Pro to attempt to figure out the native language of the user.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

In order for this method to work, the object must have the focus.

Word Pro: InsertTOC method

{button ,AL('H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL('H_INSERTTOC_METHOD_EXSCRIPT',1)} [See example](#)

Inserts a Table of Contents (TOC) in the currently active document. By default, this method inserts a TOC at the beginning of the document in a separate division, based on the default SmartMaster (default.mwp). The default TOC derives its entries from the entire document.

Syntax

[objectreference].InsertTOC([TOCGeneration,] [TOCPlacement,] [UseSeparateDivision,] [SmartMasterName])

Parameters

TOCGeneration

Specifies the scope of the new TOC in terms of where the TOC will look for its TOC entries. Data type is Variant, which allows the value of this parameter to be one of the constants listed below or its numeric equivalent (in parentheses). Default is \$LwpGenerateAcrossEntireDoc, which generates a TOC for the entire document.

<u>Value</u>	<u>Effect</u>
\$LwpGenerateAcrossCurrentDiv (414)	Generates a TOC using entries found in the current division.
\$LwpGenerateAcrossCurrentSect (415)	Generates a TOC using entries found in the current section.
\$LwpGenerateAcrossEntireDoc (412)	Generates a TOC using all entries in the currently active document.
\$LwpGenerateAcrossGroupedDivs (413)	Generates a TOC using entries found in the currently active group of divisions.
\$LwpGenerateAcrossSelectedText (416)	Generates a TOC using entries found in the current selection.

TOCPlacement

Data type is Variant. Optional parameter. Default is \$LwpTOCPlacementBeginofdoc. The value of this parameter must be one of the strings below or its code equivalent.

<u>Value</u>	<u>Effect</u>
\$LwpTOCPlacementBeginofdivision (1842)	Places the new TOC at the beginning of the currently active division.
\$LwpTOCPlacementBeginofdoc (1841)	Places the new TOC at the beginning of the currently active document.
\$LwpTOCPlacementBeginofgroup (1843)	Places the new TOC at the beginning of the currently active group of divisions.
\$LwpTOCPlacementBeginofsection (1845)	Places the new TOC at the beginning of the currently active section.
\$LwpTOCPlacementInsertionpoint (1844)	Places the new TOC at the insertion point in the currently active document.

UseSeparateDivision

A Numeric expression which allows you to place the new TOC in a separate division. Data type is Integer. The legal values for this parameter are -1 and 0 but you may use the LotusScript constants True (-1) and False (0) instead of the integer values. Optional parameter. Default is True, which places the new TOC in its own division.

SmartMasterName

An optional String expression representing the name of the SmartMaster used for the TOC division.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Word Pro: InternalCopy method

```
{button ,AL('H_WPAPPLICATION_CLASS;H_CLICKHERE_CLASS;H_TEXTMARKER_CLASS;H_TEXT_CLASS',0)}
```

[See list of classes](#)

```
{button ,AL('H_INTERNALCOPY_METHOD_EXSCRIPT',1)} See example
```

Copies the selected items into either the AppFoundry or the TempFoundry property on WPAApplication. Unlike the Copy command found on the Edit menu, the selected items are not placed in the external Windows Clipboard.

Syntax

```
[objectreference].InternalCopy(Temporary)
```

Parameters

Temporary

An Integer which indicates whether or not the selected items should be copied into the temporary Foundry object found in the TempFoundry property. If the value of this parameter is -1 (True), the selected items are copied to TempFoundry. If the value is 0 (False), then the selected items are copied to the application Foundry object located in the AppFoundry property on WPAApplication. Optional parameter. Default is False (0), which copies the selected items to the AppFoundry property.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

The Foundry object located in the AppFoundry property is the usual place where things are stored during the copy process. Word Pro automatically clears the contents of AppFoundry and TempFoundry each time you use the InternalCopy method.

Word Pro: InternalCut method

```
{button ,AL('H_WPAPPLICATION_CLASS;H_CLICKHERE_CLASS;H_TEXTMARKER_CLASS;H_TEXT_CLASS',0)}
```

[See list of classes](#)

```
{button ,AL('H_INTERNALCUT_METHOD_EXSCRIPT',1)} See example
```

Deletes the selected items from the document and places them in the AppFoundry or TempFoundry property on WPAApplication. Unlike the Cut command found on the Edit menu, the selected items are not placed in the external Windows Clipboard.

Syntax

[objectreference].InternalCut(Temporary)

Parameters

Temporary

An Integer which indicates whether or not the cut items should be placed the temporary Foundry object found in the TempFoundry property. If the value of this parameter is -1 (True), the cut items are placed in TempFoundry. If the value is 0 (False), then the cut items are placed in the application Foundry object located in the AppFoundry property on WPAApplication. Optional parameter. Default is False (0), which places the cut items in the AppFoundry property.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

The Foundry object located in the AppFoundry property is the usual place where things are stored during the copy process. Word Pro automatically clears the contents of AppFoundry and TempFoundry each time you use the InternalCut method.

Word Pro: InternalPaste method

```
{button ,AL('H_WPAPPLICATION_CLASS;H_CLICKHERE_CLASS;H_TEXTMARKER_CLASS;H_TEXT_CLASS',0)}
```

[See list of classes](#)

```
{button ,AL('H_INTERNALPASTE_METHOD_EXSCRIPT',1)} See example
```

Pastes items from the specified Foundry object into the active document. Unlike the Paste command found on the Edit menu, the pasted items are drawn from the Foundry object, not the external Windows Clipboard.

Note To paste items from the external Windows clipboard into a document, use the [Paste](#) method found on WPAApplication.

Syntax

```
[objectreference].InternalPaste(FoundryType)
```

Parameters

FoundryType

Specifies the Foundry object used as the source for the paste operation. Data type is Variant, which allows the value of this parameter to be one of the constants listed below or its numeric equivalent (in parentheses).

<u>Value</u>	<u>Effect</u>
\$LwpFoundryTypeApplication (346)	Specifies the AppFoundry property on WPAApplication as the source for the paste.
\$LwpFoundryTypeDocument (345)	Specifies the Foundry property on Division as the source for the paste.
\$LwpFoundryTypeTemporary (347)	Specifies the TempFoundry property on WPAApplication as the source for the paste.

When you call InternalPaste from WPAApplication, there is no default value. You must provide one of the values listed above. However, when you call InternalPaste from a ClickHere, Text, or TextMarker object, this parameter uses the default value of \$LwpFoundryTypeApplication. You can override the default if you want.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Word Pro: InvalidateButton method

{button ,AL('H_STATUSBARBUTTON_CLASS',0)} [See list of classes](#)

{button ,AL('H_INVALIDATEBUTTON_METHOD_EXSCRIPT',1)} [See example](#)

Cause the status bar button to be repainted (refreshed). Forces the repaint by processing the written routine.

Syntax

[objectreference].InvalidateButton()

Parameters

Return value

Integer; always True.

Usage

You can change or update the button. When the button is invalidated, an event is emitted. If it is a text button (LwpButtonTypeText), the StatusBarButtonOverrideText event occurs. If it is a graphic button (LwpButtonTypeGraphics), the StatusBarButtonOverrideGraphic event occurs. If it is a text and graphic button, the StatusBarButtonOverrideTextAndGraphic event occurs.

If you do not respond to the events for a custom button that is invalidated, it will be blank.

Note If the text on the status bar button is never going to change, you can use the LwpButtonNoTextFromHost (&H800) parameter when the button is created.

Word Pro: InvalidateWholeBar method

{button ,AL('H_STATUSBAR_CLASS',0)} [See list of classes](#)

{button ,AL('H_INVALIDATEWHOLEBAR_METHOD_EXSCRIPT',1)} [See example](#)

Repaints (refreshes) the entire status bar.

Syntax

[objectreference].InvalidateWholeBar()

Parameters**Return value**

The return values for this method will always be -1 and 0. When testing the return value, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Use this method after you add or delete a button on the status bar to invalidate or refresh the bar so that the new addition or deletion can display.

Word Pro: Invalidate method

{button ,AL('H_DOCWINDOW_CLASS',0)} [See list of classes](#)

{button ,AL('H_INVALIDATE_METHOD_EXSCRIPT',1)} [See example](#)

Syntax

[objectreference].Invalidate()

Parameters**Return value****Usage**

Word Pro: IsDataNameUsed method

{button ,AL(^H_SCRIPTDATASET_CLASS;H_WPDATASET_CLASS',0)} [See list of classes](#)

Determines if a particular data name is being used for an object.

Syntax

[objectreference].IsDataNameUsed(DataName)

Parameters

DataName

A variable name in the data set. DataName is a String expression.

Return value

Integer. This method returns True (-1) if the the value specified in the DataName parameter is used as a data name. This method returns False (0) if the value specified in the DataName parameter is not used as a data name.

Usage

Word Pro: IsEmpty method

```
{button ,AL('H_BASECOLLECTION_CLASS;H_BAGCOLLECTION_CLASS;H_BOOKMARKCOLLECTION_CLASS;  
H_CELLCOLLECTION_CLASS;H_CELLLAYOUTCOLLECTION_CLASS;H_CHARACTERSTYLECOLLECTION_  
CLASS;H_CLICKHERECOLLECTION_CLASS;H_CONNECTEDLAYOUTCOLLECTION_CLASS;H_CONTENTC  
OLLECTION_CLASS;H_DDELINKCOLLECTION_CLASS;H_DIVISIONCOLLECTION_CLASS;H_DOCUMENTS_  
CLASS;H_EDITORCOLLECTION_CLASS;H_ENDNOTELAYOUTCOLLECTION_CLASS;H_DOCINFOFIELDCOL  
LECTION_CLASS;H_POWERFIELDCOLLECTION_CLASS;H_FOOTERLAYOUTCOLLECTION_CLASS;H_FOO  
TNOTECOLLECTION_CLASS;H_FOOTNOTELAYOUTCOLLECTION_CLASS;H_FRAMELAYOUTCOLLECTION_  
_CLASS;H_GLOSSARYCOLLECTION_CLASS;H_GRAPHICCOLLECTION_CLASS;H_GRAPHICOLEBJECTC  
OLLECTION_CLASS;H_GROUPLAYOUTCOLLECTION_CLASS;H_HEADERLAYOUTCOLLECTION_CLASS;H_I  
CONBARCOLLECTION_CLASS;H_LAYOUTCOLLECTION_CLASS;H_MARKERCOLLECTION_CLASS;H_MEN  
UITEMCOLLECTION_CLASS;H_NOTELAYOUTCOLLECTION_CLASS;H_OLEOBJECTCOLLECTION_CLASS;H_  
_OUTLINESEQCOLLECTION_CLASS;H_OUTLINESEQITEMCOLLECTION_CLASS;H_PAGELAYOUTCOLLECT  
ION_CLASS;H_PARAGRAPHSTYLECOLLECTION_CLASS;H_PARALLELCOLSCOLLECTION_CLASS;H_ROW  
LAYOUTCOLLECTION_CLASS;H_RUBYLAYOUTCOLLECTION_CLASS;H_SECTIONCOLLECTION_CLASS;H_  
SILVERBULLETCOLLECTION_CLASS;H_SMARTCORRECTCOLLECTION_CLASS;H_SMARTFILLCOLLECTIO  
N_CLASS;H_STATUSBARBUTTONCOLLECTION_CLASS;H_STRINGCOLLECTION_CLASS;H_SUPERTABLEC  
OLLECTION_CLASS;H_SUPERTABLELAYOUTCOLLECTION_CLASS;H_TABLECOLLECTION_CLASS;H_TABL  
EHEADINGCOLLECTION_CLASS;H_TABLEHEADINGLAYOUTCOLLECTION_CLASS;H_TABLELAYOUTCOLLE  
CTION_CLASS;H_TABLEMARKERCOLLECTION_CLASS;H_TABLEONLYCOLLECTION_CLASS;H_TEXTCOLL  
ECTION_CLASS;H_TEXTMARKERCOLLECTION_CLASS;H_TEXTSTYLECOLLECTION_CLASS;H_VERSIONC  
OLLECTION_CLASS;H_DOCWINDOWCOLLECTION_CLASS;H_WPDATASETCOLLECTION_CLASS;H_DROP  
CAPLAYOUTCOLLECTION_CLASS;H_CHARTDATAPOINTS_CLASS;H_CHARTLEGENDENTRIES_CLASS;H_  
CHARTMAJORGRIDLINES_CLASS;H_CHARTPIES_CLASS;H_CHARTPIESLICEGROUPS_CLASS;H_CHARTP  
IESLICES_CLASS;H_CHARTPIETITLES_CLASS;H_CHARTSERIESCOLLECTION_CLASS;H_CHARTTABLES  
ERIESCOLLECTION_CLASS;H_CHARTTEXTENTRIES_CLASS;H_CHARTTEXTLABELS_CLASS;')0}} See list of  
classes
```

Indicates whether or not a collection contains any items.

Syntax

```
[objectreference].IsEmpty()
```

Parameters

Return value

This method returns a long value of 1 or 0 indicating whether or not the specified collection object contains items.

Usage

Use this method to determine whether or not a specific collection object currently contains any items. If the method returns 1, then the collection object does not contain items. If the method returns 0, then the collection object does contain items.

Collections and layout objects

Collections that store user created layout objects also store default layout objects. A default layout object is represented by a style in Word Pro. For example, when you look at the Style panel in the InfoBox, you will see a Default Frame style for frame objects, a Default Cell style for cell objects, and so on.

Because default layout objects (styles) are stored in collections with user-created layout objects, the IsEmpty method of certain collections will always return 0, indicating that the collection always contains items.

To determine whether objects contained within a collection are default layout objects or user-created layout objects, refer to the [IsStyle](#) property of each collection item.

Word Pro: IsExportedAsNotesFX method

{button ,AL('H_DOCINFO_CLASS',0)} [See list of classes](#)

{button ,AL('H_ISEXPORTEDASNOTESFX_METHOD_EXSCRIPT',1)} [See example](#)

Indicates whether or not the specified DocInfo field is exported as a Notes FX field.

Syntax

[objectreference].IsExportedAsNotesFX(Type)

Parameters

Type

You can return one of the Variant data types below to determine if it is a DocInfo field exported as a NotesFX file. The value of this parameter must be one of the strings below or its numeric equivalent.

<u>Value</u>	<u>Effect</u>
\$LwpDocVarAllversionnames (210)	The names of all versions of a document.
\$LwpDocVarCreatedby (2046)	The Word Pro user name of the person who first saved the document.
\$LwpDocVarDatecreated (196)	The date and time the document was first created.
\$LwpDocVarDatelastrevision (197)	The last time the document was opened, edited, and saved.
\$LwpDocVarDescription (195)	Displays a description of the document.
\$LwpDocVarDivisionname (203)	The name of a division within the document.
\$LwpDocVarDoccategory (212)	The assigned category of the division or document.
\$LwpDocVarDocsize (202)	The size of the document in kilobytes.
\$LwpDocVarField (191)	Specifies the DocInfo field you want to export as Notes FX.
\$LwpDocVarFilename (192)	The file name you specified in the Save As dialog box when you saved the document.
\$LwpDocVarKeywords (215)	Displays the assigned keywords for the document.
\$LwpDocVarLasteditor (2047)	The initials of the editor who last saved the document.
\$LwpDocVarNone (190)	No DocInfo fields are set or reset as Notes FX.
\$LwpDocVarNumchars (201)	The number of characters in the document.
\$LwpDocVarNumpages (199)	The number of pages in the document.
\$LwpDocVarNumversions (209)	The number of versions in the document.
\$LwpDocVarNumwords (200)	The number of words in the document.
\$LwpDocVarOthereditors (2048)	The Word Pro user names of other people who saved the document.
\$LwpDocVarOtherversionededitors (207)	The Word Pro user names of other people who saved a version of the document.
\$LwpDocVarPath (193)	The drive and folder where the document is located.
\$LwpDocVarSectionname (204)	The name of a section in the document.
\$LwpDocVarStylesheet (194)	The style sheet used in the document.
\$LwpDocVarTotaledittime (198)	The total number of minutes the document was open.
\$LwpDocVarVersioncreatedate (206)	The date and time the version was first created.
\$LwpDocVarVersioncreatedby (205)	The Word Pro user name of the person who first saved the version of the document.
\$LwpDocVarVersionlasteditdate (213)	The date and time that the version was last saved.

\$LwpDocVarVersionlasteditedby (214)	The initials of the editor who last saved the version.
\$LwpDocVarVersionname (208)	The name of the current version.
\$LwpDocVarVersionnumrevisions (2049)	The number of times the version was opened, edited, and saved.
\$LwpDocVarVersionremarks (211)	Any remarks written by an editor to be reviewed by other assigned editors.

Return value

This method returns a value of 1 or 0 indicating that the method succeeded or failed respectively.

Usage

Word Pro: IsFilterTypePresent method

{button ,AL('H_FILTER_CLASS',0)} [See list of classes](#)

{button ,AL('H_ISFILTERTYPEPRESENT_METHOD_EXSCRIPT',1)} [See example](#)

Determine whether or not the specified filter resides in Word Pro.

Syntax

[objectreference].IsFilterTypePresent(Type, Import)

Parameters

Type

A String expression.

Import

A Numeric expression. You must use an Integer as the numeric expression.

Return value

Usage

Word Pro: IsMarkerEqualToSelection method

{button ,AL(^H_CLICKHERE_CLASS;H_TEXTMARKER_CLASS;H_TEXT_CLASS',0)} [See list of classes](#)

Determines whether the content of a marked range is the same as the content of the current selection.

Syntax

[objectreference].IsMarkerEqualToSelection(Marker)

Parameters

Marker

A String expression which specifies the name of the marked range you are checking.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Use this method to confirm that all of a marked range has been selected.

Word Pro: IsPointWithin method

{button ,AL(^H_BASECONTAINER_CLASS;H_CELLCONTAINER_CLASS;H_FRAMECONTAINER_CLASS;H_NOTECONTAINER_CLASS;H_RUBYCONTAINER_CLASS;H_DROPDOWNCONTAINER_CLASS;H_PAGECONTAINER_CLASS;H_SUBPAGECONTAINER_CLASS;H_SUPERPAGECONTAINER_CLASS;H_ROWCONTAINER_CLASS;H_SUPERTABLECONTAINER_CLASS;H_TABLECONTAINER_CLASS;H_PARALLELCOLSCONTAINER_CLASS;H_TABLEONLYCONT_CLASS;H_CLICKHERE_CLASS;H_TEXTMARKER_CLASS;H_TEXT_CLASS',0)} [See list of classes](#)

Determines whether or not the specified coordinate is located within the selection in the object from which this method is called.

Syntax

[objectreference].IsPointWithin(X, Y)

Parameters

X

Indicates the position of the point in units of [Twips](#) on the X axis. There are 1440 Twips per inch. Data type is Long.

Y

Indicates the position of the point in units of [Twips](#) on the Y axis. There are 1440 Twips per inch. Data type is Long.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Word Pro: IsWMCommandValid method

{button ,AL('H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL('H_ISWMCOMMANDVALID_METHOD_EXSCRIPT',1)} [See example](#)

Determines whether the specified Word Pro menu item is available for use at the time this method is called.

Syntax

[objectreference].IsWMCommandValid(CommandID)

Parameters

CommandID

A Numeric expression (or constant) which specifies the ID for the menu item you are checking. Data type is Integer, but you can use the appropriate constant for the menu ID. A complete list of menu ID constants is listed under [Word Pro Menu Command IDs](#)

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the menu item is valid or invalid respectively.

Usage

Word Pro: Italic method

{button ,AL('H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL('H_ITALIC_METHOD_EXSCRIPT',1)} [See example](#)

Sets the italic attribute for selected text, or all following text, if no text is selected. Acts as a toggle, turning the attribute off if it is on and on if it is off. Equivalent to choosing Text - Attributes - Italic.

Syntax

[objectreference].Italic()

Parameters

None.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Word Pro: Item method

```
{button ,AL(^H_BAGCOLLECTION_CLASS;H_BOOKMARKCOLLECTION_CLASS;H_CELLCOLLECTION_CLASS;  
H_CELLLAYOUTCOLLECTION_CLASS;H_CHARACTERSTYLECOLLECTION_CLASS;H_CLICKHERECOLLEC  
TION_CLASS;H_CONNECTEDLAYOUTCOLLECTION_CLASS;H_CONTENTCOLLECTION_CLASS;H_DDELIN  
KCOLLECTION_CLASS;H_DIVISIONCOLLECTION_CLASS;H_DOCUMENTS_CLASS;H_EDITORCOLLECTION  
_CLASS;H_ENDNOTELAYOUTCOLLECTION_CLASS;H_DOCINFOFIELDCOLLECTION_CLASS;H_POWERFIE  
LDCOLLECTION_CLASS;H_FOOTERLAYOUTCOLLECTION_CLASS;H_FOOTNOTECOLLECTION_CLASS;H_  
FOOTNOTELAYOUTCOLLECTION_CLASS;H_FRAMELAYOUTCOLLECTION_CLASS;H_GLOSSARYCOLLECT  
ION_CLASS;H_GRAPHICCOLLECTION_CLASS;H_GRAPHICOLEBJECTCOLLECTION_CLASS;H_GROUPLA  
YOUTCOLLECTION_CLASS;H_HEADERLAYOUTCOLLECTION_CLASS;H_ICONBARCOLLECTION_CLASS;H_  
_LAYOUTCOLLECTION_CLASS;H_MARKERCOLLECTION_CLASS;H_MENUITEMCOLLECTION_CLASS;H_N  
OTELAYOUTCOLLECTION_CLASS;H_OLEOBJECTCOLLECTION_CLASS;H_OUTLINESEQCOLLECTION_CL  
ASS;H_OUTLINESEQITEMCOLLECTION_CLASS;H_PAGELAYOUTCOLLECTION_CLASS;H_PARAGRAPHST  
YLECOLLECTION_CLASS;H_PARALLELCOLSCOLLECTION_CLASS;H_ROWLAYOUTCOLLECTION_CLASS;  
H_RUBYLAYOUTCOLLECTION_CLASS;H_SECTIONCOLLECTION_CLASS;H_SILVERBULLETCOLLECTION  
_CLASS;H_SMARTCORRECTCOLLECTION_CLASS;H_SMARTFILLCOLLECTION_CLASS;H_STATUSBARBU  
TONCOLLECTION_CLASS;H_STRINGCOLLECTION_CLASS;H_SUPERTABLECOLLECTION_CLASS;H_SUPE  
RTABLELAYOUTCOLLECTION_CLASS;H_TABLEHEADINGCOLLECTION_CLASS;H_TABLEHEADINGLAYOUT  
COLLECTION_CLASS;H_TABLELAYOUTCOLLECTION_CLASS;H_TABLEMARKERCOLLECTION_CLASS;H_T  
ABLECOLLECTION_CLASS;H_TEXTCOLLECTION_CLASS;H_TEXTMARKERCOLLECTION_CLASS;H_TEXTS  
TYLECOLLECTION_CLASS;H_VERSIONCOLLECTION_CLASS;H_DOCWINDOWCOLLECTION_CLASS;H_WP  
DATASETCOLLECTION_CLASS;H_DROPCAPLAYOUTCOLLECTION_CLASS;H_TABLEONLYCOLLECTION_C  
LASS;' ,0)} See list of classes
```

```
{button ,AL(^H_ITEM_METHOD_EXSCRIPT',1)} See example
```

Returns an item from a collection class. This method is defined as a part of every collection class. In all but a few collection classes, the return value depends on the type of item stored in a particular collection.

Syntax

[objectreference].Item(idx)

Parameters

idx

Idx represents index. The index specifies which item in a collection you want to return.

A Long data type index specifies the numeric position of an index item in a collection. For example, items in an array are accessed by number (subscript). Therefore, you use a Long data type index to return an index item in an array.

A String data type index item returns the string name of an index item in a collection. For example, items in a list are accessed by string name rather than by number (subscript). Therefore, you use a String data type index to return an index item in a list.

The String idx data type is the parameter used in the Item method for most collection classes, with the exception of five classes. There are four classes that use the Long idx data type parameter. They are: DddLinkCollection, StringCollection, UnitCollection, VersionCollection. The final exception is the SmartFillCollection class, which uses the Integer idx parameter.

Return value

Each collection class returns a different data type.

<u>Class</u>	<u>Returns</u>
BagCollection	Bag object.
BookmarkCollection	Bookmark object.
CellCollection	CellEngine object.
CellLayoutCollection	CellLayout object.
CharacterStyleCollection	CharacterStyle object.
ClickHereCollection	ClickHere object.
ConnectedLayoutCollection	ConnectedLayout object.
ContentCollection	Variant object.
DdeLinkCollection	DdeLink object.

DivisionCollection	Division object.
DocInfoFieldCollection	DocInfoField object.
Documents	TextDocument object.
DropCapLayoutCollection	DropCapLayout object.
EditorCollection	Editor object.
EndnoteLayoutCollection	EndnoteLayout object.
DocInfoFieldCollection	DocInfoField object.
PowerFieldCollection	PowerField object.
FooterLayoutCollection	FooterLayout object.
FootnoteCollection	Footnote object.
FootnoteLayoutCollection	FootnoteLayout object.
FrameLayoutCollection	FrameLayout object.
GlossaryCollection	Glossary object.
GraphicCollection	Graphic object.
GraphicOleObjectCollection	Variant object.
GroupLayoutCollection	GroupLayout object.
HeaderLayoutCollection	HeaderLayout object.
IconBarCollection	IconBar object.
LayoutCollection	Variant object.
MarkerCollection	Variant object.
MenuItemCollection	MenuItem object.
NoteLayoutCollection	NoteLayout object.
OleObjectCollection	OleObject object.
OutlineSeqCollection	OutlineStyleSequence object.
OutlineSeqItemCollection	OutSeqItem object.
PageLayoutCollection	PageLayout object.
ParagraphStyleCollection	ParagraphStyle object.
ParallelColsCollection	ParallelColumns object.
RowLayoutCollection	RowLayout object.
RubyLayoutCollection	RubyLayout object.
SectionCollection	Section object.
SilverBulletCollection	SilverBullet object.
SmartCorrectCollection	SmartCorrect object.
SmartFillCollection	SmartFill object.
StatusBarButtonCollection	StatusBarButton object.
StringCollection	String object.
SuperTableCollection	SuperTable object.
SuperTableLayoutCollection	SuperTableLayout object.
TableCollection	Variant object.
TableHeadingCollection	TableHeading object.
TableHeadingLayoutCollection	TableHeadingLayout object.
TableLayoutCollection	TableLayout object.
TableMarkerCollection	TableMarker object.
TableOnlyCollection	Table object.
TextCollection	Text object.

TextMarkerCollection
TextStyleCollection
DocWindowCollection

TextMarker object.
Variant object.
DocWindow object.

Usage

Word Pro: LabelCreate method

{button ,AL('H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

Syntax

[objectreference].LabelCreate(NumAcross, NumDown, PageWidth, PageHeight, TopMargin, SideMargin, TotalCellWidth, TotalCellHeight, CellWidth, CellHeight)

Parameters

NumAcross

Data type is Integer.

NumDown

Data type is Integer.

PageWidth

Data type is Long.

PageHeight

Data type is Long.

TopMargin

Data type is Long.

SideMargin

Data type is Long.

TotalCellWidth

Data type is Long.

TotalCellHeight

Data type is Long.

CellWidth

Data type is Long.

CellHeight

Data type is Long.

Return value

Integer.

Usage

Word Pro: LabelMerge method

{button ,AL('H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

Syntax

[objectreference].LabelMerge(NumAcross, NumDown, PageWidth, PageHeight, TopMargin, SideMargin, TotalCellWidth, TotalCellHeight, CellWidth, CellHeight)

Parameters

NumAcross

Data type is Integer.

NumDown

Data type is Integer.

PageWidth

Data type is Long.

PageHeight

Data type is Long.

TopMargin

Data type is Long.

SideMargin

Data type is Long.

TotalCellWidth

Data type is Long.

TotalCellHeight

Data type is Long.

CellWidth

Data type is Long.

CellHeight

Data type is Long.

Return value

Integer.

Usage

Word Pro: Link method

{button ,AL('H_DIVISION_CLASS;H_TEXTDOCUMENT_CLASS',0)} [See list of classes](#)

{button ,AL('H_LINK_METHOD_EXSCRIPT',1)} [See example](#)

Syntax

[objectreference].Link(Path,Type)

Parameters

Path

A String expression.

Type

A String expression.

Return value

Usage

Word Pro: Localize method

{button ,AL(^H_DIVISION_CLASS;H_TEXTDOCUMENT_CLASS',0)} [See list of classes](#)

Syntax

[objectreference].Localize(SetToLocal)

Parameters

SetToLocal

A Boolean expression, either True or False.

Return value

Usage

Word Pro: LowerCase method

{button ,AL(^H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL(^H_LOWERCASE_METHOD_EXSCRIPT',1)} [See example](#)

Sets the lowercase attribute for selected text or all following text, if no text is selected. Acts as a toggle, turning the attribute off if it is on and on if it is off. Equivalent to choosing Text - Attributes - Other and then "Lower Case" in the Attributes box.

Syntax

[objectreference].LowerCase()

Parameters

None.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Word Pro: MacroEndRecord method

{button ,AL(^H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

Stops the Script Recorder in Word Pro.

Syntax

[objectreference].MacroEndRecord()

Parameters

None.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Word Pro: MacroPlay method

{button ,AL(^H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

Plays a converted Ami Pro macro.

Syntax

[objectreference].MacroPlay(MacroFileName)

Parameters

MacroFileName

A String expression specifying the path and name of the Ami Pro macro file you want to play.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Word Pro: MacroRecord method

{button ,AL(`H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL(`H_MACRORECORD_METHOD_EXSCRIPT',1)} [See example](#)

Allows you to create a script and save it to a file.

Syntax

WPApplication.MacroRecord(MacroFileName)

Parameters

MacroFileName

Data type is String. The filename must be a valid .LWP or .LSS file.

Return value

Integer. The return values for this method will always be -1 or 0. When testing the return value, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Corresponds to the Record Script dialog box. Equivalent to choosing Edit - Script & Macros - Record Script.

Word Pro: MailDocument method

{button ,AL(^H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

Saves and mails the current document using the MAPI application specified in your WIN.INI file.

Syntax

[objectreference].MailDocument()

Parameters

None.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Word Pro: MakeTableFromText method

{button ,AL(^H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL(^H_MAKETABLEFROMTEXT_METHOD_EXSCRIPT',1)} [See example](#)

Converts the selected text into a table. Equivalent to choosing Create - Table when you have text selected.

Syntax

[objectreference].MakeTableFromText()

Parameters

None.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Word Pro tries to automatically parse the selected text into a table, using tab marks as the cell delimiters and paragraph marks as the row delimiters.

Word Pro: MakeUniqueLinkName method

{button ,AL('H_DDELINKMANAGER_CLASS',0)} [See list of classes](#)

{button ,AL('H_MAKEUNIQUELINKNAME_METHOD_EXSCRIPT',1)} [See example](#)

This method returns the string name. If the name is not unique, a number is added to the link's name.

Syntax

[objectreference].MakeUniqueLinkName(LinkName)

Parameters

LinkName

A String expression representing the name of the link. Required parameter.

Return value

String.

Usage

If the name of the Dde link is not unique, a number is added to the end of the name. This is useful when you are creating many links in a division or section, and want to name them by the division or section name (for example, DdeSection1, DdeSection2, and so on).

Word Pro: ManualFrame method

{button ,AL(`H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL(`H_MANUALFRAME_METHOD_EXSCRIPT',1)} [See example](#)

Turns on the manual frame drawing tool so you can draw a frame manually. Equivalent to Choosing Create - Frame and clicking Size & Place Frame Manually.

Syntax

[objectreference].ManualFrame([FrameStyle])

Parameters

FrameStyle

A String expression which allows you to specify the frame style for a created frame. Optional parameter.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Word Pro: ManualLinkFrames method

{button ,AL(^H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

Turns on the manual frame linking tool. Equivalent to selecting a frame and choosing Frame - Link Frame Contents.

Syntax

[objectreference].ManualLinkFrames()

Parameters

None

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

A non-linked frame must be selected when this method is called. You must click inside another non-selected, non-linked frame to link the contents of the two frames.

Word Pro: ManualTable method

{button ,AL('H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL('H_MANUALTABLE_METHOD_EXSCRIPT',1)} [See example](#)

Turns on the ManualTable drawing tool which allows the user to draw a table manually. Equivalent to choosing Create - Table and clicking Size & Place Table Manually.

Syntax

[objectreference].ManualTable([TableStyle,] [Columns,] [Rows])

Parameters

TableStyle

A String expression which allows you to specify the table style you want to use for the table the user will create. Optional parameter.

Columns

A Numeric expression which allows you to specify the number of columns that will be in the table the user draws. Optional parameter. Data type is Integer.

Rows

A Numeric expression which allows you to specify the number of rows that will be in the table the user draws. Optional parameter. Data type is Integer.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Word Pro: MarkRevisionInsert method

{button ,AL(^H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL(^H_MARKREVISIONINSERT_METHOD_EXSCRIPT',1)} [See example](#)

Marks the selected text as text which was a revision insertion.

Syntax

[objectreference].MarkRevisionInsert()

Parameters

None.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Word Pro: Mark method

```
{button ,AL('H_WPAPPLICATION_CLASS;H_GRAPHICOLEBJECT_CLASS;H_GRAPHIC_CLASS;H_OLEOBJECT_CLASS;H_LAYOUT_CLASS;H_CELLGROUPLAYOUT_CLASS;H_CELLLAYOOUT_CLASS;H_CONNECTEDLAYOOUT_CLASS;H_COLUMNGROUPLAYOUT_CLASS;H_FOOTERLAYOOUT_CLASS;H_FRAMELAYOOUT_CLASS;H_FRAMEGROUPLAYOOUT_CLASS;H_DROPPLAYOOUT_CLASS;H_GROUPLAYOOUT_CLASS;H_HEADERLAYOOUT_CLASS;H_NOTELAYOOUT_CLASS;H_PAGELAYOOUT_CLASS;H_ROWGROUPLAYOOUT_CLASS;H_ROWLAYOOUT_CLASS;H_RUBYLAYOOUT_CLASS;H_SUPERTABLEGROUPLAYOOUT_CLASS;H_SUPERTABLELAYOOUT_CLASS;H_TABLELAYOOUT_CLASS;H_ENDNOTELAYOOUT_CLASS;H_FOOTNOTELAYOOUT_CLASS;H_TABL EHEADINGLAYOOUT_CLASS;H_TOCSUPERTABLELAYOOUT_CLASS;H_CLICKHERE_CLASS;H_TEXTMARKER _CLASS;H_BASSETABLE_CLASS;H_FOOTNOTETABLE_CLASS;H_PARALLELCOLUMNNS_CLASS;H_GLOSSA _RY_CLASS;H_TABLEHEADING_CLASS;H_TABLE_CLASS;H_TEXT_CLASS',0)} See list of classes
```

```
{button ,AL('H_MARK_METHOD_EXSCRIPT',1)} See example
```

Inserts a Marker object of the type you specify, or modifies an existing Marker object. This method is defined by several classes and some classes require additional parameters. For a list of parameters, see Syntax. For details on calling the method from a particular class, see Usage.

Syntax

When calling the method from the *WPApplication* object or a *Graphic*, *OleObject*, *Layout* object:

```
[objectreference].Mark(MarkerType)
```

When calling the method from a *Text*, *TextMarker*, or *ClickHere* object:

```
[objectreference].Mark(MarkerType, [MarkerName,] [RangePart])
```

When calling the method from a *FootnoteTable*, *ParallelColumns*, *TableHeading*, or *Table* object:

```
[objectreference].Mark(MarkerType, [StartRow,] [StartCol,] [NumOfRows,] [NumOfCols])
```

Parameters

MarkerType

Allows you to specify the type of marker you want to insert or modify. Data type is Variant, which allows the value of this parameter to be one of the constants listed below or its numeric equivalent (in parentheses). There is no default value.

<u>Value</u>	<u>Effect</u>
\$LwpMarkerTypeBookmark (590)	Specifies a bookmark marker.
\$LwpMarkerTypeClickhere (593)	Specifies a ClickHere marker.
\$LwpMarkerTypeDde (591)	Specifies a DDE marker.
\$LwpMarkerTypeDefault (589)	Specifies a positional marker not of any other type and is only used to mark a spot or position.
\$LwpMarkerTypeField (592)	Specifies a field marker.
\$LwpMarkerTypeRuby (594)	Specifies a Ruby marker. Used only in the Asian-language versions of Word Pro.

MarkerName

A String expression representing the name of the Marker object you want to modify. Use this parameter to modify an existing Marker object. Do not include a value for this parameter if you are inserting a new Marker object. This parameter is only available when you call this method from a ClickHere, Text, or TextMarker object.

RangePart

Specifies which part of a range marker you want to modify. Do not include a value for this parameter if you are inserting a new Marker object. To use this parameter, you must also provide a value for the MarkerName parameter. Use one of the constants below (or its numeric equivalent) to tell Word Pro to modify the starting point of a marker, the ending point of a marker, or both. Data type is Variant which allows the value of this parameter to be one of the constants listed below or its numeric equivalent (in parentheses). Default is \$LwpRangePartBoth. This parameter is only available when you call this method from a ClickHere, Text, or TextMarker object.

<u>Value</u>	<u>Effect</u>
\$LwpRangePartBoth (1663)	Modifies the starting point and ending point of the named marker to match the position of the insertion point. If something is selected, the

	starting and ending points match the starting and ending points of the selection.
\$LwpRangePartEnd (1665)	Modifies the ending point of the named marker to match the position of the insertion point. If something is selected, the ending point matches the ending point of the selection.
\$LwpRangePartStart (1664)	Modifies the starting point of the marker to match the position of the insertion point. If something is selected, the starting point matches the starting point of the selection.

StartRow

Modifies the starting row for the named marker object within a table object. Used only with FootnoteTable, ParallelColumns, TableHeading, and Table objects. Optional parameter. Data type is Integer.

StartCol

Modifies the starting column for the named marker object within a table object. Used only with FootnoteTable, ParallelColumns, TableHeading, and Table objects. Optional parameter. Data type is Integer.

NumOfRows

Modifies the number of rows for the named marker object within a table object. Used only with FootnoteTable, TableHeading, and Table objects. Optional parameter. Data type is Integer.

NumOfCols

Modifies the number of columns for the named marker object within a table object. Used only with FootnoteTable, ParallelColumns, TableHeading, and Table objects. Optional parameter. Data type is Integer.

Return value

A String value representing the name of the marker object you insert or modify.

Usage

[TextMarker]

When you use this method to create a text marker, Word Pro automatically sets the Collapsible property to False. This means that any subsequent text that is inserted will be included in the marker. You can restrict the scope of the marker to the text selected when it is created by moving the caret outside of the marker range before inserting additional text, or by setting the Collapsible property to True before inserting additional text.

{button ,AL(`H_COLLAPSIBLE_PROPERTY_MEMDEF;H_EXPANDABLE_PROPERTY_MEMDEF',0)} [See related topics](#)

Word Pro: Maximize method

{button ,AL('H_APPLICATIONWINDOW_CLASS;H_DOCWINDOW_CLASS',0)} [See list of classes](#)

{button ,AL('H_MAXIMIZE_METHOD_EXSCRIPT',1)} [See example](#)

Maximizes the Word Pro application or document window.

Syntax

[objectreference].Maximize()

Parameters

None.

Return value

Integer. The return value for this method will always be -1 or 0. When testing the return value, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Word Pro: MergeAddDataRecord method

{button ,AL('H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

Adds a new record to the active Merge data file.

Syntax

[objectreference].MergeAddDataRecord(RecordEntry)

Parameters

RecordEntry

A String expression representing data for the record you are adding. This string must include the data for each field, separated by the appropriate field delimiters, and end with the appropriate record delimiter. For example, here is a String for a data file that uses "~" as the field delimiter and "|" as the record delimiter:

```
Jane~Doe~100 Main St.~Atlanta~GA|
```

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Word Pro: MergeSetDataFile method

{button ,AL('H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL('H_MERGESETDATAFILE_METHOD_EXSCRIPT',1)} [See example](#)

Assigns the specified Merge data file to the currently active document.

Syntax

[objectreference].MergeSetDataFile(DataFilePath, DocWindowHwnd, DataFileType, DataFileTypeID[, DescriptionFile] [, Delimiter])

Parameters

DataFilePath

A String expression representing the path and name of the data file you want to use with the current document. Optional parameter.

DocWindowHwnd

A Numeric expression which specifies the window ID for the Merge document to which you want to attach the data file. Data type is Long. Use a value of 0 (zero) to attach the data file to the currently active window.

DataFileType

A String expression which specifies the file type of the data file. Use the null string ("") to have Word Pro automatically detect the file type. If you provide a value for DataFileType, the DataFileTypeID is not necessary.

DataFileTypeID

A String expression which specifies the ID which Windows registers to represent a specific filter. Use the null string ("") to have Word Pro automatically assign the filter to a Windows registry ID.

DescriptionFile

A String expression that tells Word Pro the name of the description file for your Merge data file. In most data files, each line is a single record and the first record contains the names of the Merge fields. However, some data files are exported from databases and do not have the field names or field delimiters. These data files must be accompanied by a separate description file containing the names of the Merge fields and the field delimiter or field sizes. If you provide no value for this parameter and the data file is not a Word Pro or Ami Pro document, Word Pro prompts you to find out if the first record in the file contains the field names.

Do not use this parameter if you are using an Ami Pro or Word Pro data file. Data files created by Word Pro and Ami Pro do not need a description file. If your data file is in ASCII text or spreadsheet format, and the first line contains the field names, you do not need a description file but you must use a null string ("") for this parameter.

Delimiter

A String expression that identifies the field delimiter for your data file. The value of this parameter should be the single character that acts as the field delimiter. Use a null string ("") to indicate that your data file is tab-delimited. If you provide no value for this field, Word Pro assumes that your data file is formatted in fixed-length ASCII and looks for the description file specified in the DescriptionFile parameter.

Note The record delimiter is presumed to be a line break (paragraph marker).

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Word Pro first checks to be sure the file named in DataFilePath is a valid Merge data file. If the file is not a data file, this method returns a value of 0 (False), indicating failure.

Word Pro: MergeStart method

{button ,AL('H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL('H_MERGESTART_METHOD_EXSCRIPT',1)} [See example](#)

Starts the Merge operation between the Merge document and the Merge data file.

Syntax

[objectreference].MergeStart([MergeAction,] [MergeFilePath])

Parameters

MergeAction

If no Merge document is active, you can specify which Merge document you want to use in starting the Merge operation. Data type is Variant, which allows the value of this parameter to be one of the string values listed below or its numeric equivalent (in parentheses). Default is a null string which causes Word Pro to start the Merge operation, using the currently active Merge document and its assigned data file. If no Merge document is active and no value is included for this parameter, this method returns an error.

<u>Value</u>	<u>Effect</u>
\$LwpMergeActionNewfile (1492)	Opens a new Merge document, based on the SmartMaster specified in the MergeFilePath parameter.
\$LwpMergeActionOpenfile (1493)	Opens the Merge document specified in MergeFilePath and uses that document's data file for the Merge.

MergeFilePath

A String expression representing the path and name of a Merge document or SmartMaster for a new Merge document. Optional parameter.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Use this method with no parameters to start a Merge operation. Then use the Merge method to issue specific instructions about how to carry out the Merge operation, or to switch Merge documents.

Word Pro: Merge method

{button ,AL('H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL('H_MERGE_METHOD_EXSCRIPT',1)} [See example](#)

Allows you to issue instructions to Word Pro during the Merge operation. You can change Merge documents, end the Merge process, send the Merge job to the printer, and so on.

Syntax

[objectreference].Merge(MergeAction,[MergeFilePath])

Parameters

MergeAction

Indicates what action to take with the currently active Merge document. You can use one of the following strings or its numeric equivalent (in parentheses) for the value of this parameter:

<u>Value</u>	<u>Effect</u>
\$LwpMergeActionNewfile (1492)	Opens a new Merge document based on the SmartMaster specified in the MergeFilePath parameter. The currently active Merge data file is assigned to the new merge document.
\$LwpMergeActionOpenfile (1493)	Opens the Merge document specified in MergeFilePath and uses the currently active data file for the Merge.
\$LwpMergeActionNextRecord (1494)	Moves to the next record in the Merge data file. Each time you use this value, you must use this method again with the value, \$LwpMergeActionMergeOne, to get the next record to appear on screen.
\$LwpMergeActionMergeOne (1496)	Updates the screen to display the latest record merged.
\$LwpMergeActionClose (1497)	Ends the Merge process. You must set the MSStep value to 0, 1, or 2.
\$LwpMergeActionContinue (1498)	Sends the entire Merge job to the printer.

MergeFilePath

A String expression representing the path and name of a Merge document or SmartMaster for a new Merge document. Optional parameter.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Use this method to prepare a Merge document for a Merge operation, or to issue instructions during a Merge operation. This method is capable of opening a new Merge document (for example, a letter or envelope) based on a SmartMaster, opening an existing Merge document, switching from one data file to another, moving to the next record in the data file, ending the Merge operation, or sending the Merge job to the printer.

Word Pro: Messages method

{button ,AL('H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL('H_MESSAGES_METHOD_EXSCRIPT',1)} [See example](#)

Handles message boxes while a script is running.

Syntax

[objectreference].Messages(MsgBoxAction)

Parameters

MsgBoxAction

Specifies whether Word Pro should display a message box and wait for the user's response, or take the default response identified in the message box. Data type is Variant, which allows the value of this parameter to be one of the constants listed below or its numeric equivalent (in parentheses). There is no default value.

\$LwpDisplayMsgboxAndWait (2073)

\$LwpTakeDefaultMsgboxAnswer (2074)

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Word Pro: Minimize method

{button ,AL(`H_APPLICATIONWINDOW_CLASS;H_DOCWINDOW_CLASS',0)} [See list of classes](#)

{button ,AL(`H_MINIMIZE_METHOD_EXSCRIPT',1)} [See example](#)

Minimizes the Word Pro application or document window to an icon.

Syntax

[objectreference].Minimize()

Parameters

None.

Return value

Integer. The return value for this method will always be -1 or 0. When testing the return value, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Word Pro: MirrorPage method

```
{button ,AL('H_LAYOUT_CLASS;H_CELLGROUPLAYOUT_CLASS;H_CELLLAYOUT_CLASS;H_CONNECTEDLAYOUT_CLASS;H_COLUMNGROUPLAYOUT_CLASS;H_FOOTERLAYOUT_CLASS;H_FRAMELAYOUT_CLASS;H_FRAMEGROUPLAYOUT_CLASS;H_DROP_CAP_LAYOUT_CLASS;H_GROUPLAYOUT_CLASS;H_HEADERLAYOUT_CLASS;H_NOTELAYOUT_CLASS;H_PAGELAYOUT_CLASS;H_ROWGROUPLAYOUT_CLASS;H_ROWLAYO  
UT_CLASS;H_RUBY_LAYOUT_CLASS;H_SUPERTABLEGROUPLAYOUT_CLASS;H_SUPERTABLELAYOUT  
_CLASS;H_TABLELAYOUT_CLASS;H_ENDNOTE_LAYOUT_CLASS;H_FOOTNOTE_LAYOUT_CLASS;H_TABLE  
HEADING_LAYOUT_CLASS;H_TOCSUPERTABLELAYOUT_CLASS',0)} See list of classes
```

```
{button ,AL('H_MIRRORPAGE_METHOD_EXSCRIPT',1)} See example
```

Makes a mirror image of the referenced layout object and sets up a left/right (complex) page.

Syntax

[objectreference].MirrorPage()

Parameters

Return value

The return value for this method will always be -1 or 0. When testing the return value, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

You must first create the left/right page layout. Then invoke this method to make one page of that left/right layout a mirror image of the other. This method only works on left/right page layouts.

Word Pro: MorphSelectionToTable method

{button ,AL(^H_CLICKHERE_CLASS;H_TEXTMARKER_CLASS;H_TEXT_CLASS',0)} [See list of classes](#)

{button ,AL(^H_MORPHSELECTIONTOTABLE_METHOD_EXSCRIPT',1)} [See example](#)

Converts the current selection of text into a table. Equivalent to choosing Create - Table and clicking Yes at the prompt.

Syntax

[objectreference].MorphSelectionToTable()

Parameters

None.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Word Pro: MostRecentVersion method

{button ,AL(^H_VERSIONMANAGER_CLASS',0)} [See list of classes](#)

Syntax

[objectreference].MostRecentVersion()

Parameters

None.

Return value

Integer.

Usage

```
'Example: IncludeURL property
'This example asks the user if a link should be listed at the bottom of the
'exported HTML document, and then sets the appropriate option.
'The link address and description are set as well.

stat = Messagebox ("Do you want to display a link at the bottom of the Web page?", 36,
"Example Script")
If stat = 6 Then    ' user said yes
    .ApplicationWindow.UserInterfacePrefs.HTMLOptions.IncludeURL = True
    .ApplicationWindow.UserInterfacePrefs.HTMLOptions.URLAddress = "www.lotus.com"
    .ApplicationWindow.UserInterfacePrefs.HTMLOptions.URLDescription = "Lotus Home
Page"
Else
    .ApplicationWindow.UserInterfacePrefs.HTMLOptions.IncludeURL = False
End If
```

```
'Example: InitialParaStyle property
'This example asks the user for the name of a paragraph style
'to use for frame captions.

' First get what we're using now.
CurStyle = .ActiveDocument.FrameCaptionOptions.InitialParaStyle
' Ask user for the new style name. (Defaults to old style name)
NewStyle = Inputbox ("Type the name of the style you want to use for frame captions:",
"Example Script", CurStyle)
If newstyle = "" Then Exit Sub 'user cancelled
'Apply the new style name to the property.
.ActiveDocument.FrameCaptionOptions.InitialParaStyle = newstyle
```

```
'Example: IsNoPrintZone property
'This example asks the user if the No-Print zone should be displayed outside
'the margins of the document, and then sets the appropriate option.
'The link address and description are set as well.

stat = MessageBox ("Do you want to display the no print zone?", 36, "Example Script")
If stat = 6 Then  ' user said yes
    CurrentWindow.WinViewPrefs.IsNoPrintZone = True
' must set margins in color to see no print zone.
    CurrentWindow.WinViewPrefs.IsMarginsInColor = True
Else
    CurrentWindow.WinViewPrefs.IsNoPrintZone = False
End If
```

```
'Example: LastMessageStr property
'This example turns off message box display, then executes the WordCount
'method, which would normally display the word count in a message box.
'The LastMessageStr and LastMessageType methods are used to recreate the
'message box in LotusScript

'First display the normal Word Pro message box
.Wordcount
' now turn off messages, and do it again.
.Messages Off
.Wordcount
'print the message id's and info in the LotusScript output panel
Print "LastMessage ID was " & .LastMessageID
Print "Last Message String was" & .LastMessageStr
Print "LastMessage Type was " & .LastMessageType
' and recreate the message box ourselves.
Msgbox .LastMessageStr, .LastMessageType, "Example Script"
```

```
'Example: LastMessageStr property
'This example turns off message box display, then executes the WordCount
'method, which would normally display the word count in a message box.
'The LastMessageStr and LastMessageType methods are used to recreate the
'message box in LotusScript

'First display the normal Word Pro message box
.Wordcount
' now turn off messages, and do it again.
.Messages Off
.Wordcount
'print the message id's and info in the LotusScript output panel
Print "LastMessage ID was " & .LastMessageID
Print "Last Message String was" & .LastMessageStr
Print "LastMessage Type was " & .LastMessageType
' and recreate the message box ourselves.
Msgbox .LastMessageStr, .LastMessageType, "Example Script"
```

```
'Example: LastMessageType property
'This example turns off message box display, then executes the WordCount
'method, which would normally display the word count in a message box.
'The LastMessageStr and LastMessageType methods are used to recreate the
'message box in LotusScript

'First display the normal Word Pro message box
.Wordcount
' now turn off messages, and do it again.
.Messages Off
.Wordcount
'print the message id's and info in the LotusScript output panel
Print "LastMessage ID was " & .LastMessageID
Print "Last Message String was" & .LastMessageStr
Print "LastMessage Type was " & .LastMessageType
' and recreate the message box ourselves.
Msgbox .LastMessageStr, .LastMessageType, "Example Script"
```


'Example: ReturnExits property
'This example creates a Click Here block, and sets the behavior of the block
'to go to the next block when the user presses the Return key.

```
Dim ClickHereName as String  
ClickHereName = .InsertClickHere()  
.Division.Foundry.ClickHeres(ClickHereName).ReturnExits = True
```

'Example: SplitParagraph method

' This examples inserts two paragraphs of text into the current document.

' RUNTIME DEPENDENCIES: You must have a document open for this script to work.

.Text.InsertText "This is paragraph 1."

.Text.SplitParagraph

.Text.InsertText "This is paragraph 2."

```
'Example: Split method
' This example creates a table with three rows and three columns. The first table
cell
' is then split into three rows and three columns.
' RUNTIME DEPENDENCIES: You must have a document open for this script to work.

.CreateTable False, "Default Table", 3,3
.Table.Split $LwpTableSplitTypeCell, 3,3
```

```
'Example: StartEditMergeData method
' This example adds some records to merge data file for the current document.
' RUNTIME DEPENDENCIES: You must have a document open for this script to work.
' The document must have a merge data file associated with it.

'Get the data file name for the current document
DataFile = .ActiveDocument.MergeOptions.DataFileName

'Opens the current data file invisibly for editing
.StartEditMergeData

'Add some records to the data file
.MergeAddDataRecord "John Doe~100 Main St.~ Atlanta~ GA~30319|"
.MergeAddDataRecord "Jane Doe~100 Main St.~ Atlanta~ GA~30319|"

'Save the data file
.Application.Documents.Item(DataFile).Save

'Close merge
.Merge $LwpMergeActionClose
```

```
'Example: StartEnvelopeDiv method
' This example moves the insertion point to the beginning of the current
' document and inserts sample address information. A new envelope division is
' created and an address prompt is inserted in the return address frame.
' RUNTIME DEPENDENCIES: You must have a document open for this script to work.

.Text.MoveToStart $LwpLocationTypeDocument
.Type "FirstName LastName[Enter]"
.Type "Address[Enter]"
.Type "City, State Zip"

.StartEnvelopeDiv
.GoToLayout "Envelope:Envelope Return"
.Type "Return Address Goes Here[Enter]"
```

```
'Example: StartFieldInsert method
' This example creates a data file for the current document. Two records are
' added and the Merge bar is opened so to insert fields for merging.
' RUNTIME DEPENDENCIES: You must have a document open for this script to work.

.CreateDataFile "~|", "Name~Address~City~State~Zip|", False, "C:\mergedat.lwp"

'Add some records to the virtual datafile
.MergeAddDataRecord "Jane Doe~100 Main St.~ Atlanta~ GA~30319|"
.MergeAddDataRecord "John Doe~100 Main St.~ Atlanta~ GA~30319|"

.StartFieldInsert
```

'Example: StartingNumber property

'This example sets the starting number to 2 for footnotes,

'and to 1 for endnotes at the end of the division, and the end of the document.

.Division.FootnoteOptions.FootnoteNumbering.**StartingNumber** = 2

.Division.FootnoteOptions.EndnoteDivisionNum.**StartingNumber** = 1

.Division.FootnoteOptions.EndnoteDocNum.**StartingNumber** = 1

'Example: Start method
'This example creates a frame and inserts some text.
'The Start method is called to move the insertion point to the beginning
'of the document.

```
.NewFrame 2466, 786, 4257, 2067, "Default Frame"  
.Type "This is a frame."  
.Container.Start $LwpDocumentObjectTypeDocument
```


'Example: StrikeThru method

' This example toggles the strikethru attribute for the selected text.

' RUNTIME DEPENDENCIES: You must have a document open with some text selected
' for this script to work.

.StrikeThru

'Example: SubScript method

' This example toggles the subscript attribute for the selected text.

' RUNTIME DEPENDENCIES: You must have a document open with some text selected

.Subscript

'Example: Subscript property

'This example enables the subscript attribute, inserts some text, then disables the attribute.

.Text.Font.**Subscript** = True

.Text.InsertText "This is Superscript text."

.Text.Font.**Subscript** = False

'Example: Suffix property

'This example creates a table and inserts a number in the first cell

'The number is then formatted in German Mark currency format.

```
.CreateTable False, "Default Table", 4, 4
```

```
.Type "123.45678"
```

```
.Table.CurrentCell.NumericFormat.NumDecimalPlaces = 2
```

```
.Table.CurrentCell.NumericFormat.AnyNumber.Prefix = ""
```

```
.Table.CurrentCell.NumericFormat.AnyNumber.Suffix = " DM"
```

```
.type "[Down]"
```

'Example: Suggestions property
'This example sets up a greeting for the current document.
'The greeting will suggest editing in a new version, and displaying the
'review and comment tools SmartIcons.

```
%INCLUDE "WPBITMSK.LSS" ' This statement lets us use bitmask constants  
.ActiveDocument.EditorManager.CurrentEditor.Suggestions =  
LwpEditSuggEditingInNewVersion + LwpEditSuggRevAndCommentIconbar  
.ActiveDocument.DocControl.UseGreeting = True  
.ActiveDocument.DocControl.Greeting = "Good luck, Mr. Phelps."
```

```
'Example: Sum method
' This example creates a table with 5 rows and 5 columns. Rows 1 through 4
' are consecutively numbered and summed in row 5.
' RUNTIME DEPENDENCIES: You must have a document open for this script to work.

Dim RowCounter As Integer
.CreateTable False, "Default Table", 5,5
For RowCounter = 0 To .Table.NumRows - 2
    .Type Format$(RowCounter)
    .Text.MoveDown 1
Next

.Table.Sum $LwpTableSumScopeColumn
```

'Example: SuperScript method

' This example toggles the superscript attribute for the selected text.

' RUNTIME DEPENDENCIES: You must have a document open with some text selected

.Superscript

'Example: Superscript property

'This example enables the superscript attribute, inserts some text, then disables the attribute.

```
.Text.Font.Superscript = True
```

```
.Text.InsertText "This is Superscript text."
```

```
.Text.Font.Superscript = False
```



```
'Example: SuppressHeaders property
'This example asks the user whether to print headers on filler pages in the current
division,
'and then sets the appropriate option.

stat = MessageBox ("Do you want to print headers on filler pages?", 36, "Example
Script")
If stat = 6 Then ' user said yes
    .Division.DivisionInfo.SuppressHeaders = 0
Else
    .Division.DivisionInfo.SuppressHeaders = 1
End If
```

'Example: TabExits property
'This example creates a Click Here block, and sets the behavior of the block
'to go to the next block when the user types a Tab.

```
Dim ClickHereName as String  
ClickHereName = .InsertClickHere()  
.Division.Foundry.ClickHeres(ClickHereName).TabExits = True
```

```
'Example: TableOfContents property
'This example asks the user if a table of contents should be created for
'HTML documents which are exported, and then sets the appropriate option.

stat = MessageBox ("Do you want to create a table of contents frame?", 36, "Example
Script")
If stat = 6 Then ' user said yes
    .ApplicationWindow.UserInterfacePrefs.HTMLOptions.TableOfContents = True
Else
    .ApplicationWindow.UserInterfacePrefs.HTMLOptions.TableOfContents = False
End If
```

'Example: TabOrder property

'This example creates a new Click Here block, and sets the tab order to the number of Click Here blocks.

Dim TabIndex as Integer

Dim ClickHereName as String

ClickHereName = .InsertClickHere()

TabIndex = .Division.Foundry.ClickHeres.Count + 1 'since array is 0 based.

.Division.Foundry.ClickHeres(ClickHereName).**TabOrder**= TabIndex

'Example: TabRack property

'This example clears all local tabs, then sets left aligned tabs

'at one inch and three inches from the left margin.

```
.Text.TabRack.ClearAll
```

```
.Text.TabRack.InsertOne 1440, $LwpTabTypeLeft, $LwpTabLeaderNone, $LwpTabRelativeLeft,  
0
```

```
.Text.TabRack.InsertOne 4320, $LwpTabTypeLeft, $LwpTabLeaderNone, $LwpTabRelativeLeft,  
0
```

'Example: TabSpacing property

'This example sets the spacing of default tabs to 1 inch (1440 twips).

.Division.**TabSpacing** = 1440

'Example: TeamMail method

' This example displays the Team Mail dialog box.

' RUNTIME DEPENDENCIES: You must have a document open for this script to work.

.TeamMail

```
'Example: TextAfter property
'This example asks the user for the text to place after the caption
'number in frame captions.

' First get what we're using now.
CurText = .ActiveDocument.FrameCaptionOptions.TextAfter
' Ask user for the new text. (Defaults to old value)
NewText = Inputbox ("Type the text you want to use after the caption number:",
"Example Script", CurText)
If NewText = "" Then Exit Sub 'user cancelled
'Apply the new text to the property.
.ActiveDocument.FrameCaptionOptions.TextAfter = NewText
```



```
'Example: TextBefore property
'This example asks the user for the text to place before the caption
'number in frame captions.

' First get what we're using now.
CurText = .ActiveDocument.FrameCaptionOptions.TextBefore
' Ask user for the new text. (Defaults to old value)
NewText = Inputbox ("Type the text you want to use before the caption number:",
"Example Script", CurText)
If NewText = "" Then Exit Sub 'user cancelled
'Apply the new text to the property.
.ActiveDocument.FrameCaptionOptions.TextBefore = NewText
```

```

'Example: TextMarkersInSelection method
' This example inserts some text and text markers in the current document.
' The text is then selected, and the names of the markers in the selection
' is printed to the LotusScript output panel.

'Insert the first sentence and a text marker
.Text.InsertText "This is a test sentence. ", False
MyMark = .Text.Mark ($LwpMarkerTypeDefault)
.Text.InsertText "A second phrase. ", False
.Division.Foundry.Markers(MyMark).Collapsible = True ' this ends the marked area

'Now insert second sentence and mark.
.Text.InsertText "This is another test sentence. ", False
MyMark = .Text.Mark ($LwpMarkerTypeDefault)
.Text.InsertText "A third phrase. ", False
.Division.Foundry.Markers(MyMark).Collapsible = True
.Text.InsertText "And a final sentence.", False
.Text.Select $LwpSelectObjectTypeSentence
'Now let's get the markers in the selection
Print "First mark test"
Forall newmarks In .Text.TextMarkersInSelection
    Print newmarks
End Forall

.Text.Select $LwpSelectObjectTypeStream
Print "Now the entire stream."
Forall newmarks In .Text.TextMarkersInSelection
    Print newmarks
End Forall

```

'Example: TextTightness property

'This example sets the kerning of the current character to 4%.

.Text.Font.**TextTightness** = 4

'Example: Text property

'This example inserts some text into the current document, then bolds the text.

```
.Text.InsertText "Now is the time for all good men to come to the aid of their party."  
"
```

```
.Text.Font.Bold = True
```

```
.Text.InsertText "This is bolded text."
```

```
.Text.Font.Bold = False
```

```
'Example: TileWindowHorz method
' This example creates two new documents based on the 'DEFAULT.MWP'
' SmartMaster.
' The script then prompts you to tile the new windows horizontally.

.NewDocument , , "DEFAULT.MWP", ,
.NewDocument , , "DEFAULT.MWP", ,
MessageBox "Click OK to tile the new windows horizontally.", MB_OK, "Example Script"

.TileWindowHorz
```

```
'Example: TileWindowVert method
' This example creates two new documents based on the 'DEFAULT.MWP'
' SmartMaster.
' The script then prompts you to tile the new windows vertically.

.NewDocument , , "DEFAULT.MWP", ,
.NewDocument , , "DEFAULT.MWP", ,
MessageBox "Click OK to tile the new windows vertically.", MB_OK, "Example Script"

.TileWindowVert
```

```
'Example: TileWindow property
' This example creates two new documents based on the 'DEFAULT.MWP'
' SmartMaster.
' The script then prompts you to tile the new windows horizontally.

.NewDocument , , "DEFAULT.MWP", ,
.NewDocument , , "DEFAULT.MWP", ,
MessageBox "Click OK to tile the new windows.", MB_OK, "Example Script"

CurrentWindow.TileWindow = $lwptiletypehorz
```

'Example: Tile method

' This example tiles any open document windows.

' RUNTIME DEPENDENCIES: You must have a document open for this script to work.

.ApplicationWindow.Tile

'Example: TimedSave method

' This example performs a Timed Save of all the open documents.

' RUNTIME DEPENDENCIES: You must have a document open for this script to work.

.TimedSave

Word Pro: IndexPrimaryStyleName property

{button ,AL('H_PREFERENCES_CLASS',0)} [See list of classes](#)

(Read-write)

Data Type

String

Syntax

indexprimarystylevalue = [objectreference].IndexPrimaryStyleName

[objectreference].IndexPrimaryStyleName = indexprimarystylevalue

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Word Pro: IndexRange property

{button ,AL('H_INDEXSECTION_CLASS',0)} [See list of classes](#)

(Read-write)

Data Type

String

Syntax

indexrangevalue = [objectreference].IndexRange

[objectreference].IndexRange = indexrangevalue

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Word Pro: IndexSecondaryStyleName property

{button ,AL('H_PREFERENCES_CLASS',0)} [See list of classes](#)

(Read-write)

Data Type

String

Syntax

indexsecondarystylevalue = [objectreference].IndexSecondaryStyleName

[objectreference].IndexSecondaryStyleName = indexsecondarystylevalue

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Word Pro: IndexSection property

{button ,AL('H_INDEXSECTION_CLASS',0)} [See list of classes](#)

(Read-write)

Data Type

String

Syntax

indexsectionvalue = [objectreference].IndexSection

[objectreference].IndexSection = indexsectionvalue

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Word Pro: IndexSeparatorStyleName property

{button ,AL('H_PREFERENCES_CLASS',0)} [See list of classes](#)

(Read-write)

Data Type

String

Syntax

indexseparatorstylevalue = [objectreference].IndexSeparatorStyleName

[objectreference].IndexSeparatorStyleName = indexseparatorstylevalue

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Word Pro: IndexSource property

{button ,AL('H_INDEXSECTION_CLASS',0)} [See list of classes](#)

(Read-write)

Data Type

Variant (Enumerated)

GenerateFrom

Syntax

indexsourcevalue = [objectreference].IndexSource

[objectreference].IndexSource = indexsourcevalue

Legal values

\$LwpGenerateFromCurrentdivision (362)

\$LwpGenerateFromCurrentleveldivision (361)

\$LwpGenerateFromCurrentsection (363)

\$LwpGenerateFromEntiredocument (360)

\$LwpGenerateFromMarker (364)

Usage

Word Pro: InDocument property

{button ,AL(^H_OLEOBJECT_CLASS',0)} [See list of classes](#)

(Read-only)

Data Type

[Integer](#)

Syntax

indocumentvalue = [objectreference].InDocument

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Word Pro: InfoBoxSelectionText property

{button ,AL(^H_GRAPHIC_CLASS',0)} [See list of classes](#)

(Read-only)

Data Type

String

Syntax

infoboxselectiontextvalue = [objectreference].InfoBoxSelectionText

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Word Pro: InitializeRoute property

{button ,AL('H_MAILROUTING_CLASS',0)} [See list of classes](#)

(Read-write) Indicates the default state (broadcast vs. route) when you use the Initialize property to set defaults.

Data Type

Integer

Syntax

initializeroutevalue = [objectreference].InitializeRoute

[objectreference].InitializeRoute = initializeroutevalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Used in conjunction with the Initialize property. First, set Initialize to True. Then set the desired value for this property (broadcast or route). If the Initialize property is 0, this property is ignored.

Word Pro: Initialize property

{button ,AL('H_MAILROUTING_CLASS',0)} [See list of classes](#)

(Read-write) Indicates if you can set up defaults in the Mail Routing dialog box.

Data Type

Integer

Syntax

initializevalue = [objectreference].Initialize

[objectreference].Initialize = initializevalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

A value of 1 indicates user-defined settings.

Word Pro: InitialParaStyle property

{button ,AL('H_FRAMECAPTIONOPTIONS_CLASS',0)} [See list of classes](#)

{button ,AL('H_INITIALPARASTYLE_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) The name of the paragraph style to be used as a default for frame captions.

Data Type

String.

Syntax

initialparastylevalue = [objectreference].InitialParaStyle

[objectreference].InitialParaStyle = initialparastylevalue

Legal values

String.

Usage

This property is kept in the document's FrameCaptionOptions object so there can be consistency between frame captions.

Word Pro: InitialsForFilters property

{button ,AL('H_NOTELAYOUT_CLASS',0)} [See list of classes](#)

(Read-write)The initials of the user that created a specific comment note.

Data Type

String

Syntax

initialsforfiltersvalue = [objectreference].InitialsForFilters

[objectreference].InitialsForFilters = initialsforfiltersvalue

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

The default value for this property is contained in the "Initials" setting on the Personal panel of the Word Pro Preferences dialog box. Modifying the NameForFilters property of a specific comment note does not change the name setting under Word Pro Preferences.

Word Pro: InsertFont property

{button ,AL(^H_EDITOR_CLASS;H_REVISIONDISPLAY_CLASS',0)} [See list of classes](#)

(Read-only) The collection of attributes associated with a font object that marks inserted text in a document.

Data Type

[Font](#)

Syntax

insertfontvalue = [objectreference].InsertFont

Legal values

Always contains an instance of the Font class.

Usage

[Editor]

Equivalent to the "Markup for insertions" options in the Markup Options dialog box, located by clicking Markup Options on the General panel of the Word Pro Preferences dialog box.

Word Pro: InsertionMode property

{button ,AL('H_PREFERENCES_CLASS',0)} [See list of classes](#)

{button ,AL('H_INSERTIONMODE_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Sets or returns the current text insertion mode.

Data Type

Integer

Syntax

insertionmodevalue = [objectreference].InsertionMode

[objectreference].InsertionMode = insertionmodevalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

There are two modes of typing in Word Pro: Insert and Typeover.

To use Insert mode, assign this property a value of True. In Insert mode, you can insert text at the location of the insertion point without overwriting existing text.

To use Typeover mode, assign this property a value of False. In Typeover mode, typing replaces existing text at the location of the insertion point.

To toggle typing modes in Word Pro without using LotusScript, press INSERT.

Word Pro: Interactive property

{button ,AL(^H_APPLICATION_CLASS;H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

(Read-only) Indicates whether or not Word Pro is in interactive mode. If it is in interactive mode (True), the user can interact with the application using the keyboard, mouse, or other input device.

Data Type

Integer

Syntax

interactivevalue = [objectreference].Interactive

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

If you suspect that Word Pro may be engaged in an OLE automation session when your script runs, you can use this property as part of an IF...THEN statement to make sure that Word Pro is available before executing your script.

Note OLE is not supported under OS/2.

Word Pro: Interval property

{button ,AL('H_LWPTIMER_CLASS',0)} [See list of classes](#)

(Read-write)

Data Type

Integer

Syntax

intervalvalue = [objectreference].Interval

[objectreference].Interval = intervalvalue

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Word Pro: IsActionOnButtonDown property

{button ,AL('H_ICONBARMANAGER_CLASS',0)} [See list of classes](#)

(Write-only) Indicates if the action represented by an icon occurs when the icon is pressed down. Typically, the icon action occurs on the WM_LBUTTONDOWN. Setting this flag causes the action to occur on the WM_LBUTTONDOWNDOWN message.

Data Type

Integer

Syntax

[objectreference].IsActionOnButtonDown = isactiononbuttondownvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

For this property to function, you must first select an icon using either the SelectStandardIcon or SelectCustomIcon method. Default is False. If the flag is set to True, the action occurs when the icon is depressed. If set to False, the action occurs when the icon is let up.

Word Pro: IsActive property

{button ,AL(^H_GRAPHICOLEBJECT_CLASS;H_GRAPHIC_CLASS;H_OLEOBJECT_CLASS',0)} [See list of classes](#)

(Read-only)

Data Type

[Integer](#)

Syntax

isactivevalue = [objectreference].IsActive

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

UsageUsage

Word Pro: IsAmikake property

{button ,AL(^H_AMIKAKE_CLASS',0)} [See list of classes](#)

(Read-only)

Data Type

Integer.

Syntax

isamikakevalue = [objectreference].IsAmikake

Legal values**Usage**

Word Pro: IsAmiProTableImport property

{button ,AL(^H_MERGEOPTIONS_CLASS',0)} [See list of classes](#)

(Read-write) Indicates whether the Merge data file is derived from an Ami Pro table.

Data Type

Integer

Syntax

isamiprotatableimportvalue = [objectreference].IsAmiProTableImport

[objectreference].IsAmiProTableImport = isamiprotatableimportvalue

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Word Pro: IsAnnotateOnly property

{button ,AL(^H_DIVISION_CLASS;H_TEXTDOCUMENT_CLASS',0)} [See list of classes](#)

(Read-write)

Allows an editor to edit a document only by creating and adding to Comment Notes.

Data Type

Integer

Syntax

isannotateonlyvalue = [objectreference].IsAnnotateOnly

[objectreference].IsAnnotateOnly = isannotateonlyvalue

Legal values

The legal values for this property are -1 and 0 but you may use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Equivalent to choosing File - TeamSecurity, clicking the Editing Rights panel, and selecting "Allowed as Comments Notes only" in the "Edits are" box.

Word Pro: IsAsciiCRLF property

{button ,AL('H_FILTER_CLASS',0)} [See list of classes](#)

{button ,AL('H_ISASCIICRLF_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write)

Data Type

[Integer](#)

Syntax

isasciicrlfvalue = [objectreference].IsAsciiCRLF

[objectreference].IsAsciiCRLF = isasciicrlfvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Word Pro: IsAsciiKeepStyle property

{button ,AL('H_FILTER_CLASS',0)} [See list of classes](#)

{button ,AL('H_ISASCIKEEPSTYLE_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write)

Data Type

[Integer](#)

Syntax

isasciikeepstylevalue = [objectreference].IsAsciiKeepStyle

[objectreference].IsAsciiKeepStyle = isasciikeepstylevalue

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Word Pro: IsAutoGrow property

{button ,AL(^H_BASETABLE_CLASS;H_FOOTNOTETABLE_CLASS;H_PARALLELCOLUMNS_CLASS;H_GLOSSARY_CLASS;H_TABLEHEADING_CLASS;H_TABLE_CLASS',0)} [See list of classes](#)

{button ,AL(^H_ISAUTOGROW_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Indicates whether or not the height of a row in a table object increases when text wraps to another line, or when the point or font size of the text changes.

Data Type

Integer

Syntax

isautogrowvalue = [objectreference].IsAutoGrow

[objectreference].IsAutoGrow = isautogrowvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Equivalent to simultaneously modifying the "Automatic row height" setting for all rows within a table.

Word Pro: IsBadReference property

{button ,AL('H_CELLLAYOUT_CLASS;H_CONNECTEDLAYOUT_CLASS',0)} [See list of classes](#)

(Read-write) Indicates whether or not a formula in a table cell refers to a cell that does not exist in the table.

Data Type

Integer

Syntax

isbadreferencevalue = [objectreference].IsBadReference

[objectreference].IsBadReference = isbadreferencevalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

If this property returns True, the formula in the table cell refers to a cell that does not exist in the table. If this property returns False, the formula in the table cell does not refer to a non-existent cell.

Word Pro: IsBorder property

{button ,AL(^H_CHARACTERBORDER_CLASS;H_PARAGRAPHBORDER_CLASS,0)} [See list of classes](#)

(Read-only)

Data Type

Integer

Syntax

isbordervalue = [objectreference].IsBorder

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Word Pro: IsBothSidesEqual property

{button ,AL('H_INDENT_CLASS;H_RELATIVEINDENT_CLASS',0)} [See list of classes](#)

(Read-write)

Data Type

Integer

Syntax

isbothsidesequalvalue = [objectreference].IsBothSidesEqual

[objectreference].IsBothSidesEqual = isbothsidesequalvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Word Pro: IsBreakable property

{button ,AL(^H_LAYOUT_CLASS;H_CELLGROUPLAYOUT_CLASS;H_CELLLAYOUT_CLASS;H_CONNECTEDLAYOUT_CLASS;H_COLUMNGROUPLAYOUT_CLASS;H_FOOTERLAYOUT_CLASS;H_FRAMELAYOUT_CLASS;H_FRAMEGROUPLAYOUT_CLASS;H_DROP_CAPLAYOUT_CLASS;H_GROUPLAYOUT_CLASS;H_HEADERLAYOUT_CLASS;H_NOTELAYOUT_CLASS;H_PAGELAYOUT_CLASS;H_ROWGROUPLAYOUT_CLASS;H_ROWLAYOUT_CLASS;H_RUBY_LAYOUT_CLASS;H_SUPERTABLEGROUPLAYOUT_CLASS;H_SUPERTABLELAYOUT_CLASS;H_TABLELAYOUT_CLASS;H_ENDNOTELAYOUT_CLASS;H_FOOTNOTELAYOUT_CLASS;H_TABLEHEADINGLAYOUT_CLASS;H_TOCSUPERTABLELAYOUT_CLASS',0)} [See list of classes](#)

(Read-only) Indicates whether or not a break can be inserted into the content of a layout object.

Data Type

Integer

Syntax

isbreakablevalue = [objectreference].IsBreakable

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Word Pro: IsBubbleHelp property

{button ,AL('H_ICONBARMANAGER_CLASS',0)} [See list of classes](#)

(Write-only) Indicates whether bubble help for icons is shown when the cursor hovers over an icon. If set to True, bubble help displays. This flag can be changed through the user interface in the "Show icon descriptions (bubble help)" field in the SmartIcons Setup dialog box.

Data Type

Integer

Syntax

[objectreference].IsBubbleHelp= isbubblehelpvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

You can write a script that lets bubble help display or not display over all icons. The user interface is found in the "Show icon descriptions (bubble help)" field in the SmartIcons Setup dialog box.

Word Pro: IsCellMenuEnabled property

{button ,AL('H_CONTEXTMENUOPTIONS_CLASS',0)} [See list of classes](#)

(Read-write) Controls whether or not the cell menu displays when the cursor is within a cell object.

Data Type

Integer

Syntax

iscellmenuenabledvalue = [objectreference].IsCellMenuEnabled

[objectreference].IsCellMenuEnabled = iscellmenuenabledvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

A True setting allows the cell menu to display when the focus is in a cell object. A False setting prevents the display of the cell menu when the focus is in a cell object.

Word Pro: IsChangedOtherThanLinkTo property

{button ,AL(^H_DIVISION_CLASS;H_TEXTDOCUMENT_CLASS',0)} [See list of classes](#)

(Read-write)

Data Type

Integer.

Syntax

[objectreference].IsChangedOtherThanLinkTo = ischangedotherthanlinktovalue

ischangedotherthanlinktovalue = [objectreference].IsChangedOtherThanLinkTo

Legal values**Usage**

Word Pro: IsChangedSinceTimeSave property

{button ,AL(^H_DIVISION_CLASS;H_TEXTDOCUMENT_CLASS',0)} [See list of classes](#)

(Read-write)

Data Type

Integer

Syntax

ischangedsincetimesavevalue = [objectreference].IsChangedSinceTimeSave

[objectreference].IsChangedSinceTimeSave = ischangedsincetimesavevalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Word Pro: IsChangedToLinkTo property

{button ,AL(^H_DIVISION_CLASS;H_TEXTDOCUMENT_CLASS',0)} [See list of classes](#)

(Read-write)

Data Type

Integer.

Syntax

[objectreference].IsChangedToLinkTo = ischangedtolinktovalue

ischangedtolinktovalue = [objectreference].IsChangedToLinkTo

Legal values

Usage

Word Pro: IsChanged property

```
{button ,AL(^H_DIVISION_CLASS;H_MARKER_CLASS;H_POWERFIELD_CLASS;H_CLICKHERE_CLASS;H_TEXT  
MARKER_CLASS;H_RUBYMARKER_CLASS;H_TABLEMARKER_CLASS;H_TEXTDOCUMENT_CLASS';0))  
See list of classes
```

(Read-write) Indicates whether the specified object has changed since the most recent save.

Data Type

Integer

Syntax

ischangedvalue = [objectreference].IsChanged

[objectreference].IsChanged = ischangedvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Also refer to the Changed property of the Document and DivisionInfo classes.

Word Pro: IsChartLink property

{button ,AL(^H_GRAPHICOLEBJECT_CLASS;H_GRAPHIC_CLASS;H_OLEOBJECT_CLASS',0)} [See list of classes](#)

(Read-only)

Data Type

Integer.

Syntax

ischartlinkvalue = [objectreference].IsChartLink

Legal values

Usage

Word Pro: IsChildSpannable property

```
{button ,AL(^H_LAYOUT_CLASS;H_CELLGROUPLAYOUT_CLASS;H_CELLLAYOUT_CLASS;H_CONNECTEDLAY  
OUT_CLASS;H_COLUMNGROUPLAYOUT_CLASS;H_FOOTERLAYOUT_CLASS;H_FRAMELAYOUT_CLASS;H  
_FRAMEGROUPLAYOUT_CLASS;H_DROP_CAPLAYOUT_CLASS;H_GROUPLAYOUT_CLASS;H_HEADERLAY  
OUT_CLASS;H_NOTELAYOUT_CLASS;H_PAGELAYOUT_CLASS;H_ROWGROUPLAYOUT_CLASS;H_ROWLA  
YOUT_CLASS;H_RUBYLAYOUT_CLASS;H_SUPERTABLEGROUPLAYOUT_CLASS;H_SUPERTABLELAYOUT  
_CLASS;H_TABLELAYOUT_CLASS;H_ENDNOTELAYOUT_CLASS;H_FOOTNOTELAYOUT_CLASS;H_TABLE  
HEADINGLAYOUT_CLASS;H_TOCSUPERTABLELAYOUT_CLASS',0)} See list of classes
```

(Read-write) Allows a layout object's children to span across pages.

Data Type

Integer

Syntax

ischildspannablevalue = [objectreference].IsChildSpannable

[objectreference].IsChildSpannable = ischildspannablevalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Word Pro: IsCollapsed property

```
{button ,AL(^H_LAYOUT_CLASS;H_CELLGROUPLAYOUT_CLASS;H_CELLLAYOUT_CLASS;H_CONNECTEDLAYOUT_CLASS;H_COLUMNGROUPLAYOUT_CLASS;H_FOOTERLAYOUT_CLASS;H_FRAMELAYOUT_CLASS;H_FRAMEGROUPLAYOUT_CLASS;H_DROPDOWNLAYOUT_CLASS;H_GROUPLAYOUT_CLASS;H_HEADERLAYOUT_CLASS;H_NOTELAYOUT_CLASS;H_PAGELAYOUT_CLASS;H_ROWGROUPLAYOUT_CLASS;H_ROWLAYO  
UT_CLASS;H_RUBYLAYOUT_CLASS;H_SUPERTABLEGROUPLAYOUT_CLASS;H_SUPERTABLELAYOUT_CLASS;H_TABLELAYOUT_CLASS;H_ENDNOTELAYOUT_CLASS;H_FOOTNOTELAYOUT_CLASS;H_TABLE  
HEADINGLAYOUT_CLASS;H_TOCSUPERTABLELAYOUT_CLASS',0)} See list of classes
```

(Read-write) Used with the NoteLayout property to determine if a note is collapsed.

Data Type

Integer

Syntax

iscollapsedvalue = [objectreference].IsCollapsed

[objectreference].IsCollapsed = iscollapsedvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

When a NoteLayout object is collapsed, it appears in Word Pro as a small colored box.

Word Pro: IsCollapsible property

```
{button ,AL(^H_LAYOUT_CLASS;H_CELLGROUPLAYOUT_CLASS;H_CELLLAYOUT_CLASS;H_CONNECTEDLAY  
OUT_CLASS;H_COLUMNGROUPLAYOUT_CLASS;H_FOOTERLAYOUT_CLASS;H_FRAMELAYOUT_CLASS;H  
_FRAMEGROUPLAYOUT_CLASS;H_DROP_CAPLAYOUT_CLASS;H_GROUPLAYOUT_CLASS;H_HEADERLAY  
OUT_CLASS;H_NOTELAYOUT_CLASS;H_PAGELAYOUT_CLASS;H_ROWGROUPLAYOUT_CLASS;H_ROWLA  
YOUT_CLASS;H_RUBYLAYOUT_CLASS;H_SUPERTABLEGROUPLAYOUT_CLASS;H_SUPERTABLELAYOUT  
_CLASS;H_TABLELAYOUT_CLASS;H_ENDNOTELAYOUT_CLASS;H_FOOTNOTELAYOUT_CLASS;H_TABLE  
HEADINGLAYOUT_CLASS;H_TOCSUPERTABLELAYOUT_CLASS',0)} See list of classes
```

(Read-write) Determines whether or not a layout object can shrink to an icon.

Data Type

Integer

Syntax

iscollapsiblevalue = [objectreference].IsCollapsible

[objectreference].IsCollapsible = iscollapsiblevalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

May cause unpredictable results if used with layout objects which are not comment note objects.

Word Pro: IsColumnBreakable property

{button ,AL('H_LAYOUT_CLASS;H_CELLGROUPLAYOUT_CLASS;H_CELLLAYOUT_CLASS;H_CONNECTEDLAY
OUT_CLASS;H_COLUMNGROUPLAYOUT_CLASS;H_FOOTERLAYOUT_CLASS;H_FRAMELAYOUT_CLASS;H
_FRAMEGROUPLAYOUT_CLASS;H_DROP_CAPLAYOUT_CLASS;H_GROUPLAYOUT_CLASS;H_HEADERLAY
OUT_CLASS;H_NOTELAYOUT_CLASS;H_PAGELAYOUT_CLASS;H_ROWGROUPLAYOUT_CLASS;H_ROWLA
YOUT_CLASS;H_RUBYLAYOUT_CLASS;H_SUPERTABLEGROUPLAYOUT_CLASS;H_SUPERTABLELAYOUT
_CLASS;H_TABLELAYOUT_CLASS;H_ENDNOTELAYOUT_CLASS;H_FOOTNOTELAYOUT_CLASS;H_TABLE
HEADINGLAYOUT_CLASS;H_TOCSUPERTABLELAYOUT_CLASS',0)} [See list of classes](#)

(Read-only) Indicates whether or not a column break can be inserted into the content of a layout object.

Data Type

Integer

Syntax

iscolumnbreakablevalue = [objectreference].IsColumnBreakable

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Word Pro: IsColumnBreakAfter property

{button ,AL('H_BREAKS_CLASS',0)} [See list of classes](#)

{button ,AL('H_ISCOLUMNBREAKAFTER_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Determines whether or not there is a column break after the current paragraph.

Data Type

[Integer](#)

Syntax

iscolumnbreakaftervalue = [objectreference].IsColumnBreakAfter

[objectreference].IsColumnBreakAfter = iscolumnbreakaftervalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

This property is equivalent to the "Break column" setting of the "After paragraph" option, located in the Text InfoBox.

Word Pro: IsColumnBreakBefore property

{button ,AL('H_BREAKS_CLASS',0)} [See list of classes](#)

{button ,AL('H_ISCOLUMNBREAKBEFORE_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Determines whether or not Word Pro will add a column break before the specified object.

Data Type

[Integer](#)

Syntax

iscolumnbreakbeforevalue = [objectreference].IsColumnBreakBefore

[objectreference].IsColumnBreakBefore = iscolumnbreakbeforevalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Setting this property to True is equivalent to selecting "Break column" in the "Before paragraph" box on the Breaks panel of the Text InfoBox.

Setting this property to False is equivalent to deselecting that option.

Word Pro: IsComplex property

```
{button ,AL('H_LAYOUT_CLASS;H_CELLGROUPLAYOUT_CLASS;H_CELLLAYOOUT_CLASS;H_CONNECTEDLAYOOUT_CLASS;H_COLUMNGROUPLAYOUT_CLASS;H_FOOTERLAYOOUT_CLASS;H_FRAMELAYOOUT_CLASS;H_FRAMEGROUPLAYOUT_CLASS;H_DROP_CAPLAYOOUT_CLASS;H_GROUPLAYOOUT_CLASS;H_HEADERLAYOOUT_CLASS;H_NOTELAYOOUT_CLASS;H_PAGELAYOOUT_CLASS;H_ROWGROUPLAYOOUT_CLASS;H_ROWLAYOOUT_CLASS;H_RUBYLAYOOUT_CLASS;H_SUPERTABLEGROUPLAYOOUT_CLASS;H_SUPERTABLELAYOOUT_CLASS;H_TABLELAYOOUT_CLASS;H_ENDNOTELAYOOUT_CLASS;H_FOOTNOTELAYOOUT_CLASS;H_TABLEHEADINGLAYOOUT_CLASS;H_TOCSUPERTABLELAYOOUT_CLASS',0)} See list of classes
```

```
{button ,AL('H_ISCOMPLEX_PROPERTY_EXSCRIPT',1)} See example
```

(Read-write) Indicates whether or not a page layout pertains to all pages or just odd/even pages.

Data Type

[Integer](#)

Syntax

iscomplexvalue = [objectreference].IsComplex

[objectreference].IsComplex = iscomplexvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Equivalent to the "Settings for" option on the Size & Margins panel of the Page layout InfoBox for page layout objects.

```
{button ,AL('H_MIRRORPAGE_METHOD_MEMDEF',0)} See related topics
```

Word Pro: IsConnected property

```
{button ,AL(^H_LAYOUT_CLASS;H_CELLGROUPLAYOUT_CLASS;H_CELLLAYOUT_CLASS;H_CONNECTEDLAY  
OUT_CLASS;H_COLUMNGROUPLAYOUT_CLASS;H_FOOTERLAYOUT_CLASS;H_FRAMELAYOUT_CLASS;H  
_FRAMEGROUPLAYOUT_CLASS;H_DROP_CAPLAYOUT_CLASS;H_GROUPLAYOUT_CLASS;H_HEADERLAY  
OUT_CLASS;H_NOTELAYOUT_CLASS;H_PAGELAYOUT_CLASS;H_ROWGROUPLAYOUT_CLASS;H_ROWLA  
YOUT_CLASS;H_RUBYLAYOUT_CLASS;H_SUPERTABLEGROUPLAYOUT_CLASS;H_SUPERTABLELAYOUT  
_CLASS;H_TABLELAYOUT_CLASS;H_ENDNOTELAYOUT_CLASS;H_FOOTNOTELAYOUT_CLASS;H_TABLE  
HEADINGLAYOUT_CLASS;H_TOCSUPERTABLELAYOUT_CLASS',0)} See list of classes
```

(Read-only) Indicates whether or not a frame is a member of a group.

Data Type

Integer

Syntax

isconnectedvalue = [objectreference].IsConnected

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

This property applies only to frame layout objects. A frame layout object is a member of a group if it is grouped together with sibling frame layout objects.

Word Pro: IsContinuedFrom property

{button ,AL('H_FOOTNOTEOPTIONS_CLASS',0)} [See list of classes](#)

{button ,AL('H_ISCONTINUEDFROM_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Determines whether the "continued from" message is generated.

Data Type

[Integer](#)

Syntax

iscontinuedfromvalue = [objectreference].IsContinuedFrom

[objectreference].IsContinuedFrom = iscontinuedfromvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

This property is equivalent to the "Generate continued from message" box on the Continued Messages panel of the Footnote and Endnote Options dialog box.

Word Pro: IsContinuedOn property

{button ,AL('H_FOOTNOTEOPTIONS_CLASS',0)} [See list of classes](#)

{button ,AL('H_ISCONTINUEDON_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Determines whether the "continued on" message is generated.

Data Type

[Integer](#)

Syntax

iscontinuedonvalue = [objectreference].IsContinuedOn

[objectreference].IsContinuedOn = iscontinuedonvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Equivalent to the "Generate continued on message" option on the Continued Messages panel of the Footnote and Endnote Options dialog box.

Word Pro: IsCumulative property

{button ,AL('H_PARAGRAPHSTYLE_CLASS',0)} [See list of classes](#)

(Read-write)

Data Type

Integer

Syntax

iscumulativevalue = [objectreference].IsCumulative

[objectreference].IsCumulative = iscumulativevalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Word Pro: IsDebug property

{button ,AL(^H_MACRO_CLASS',0)} [See list of classes](#)

(Read-write)

Data Type

Integer

Syntax

isdebugvalue = [objectreference].IsDebug

[objectreference].IsDebug = isdebugvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Word Pro: IsDisableWarningMessages property

{button ,AL('H_USERINTERFACEPREFS_CLASS',0)} [See list of classes](#)

{button ,AL('H_ISDISABLEWARNINGMESSAGES_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Indicates whether Word Pro version warning messages are disabled.

Data Type

[Integer \(bool\)](#)

Syntax

isdisablewarningmessagesvalue = [objectreference].IsDisableWarningMessages

[objectreference].IsDisableWarningMessages = isdisablewarningmessagesvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

A True setting is equivalent to selecting "Version warning messages" in the "Disable" box on the General panel of the Word Pro Preferences dialog box. A False setting is equivalent to deselecting this option.

Word Pro: IsDisplayMisspelled property

{button ,AL('H_WINVIEWPREFS_CLASS',0)} [See list of classes](#)

{button ,AL('H_ISDISPLAYMISPELLED_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Determines whether misspelled words are highlighted.

Data Type

[Integer](#)

Syntax

isdisplaymisspelledvalue = [objectreference].IsDisplayMisspelled

[objectreference].IsDisplayMisspelled = isdisplaymisspelledvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Equivalent to the "Show misspelled words" option in the Show panel of the View Preferences dialog box.

If this property is set to True, Word Pro highlights misspelled words. The color of the highlight is controlled by the [SpellColor](#) property. If set to False, Word Pro does not highlight misspelled words.

Word Pro: IsDivisionExternal property

{button ,AL(`H_DIVISION_CLASS',0)} [See list of classes](#)

{button ,AL(`H_ISDIVISIONEXTERNAL_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write)

Data Type

[Integer](#)

Syntax

isdivisionexternalvalue = [objectreference].IsDivisionExternal

[objectreference].IsDivisionExternal = isdivisionexternalvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

{button ,AL(`H_EXTERNALNAME_PROPERTY_MEMDEF',0)} [See related topics](#)

Word Pro: IsDocLoading property

{button ,AL(^H_DIVISION_CLASS;H_TEXTDOCUMENT_CLASS',0)} [See list of classes](#)

(Read-write)

Data Type

Integer

Syntax

isdocloadingvalue = [objectreference].IsDocLoading

[objectreference].IsDocLoading = isdocloadingvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Word Pro: IsDocObject property

{button ,AL(^H_DIVISIONINFO_CLASS',0)} [See list of classes](#)

(Read-only)

Indicates if the server for an OLE Division is a docobject.

Data Type

[Integer](#)

Syntax

isdocobjectvalue = [objectreference].IsDocObject

Legal values

This property is read-only. The value of this property cannot be set by a script.

Usage

A DocObject is an OLE object you can edit in place within the client, i.e you do not have to launch the client application.

Word Pro: IsDocumentInRoute property

{button ,AL(^H_MAILROUTING_CLASS',0)} [See list of classes](#)

(Read-only) An internal flag that indicates a value about whether a document is in a mail route.

Data Type

Integer

Syntax

isdocumentinroutevalue = [objectreference].IsDocumentInRoute

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Word Pro: IsDoubleWordError property

{button ,AL('H_ATTRIBUTES_CLASS',0)} [See list of classes](#)

(Read-write)

Data Type

Integer

Syntax

isdoubleworderrorvalue = [objectreference].IsDoubleWordError

[objectreference].IsDoubleWordError = isdoubleworderrorvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values. Default is False (0).

Usage

Word Pro: IsDraw property

{button ,AL('H_GRAPHICOLEBJECT_CLASS;H_GRAPHIC_CLASS;H_OLEOBJECT_CLASS',0)} [See list of classes](#)

{button ,AL('H_ISDRAW_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-only) Tests to see if the graphic is a Word Pro drawing.

Data Type

[Integer](#)

Syntax

isdrawvalue = [objectreference].IsDraw

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Word Pro: IsEmptyDoc property

{button ,AL(^H_DIVISION_CLASS;H_TEXTDOCUMENT_CLASS',0)} [See list of classes](#)

(Read-only)

Data Type

Integer

Syntax

isemptydocvalue = [objectreference].IsEmptyDoc

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Word Pro: IsEmpty property

{button ,AL(^H_CONTENT_CLASS;H_GRAPHICOLEBJECT_CLASS;H_GRAPHIC_CLASS;H_OLEOBJECT_CLASSES;H_SUPERTABLE_CLASS;H_BASETABLE_CLASS;H_FOOTNOTETABLE_CLASS;H_PARALLELCOLUMNS_CLASS;H_GLOSSARY_CLASS;H_TABLEHEADING_CLASS;H_TABLE_CLASS;H_TEXT_CLASS;H_CLICKHERE_CLASS',0)} [See list of classes](#)

(Read-only) Indicates whether or not the text stream is empty.

[ClickHere block]

Indicates whether a ClickHere block is in its collapsed state.

[Content][BaseTable]

Indicates whether or not a specific content object is empty.

Data Type

Integer

Syntax

isemptyvalue = [objectreference].IsEmpty

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Returns True only for Text contents that contain no characters. For example, when a default frame is created, it contains an empty Text content. When you type some characters into the frame, the Text content is no longer empty. If you delete all those characters, it is empty again.

Word Pro: IsEnabled property

{button ,AL(^H_ICONBAR_CLASS;H_KINSOKU_CLASS',0)} [See list of classes](#)

(Read-write) Tells you whether or not an icon bar object is enabled and will display, thus allowing you to click on an icon in the set. If an icon bar is disabled, it is also usually hidden.

Data Type

Integer

Syntax

isenabledvalue = [objectreference].IsEnabled

[objectreference].IsEnabled = isenabledvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Allows you to display an icon bar object and click on the icons.

Word Pro: IsEndnoteDivision property

{button ,AL(^H_DIVISION_CLASS;H_TEXTDOCUMENT_CLASS',0)} [See list of classes](#)

(Read-only)

Data Type

Integer.

Syntax

isendnotedivisionvalue = [objectreference].IsEndnoteDivision

Legal values**Usage**

Word Pro: IsEquation property

{button ,AL(`H_GRAPHICOLEBJECT_CLASS;H_GRAPHIC_CLASS;H_OLEOBJECT_CLASS',0)} [See list of classes](#)

{button ,AL(`H_ISEQUATION_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-only) Indicates whether the specified Graphic object is an equation.

Data Type

[Integer](#)

Syntax

isequationvalue = [objectreference].IsEquation

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

A value of True indicates that the specified Graphic object is an equation.

A value of False indicates that the specified Graphic object is not an equation.

Word Pro: IsErrorChecking property

```
{button ,AL(^H_LAYOUT_CLASS;H_CELLGROUPLAYOUT_CLASS;H_CELLLAYOUT_CLASS;H_CONNECTEDLAY  
OUT_CLASS;H_COLUMNGROUPLAYOUT_CLASS;H_FOOTERLAYOUT_CLASS;H_FRAMELAYOUT_CLASS;H  
_FRAMEGROUPLAYOUT_CLASS;H_DROP_CAPLAYOUT_CLASS;H_GROUPLAYOUT_CLASS;H_HEADERLAY  
OUT_CLASS;H_NOTELAYOUT_CLASS;H_PAGELAYOUT_CLASS;H_ROWGROUPLAYOUT_CLASS;H_ROWLA  
YOUT_CLASS;H_RUBYLAYOUT_CLASS;H_SUPERTABLEGROUPLAYOUT_CLASS;H_SUPERTABLELAYOUT  
_CLASS;H_TABLELAYOUT_CLASS;H_ENDNOTELAYOUT_CLASS;H_FOOTNOTELAYOUT_CLASS;H_TABLE  
HEADINGLAYOUT_CLASS;H_TOCSUPERTABLELAYOUT_CLASS',0)} See list of classes
```

(Read-write) Allows you to set error checking to either allow or prevent an invalid operation.

Data Type

Integer

Syntax

iserrorcheckingvalue = [objectreference].IsErrorChecking

[objectreference].IsErrorChecking = iserrorcheckingvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Allows you to determine whether or not the internal error checking of a layout object is enabled. For example, error checking performed on a frame layout object prevents an illegal setting for the size or placement of the frame.

This property can be used by document filters in order to fully control layout objects and their placement within a Word Pro document.

A True setting (-1) prevents an invalid operation. A False setting (0) allows an invalid operation. Default is True.

Word Pro: IsExpandDown property

{button ,AL('H_LAYOUT_CLASS;H_CELLGROUPLAYOUT_CLASS;H_CELLLAYOUT_CLASS;H_CONNECTEDLAY
OUT_CLASS;H_COLUMNGROUPLAYOUT_CLASS;H_FOOTERLAYOUT_CLASS;H_FRAMELAYOUT_CLASS;H
_FRAMEGROUPLAYOUT_CLASS;H_DROP_CAPLAYOUT_CLASS;H_GROUPLAYOUT_CLASS;H_HEADERLAY
OUT_CLASS;H_NOTELAYOUT_CLASS;H_PAGELAYOUT_CLASS;H_ROWGROUPLAYOUT_CLASS;H_ROWLA
YOUT_CLASS;H_RUBYLAYOUT_CLASS;H_SUPERTABLEGROUPLAYOUT_CLASS;H_SUPERTABLELAYOUT
_CLASS;H_TABLELAYOUT_CLASS;H_ENDNOTELAYOUT_CLASS;H_FOOTNOTELAYOUT_CLASS;H_TABLE
HEADINGLAYOUT_CLASS;H_TOCSUPERTABLELAYOUT_CLASS',0)} [See list of classes](#)

(Read-write) Allows you to expand a layout object to the bottom to accommodate its contents.

Data Type

Integer

Syntax

isexpanddownvalue = [objectreference].IsExpandDown

[objectreference].IsExpandDown = isexpanddownvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Word Pro: IsExpandLeft property

{button ,AL('H_LAYOUT_CLASS;H_CELLGROUPLAYOUT_CLASS;H_CELLLAYOUT_CLASS;H_CONNECTEDLAY
OUT_CLASS;H_COLUMNGROUPLAYOUT_CLASS;H_FOOTERLAYOUT_CLASS;H_FRAMELAYOUT_CLASS;H
_FRAMEGROUPLAYOUT_CLASS;H_DROP_CAPLAYOUT_CLASS;H_GROUPLAYOUT_CLASS;H_HEADERLAY
OUT_CLASS;H_NOTELAYOUT_CLASS;H_PAGELAYOUT_CLASS;H_ROWGROUPLAYOUT_CLASS;H_ROWLA
YOUT_CLASS;H_RUBYLAYOUT_CLASS;H_SUPERTABLEGROUPLAYOUT_CLASS;H_SUPERTABLELAYOUT
_CLASS;H_TABLELAYOUT_CLASS;H_ENDNOTELAYOUT_CLASS;H_FOOTNOTELAYOUT_CLASS;H_TABLE
HEADINGLAYOUT_CLASS;H_TOCSUPERTABLELAYOUT_CLASS',0)} [See list of classes](#)

(Read-write) Allows you to expand a layout object to the left to accommodate its contents.

Data Type

Integer

Syntax

isexpandleftvalue = [objectreference].IsExpandLeft

[objectreference].IsExpandLeft = isexpandleftvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Word Pro: IsExpandRight property

```
{button .AL('H_DIVISIONINFO_CLASS;H_LAYOUT_CLASS;H_CELLGROUPLAYOUT_CLASS;H_CELLLAYOUT_C  
LASS;H_CONNECTEDLAYOUT_CLASS;H_COLUMNGROUPLAYOUT_CLASS;H_FOOTERLAYOUT_CLASS;H_  
FRAMELAYOUT_CLASS;H_FRAMEGROUPLAYOUT_CLASS;H_DROP_CAPLAYOUT_CLASS;H_GROUPLAYOU  
T_CLASS;H_HEADERLAYOUT_CLASS;H_NOTELAYOUT_CLASS;H_PAGELAYOUT_CLASS;H_ROWGROUPL  
AYOUT_CLASS;H_ROWLAYOUT_CLASS;H_RUBYLAYOUT_CLASS;H_SUPERTABLEGROUPLAYOUT_CLASS  
;H_SUPERTABLELAYOUT_CLASS;H_TABLELAYOUT_CLASS;H_ENDNOTELAYOUT_CLASS;H_FOOTNOTEL  
AYOUT_CLASS;H_TABLEHEADINGLAYOUT_CLASS;H_TOCSUPERTABLELAYOUT_CLASS',0)} See list of  
classes
```

(Read-write) Allows you to expand a layout object to the right to accommodate its contents.

Data Type

Integer

Syntax

isexpandrightvalue = [objectreference].IsExpandRight

[objectreference].IsExpandRight = isexpandrightvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Word Pro: IsExpandUp property

```
{button ,AL('H_LAYOUT_CLASS;H_CELLGROUPLAYOUT_CLASS;H_CELLLAYOUT_CLASS;H_CONNECTEDLAY  
OUT_CLASS;H_COLUMNGROUPLAYOUT_CLASS;H_FOOTERLAYOUT_CLASS;H_FRAMELAYOUT_CLASS;H  
_FRAMEGROUPLAYOUT_CLASS;H_DROP_CAPLAYOUT_CLASS;H_GROUPLAYOUT_CLASS;H_HEADERLAY  
OUT_CLASS;H_NOTELAYOUT_CLASS;H_PAGELAYOUT_CLASS;H_ROWGROUPLAYOUT_CLASS;H_ROWLA  
YOUT_CLASS;H_RUBYLAYOUT_CLASS;H_SUPERTABLEGROUPLAYOUT_CLASS;H_SUPERTABLELAYOUT  
_CLASS;H_TABLELAYOUT_CLASS;H_ENDNOTELAYOUT_CLASS;H_FOOTNOTELAYOUT_CLASS;H_TABLE  
HEADINGLAYOUT_CLASS;H_TOCSUPERTABLELAYOUT_CLASS',0)} See list of classes
```

(Read-write) Allows you to expand a layout object upward to accommodate its contents.

Data Type

Integer

Syntax

isexpandupvalue = [objectreference].IsExpandUp

[objectreference].IsExpandUp = isexpandupvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Word Pro: IsExportedToNotesFX property

{button ,AL('H_BOOKMARK_CLASS',0)} [See list of classes](#)

(Read-write) Determines whether data is transmitted from a Notes field to a link in an embedded Word Pro document.

Data Type

Integer

Syntax

isexportedtonotesfxvalue = [objectreference].IsExportedToNotesFX

[objectreference].IsExportedToNotesFX = isexportedtonotesfxvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

[DdeLink]

This property is not used for a DdeLink object.

[Bookmark]

When you are working in an embedded Word Pro document and the document is opened, Notes can transmit data into a Word Pro bookmark. When you close the Word Pro document, the data updates the corresponding Notes field.

Word Pro: IsExternalFile property

{button ,AL(^H_DIVISIONINFO_CLASS;H_GRAPHICOLEBJECT_CLASS;H_GRAPHIC_CLASS;H_OLEOBJECT_CLASS',0)} [See list of classes](#)

(Read-write)

Data Type

Integer

Syntax

isexternalfilevalue = [objectreference].IsExternalFile

[objectreference].IsExternalFile = isexternalfilevalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Word Pro: IsFixedLength property

{button ,AL(`H_FOOTNOTESEPOPT_CLASS;H_FOOTNOTECONTSEP_CLASS;H_FOOTNOTESEPARATOR_CLASSES',0)} [See list of classes](#)

{button ,AL(`H_ISFIXEDLENGTH_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Determines whether a footnote or endnote separator is set to a custom length or spans to the page margin.

Data Type

[Integer](#)

Syntax

isfixedlengthvalue = [objectreference].IsFixedLength

[objectreference].IsFixedLength = isfixedlengthvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

If this property is set to True, the footnote separator spans to the page margin. If set to False, the footnote separator is set to a custom length. Equivalent to choosing either "Span to margin" or "Custom length" on the Separators panel of the Footnote and Endnote options dialog box.

Word Pro: IsFooter property

{button ,AL(^H_PAGECONTAINER_CLASS;H_SUBPAGECONTAINER_CLASS;H_SUPERPAGECONTAINER_CLASS',0)} [See list of classes](#)

{button ,AL(^H_ISFOOTER_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-only) Indicates whether or not a container has a footer object.

Data Type

[Integer](#)

Syntax

isfootervalue = [objectreference].IsFooter

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Word Pro: IsFrameCaptionOptions property

{button ,AL('H_TEXTDOCUMENT_CLASS',0)} [See list of classes](#)

(Read-only)

Indicates if the frame caption options have been set for the current document.

Data Type

[Integer](#)

Syntax

isframecaptionoptionsvalue = [objectreference].IsFrameCaptionOptions

Legal values

This property is read-only. The value of this property cannot be set by a script.

Usage

This property is used internally to determine whether to use the options already stored with the document or to use the default options provided by Word Pro.

Word Pro: IsFrameMenuEnabled property

{button ,AL('H_CONTEXTMENUOPTIONS_CLASS',0)} [See list of classes](#)

(Read-write) Controls whether or not the Frame menu displays when the cursor is within a frame object.

Data Type

Integer

Syntax

isframemenueabledvalue = [objectreference].IsFrameMenuEnabled

[objectreference].IsFrameMenuEnabled = isframemenueabledvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

A True setting allows the Frame menu to display when the focus is in a frame object. A False setting prevents the display of the Frame menu.

Word Pro: IsGotoable property

{button ,AL(^H_DIVISIONINFO_CLASS',0)} [See list of classes](#)

(Read-only)

Indicates if you can use Go To to go to a document division or part.

Data Type

[Integer](#)

Syntax

isgotoablevalue = [objectreference].IsGotoable

Legal values

This property is read-only. The value of this property cannot be set by a script.

Usage

Use this property to before you use any methods pertaining to Go To, especially if the document has OLE, external, hidden, or parent divisions.

Word Pro: IsGrabBar property

{button ,AL(^H_ICONBAR_CLASS',0)} [See list of classes](#)

(Write-only) Indicates if there is a solid color grab bar adjacent to the icons in a set. The grab bar allows you to drag the set to a different position on the workspace. Word Pro displays a hand over the solid color bar that closes when you click.

Data Type

Integer

Syntax

[objectreference].IsGrabBar = isgrabbarvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

You can write a script to turn this indicator off, but if you do so, the solid color grab bar disappears from the workspace.

Word Pro: IsGrammarError property

{button ,AL('H_ATTRIBUTES_CLASS',0)} [See list of classes](#)

(Read-write)

Data Type

Integer

Syntax

isgrammarerrorvalue = [objectreference].IsGrammarError

[objectreference].IsGrammarError = isgrammarerrorvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Word Pro: IsGraphicalObject property

{button ,AL('H_GRAPHICOLEBJECT_CLASS;H_GRAPHIC_CLASS;H_OLEOBJECT_CLASS',0)} [See list of classes](#)

(Read-only)

Indicates whether or not the OLE object from which you call this property is a graphic.

Data Type

Integer

Syntax

isgraphicalobjectvalue = [objectreference].IsGraphicalObject

Legal values

A value of -1 indicates that the OLE object is a graphic. A value of 0 indicates that the OLE object is not a graphic. This property is read-only. The value of this property cannot be set by a script.

Usage

All graphics except LotusChart objects return a -1. LotusChart and other OLE objects return a 0.

Some of the graphic types recognized by Word Pro include .BMP, .GIF, .PNG, .JPG, .TIF, .SDW, and .WMF.

Word Pro: IsGraphicMenuEnabled property

{button ,AL('H_CONTEXTMENUOPTIONS_CLASS',0)} [See list of classes](#)

(Read-write) Controls whether or not the Graphic menu displays when the focus includes a graphic.

Data Type

Integer

Syntax

isgraphicmenuenabledvalue = [objectreference].IsGraphicMenuEnabled

[objectreference].IsGraphicMenuEnabled = isgraphicmenuenabledvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

A True setting allows the Graphic menu to display when the focus is in a graphic object. A False setting prevents the display of the Graphic menu.

Word Pro: IsHangover property

{button ,AL('H_KINSOKU_CLASS',0)} [See list of classes](#)

(Read-write)

Data Type

Integer

Syntax

ishangovervalue = [objectreference].IsHangover

[objectreference].IsHangover = ishangovervalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Word Pro: IsHeader property

{button ,AL(^H_PAGECONTAINER_CLASS;H_SUBPAGECONTAINER_CLASS;H_SUPERPAGECONTAINER_CLASS,0)} [See list of classes](#)

{button ,AL(^H_ISHEADER_PROPERTY_EXSCRIPT,1)} [See example](#)

(Read-only) Indicates whether or not a container has a header object.

Data Type

[Integer](#)

Syntax

isheadervalue = [objectreference].IsHeader

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Word Pro: IsHiddenMark property

{button ,AL('H_ATTRIBUTES_CLASS',0)} [See list of classes](#)

{button ,AL('H_ISHIDDENMARK_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write)

Data Type

[Integer](#)

Syntax

ishiddenmarkvalue = [objectreference].IsHiddenMark

[objectreference].IsHiddenMark = ishiddenmarkvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Word Pro: IsHideHeaderFooter property

{button ,AL('H_WINVIEWPREFS_CLASS',0)} [See list of classes](#)

{button ,AL('H_ISHIDEHEADERFOOTER_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Determines whether a document's headers and footers are hidden.

Data Type

[Integer](#)

Syntax

ishideheaderfootervalue = [objectreference].IsHideHeaderFooter

[objectreference].IsHideHeaderFooter = ishideheaderfootervalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

A True setting is equivalent to disabling the View - Show/Hide - Headers & Footers option. A False setting enables the View - Show/Hide - Headers & Footers option.

Word Pro: IsHighlightNote property

{button ,AL('H_USERINTERFACEPREFS_CLASS',0)} [See list of classes](#)

(Read-write) Sets or returns whether comment note mode is active.

Data Type

[Integer \(Bool\)](#)

Syntax

ishighlightnotevalue = [objectreference].IsHighlightNote

[objectreference].IsHighlightNote = ishighlightnotevalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values. Default is False (0).

Usage

A True setting is equivalent to selecting the "Highlight and create a comment note" option in the Review & Comment Tools icon palette. A False setting is equivalent to deselecting the "Highlight and create a comment note" option.

Word Pro: IsHorizontalScrollBarCleanScrn property

{button ,AL('H_WINVIEWPREFS_CLASS',0)} [See list of classes](#)

{button ,AL('H_ISHORIZONTALSCROLLBARCLEANSCRN_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Determines whether horizontal scroll bars are displayed in clean screen mode.

Data Type

[Integer](#)

Syntax

ishorizontalscrollbarcleanscrnvalue = [objectreference].IsHorizontalScrollBarCleanScrn

[objectreference].IsHorizontalScrollBarCleanScrn = ishorizontalscrollbarcleanscrnvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Equivalent to the "Show horizontal scroll bar" option on the Clean Screen panel of the View Preferences dialog box. If this property is set to True, Word Pro displays horizontal scroll bars in clean screen mode. If set to False, Word Pro does not display horizontal scroll bars in clean screen mode.

{button ,AL('H_ISVIEWHORZSCROLLBAR_PROPERTY_MEMDEF',0)} [See related topics](#)

Word Pro: IsIconDepressible property

{button ,AL('H_ICONBARMANAGER_CLASS',0)} [See list of classes](#)

(WriteOnly) Indicates whether or not an icon depresses if you click down on it.

Data Type

Integer

Syntax

[objectreference].IsIconDepressible = isicondepressiblevalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values. Default is True (-1).

Usage

For this property to function, you must first select an icon using either the SelectStandardIcon or SelectCustomIcon method.

Word Pro: IsInBulletEditMode property

{button ,AL(^H_CLICKHERE_CLASS;H_TEXTMARKER_CLASS;H_TEXT_CLASS',0)} [See list of classes](#)

(Read-write) Indicates whether or not the current paragraph is in bullet edit mode.

Data Type

Integer

Syntax

isinbulleteditmodevalue = [objectreference].IsInBulletEditMode

[objectreference].IsInBulletEditMode = isinbulleteditmodevalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Word Pro: IsIndexMarkAble property

```
{button ,AL(^H_LAYOUT_CLASS;H_CELLGROUPLAYOUT_CLASS;H_CELLLAYOUT_CLASS;H_CONNECTEDLAY  
OUT_CLASS;H_COLUMNGROUPLAYOUT_CLASS;H_FOOTERLAYOUT_CLASS;H_FRAMELAYOUT_CLASS;H  
_FRAMEGROUPLAYOUT_CLASS;H_DROP_CAPLAYOUT_CLASS;H_GROUPLAYOUT_CLASS;H_HEADERLAY  
OUT_CLASS;H_NOTELAYOUT_CLASS;H_PAGELAYOUT_CLASS;H_ROWGROUPLAYOUT_CLASS;H_ROWLA  
YOUT_CLASS;H_RUBYLAYOUT_CLASS;H_SUPERTABLEGROUPLAYOUT_CLASS;H_SUPERTABLELAYOUT  
_CLASS;H_TABLELAYOUT_CLASS;H_ENDNOTELAYOUT_CLASS;H_FOOTNOTELAYOUT_CLASS;H_TABLE  
HEADINGLAYOUT_CLASS;H_TOCSUPERTABLELAYOUT_CLASS',0)} See list of classes
```

(Read-only)

This language element has not yet been defined. You can check for a newer Help file on the Lotus web site.

Data Type

Integer

Syntax

isindexmarkablevalue = [objectreference].IsIndexMarkAble

Legal values

This property is read-only. The value of this property cannot be set by a script.

Usage

Word Pro: IsIndex property

{button ,AL(^H_SECTION_CLASS;H_INDEXSECTION_CLASS',0)} [See list of classes](#)

(Read-only)

Data Type

Integer

Syntax

isindexvalue = [objectreference].IsIndex

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Word Pro: IsInDraft property

{button ,AL('H_WINVIEWPREFS_CLASS',0)} [See list of classes](#)

{button ,AL('H_ISINDRAFT_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Determines whether Word Pro displays in draft view.

Data Type

[Integer](#)

Syntax

isindraftvalue = [objectreference].IsInDraft

[objectreference].IsInDraft = isindraftvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Equivalent to the "Show draft" option on the Zoom panel of the View Preferences dialog box. If this property is set to True, Word Pro displays in draft view. If set to False, Word Pro does not display in draft view.

Word Pro: IsInOleDivision property

```
{button ,AL(^H_BASECONTAINER_CLASS;H_CELLCONTAINER_CLASS;H_FRAMECONTAINER_CLASS;H_NOTE  
CONTAINER_CLASS;H_RUBYCONTAINER_CLASS;H_DROPDOWNCONTAINER_CLASS;H_PAGECONTAINER_  
CLASS;H_SUBPAGECONTAINER_CLASS;H_SUPERPAGECONTAINER_CLASS;H_ROWCONTAINER_CLASS;  
H_SUPERTABLECONTAINER_CLASS;H_TABLECONTAINER_CLASS;H_PARALLELCOLSCONTAINER_CLASS  
;H_TABLEONLYCONT_CLASS',0)} See list of classes
```

(Read-only)

This language element has not yet been defined. You can check for a newer Help file on the Lotus web site.

Data Type

Integer

Syntax

isinoledivisionvalue = [objectreference].IsInOleDivision

Legal values

This property is read-only. The value of this property cannot be set by a script.

Usage

Word Pro: IsInOutline property

{button ,AL('H_WINVIEWPREFS_CLASS',0)} [See list of classes](#)

{button ,AL('H_ISINOUTLINE_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Determines whether Word Pro displays in outline view.

Data Type

[Integer](#)

Syntax

isinoutlinevalue = [objectreference].IsInOutline

[objectreference].IsInOutline = isinoutlinevalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Equivalent to the "Show outline" option on the Outline panel of the View Preferences dialog box. If this property is set to True, Word Pro displays in outline view. If set to False, Word Pro does not display in outline view.

Word Pro: IsInPageSort property

{button ,AL('H_WINVIEWPREFS_CLASS',0)} [See list of classes](#)

{button ,AL('H_ISINPAGESORT_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Determines whether Word Pro displays in page sorter view.

Data Type

[Integer](#)

Syntax

isinpagesortvalue = [objectreference].IsInPageSort

[objectreference].IsInPageSort = isinpagesortvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

A True setting is equivalent to toggling on the View - Page Sorter menu item. A False setting is equivalent to toggling off the View - Page Sorter menu item.

Word Pro: IsLastStop property

{button ,AL(^H_MAILROUTING_CLASS',0)} [See list of classes](#)

(Read-only) Indicates whether or not the document has reached the last stop on the mail route.

Data Type

Integer

Syntax

islaststopvalue = [objectreference].IsLastStop

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

The flag indicates True if the document has reached the last stop on the mail route, and False for all other stops.

Word Pro: IsLesser property

{button ,AL(^H_PARAGRAPHSTYLE_CLASS',0)} [See list of classes](#)

(Read-write)

Data Type

Integer

Syntax

islesservalue = [objectreference].IsLesser

[objectreference].IsLesser = islesservalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Word Pro: IsLocal property

```
{button ,AL(^H_CHARACTERSTYLE_CLASS;H_LAYOUT_CLASS;H_CELLGROUPLAYOUT_CLASS;H_CELLLAYO  
UT_CLASS;H_CONNECTEDLAYOUT_CLASS;H_COLUMNGROUPLAYOUT_CLASS;H_FOOTERLAYOUT_CLA  
SS;H_FRAMELAYOUT_CLASS;H_FRAMEGROUPLAYOUT_CLASS;H_DROPCAPLAYOOUT_CLASS;H_GROUPL  
AYOUT_CLASS;H_HEADERLAYOUT_CLASS;H_NOTELAYOUT_CLASS;H_PAGELAYOUT_CLASS;H_ROWGR  
OUPLAYOUT_CLASS;H_ROWLAYOUT_CLASS;H_RUBYLAYOUT_CLASS;H_SUPERTABLEGROUPLAYOUT_  
CLASS;H_SUPERTABLELAYOUT_CLASS;H_TABLELAYOUT_CLASS;H_ENDNOTELAYOUT_CLASS;H_FOOT  
NOTELAYOUT_CLASS;H_TABLEHEADINGLAYOUT_CLASS;H_TOCSUPERTABLELAYOUT_CLASS;H_PARAG  
RAPHSTYLE_CLASS',0)} See list of classes
```

(Read-write) Indicates whether a layout object is derived from the SmartMaster or is created locally.

Data Type

Integer

Syntax

islocalvalue = [objectreference].IsLocal

[objectreference].IsLocal = islocalvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

This property is used in the same way as when using this property from a CharacterStyle or ParagraphStyle object.

Word Pro: IsLockedForRevisions property

{button ,AL(^H_DIVISION_CLASS;H_TEXTDOCUMENT_CLASS',0)} [See list of classes](#)

(Read-write)

Data Type

Integer

Syntax

islockedforrevisionsvalue = [objectreference].IsLockedForRevisions

[objectreference].IsLockedForRevisions = islockedforrevisionsvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values. Default is False (0).

Usage

Word Pro: IsLocked property

{button ,AL(^H_LAYOUT_CLASS;H_CELLGROUPLAYOUT_CLASS;H_CELLLAYOUT_CLASS;H_CONNECTEDLAYOUT_CLASS;H_COLUMNGROUPLAYOUT_CLASS;H_FOOTERLAYOUT_CLASS;H_FRAMELAYOUT_CLASS;H_FRAMEGROUPLAYOUT_CLASS;H_DROPDOWNLAYOUT_CLASS;H_GROUPLAYOUT_CLASS;H_HEADERLAYOUT_CLASS;H_NOTELAYOUT_CLASS;H_PAGELAYOUT_CLASS;H_ROWGROUPLAYOUT_CLASS;H_ROWLAYOOUT_CLASS;H_RUBYLAYOUT_CLASS;H_SUPERTABLEGROUPLAYOUT_CLASS;H_SUPERTABLELAYOUT_CLASS;H_TABLELAYOUT_CLASS;H_ENDNOTELAYOUT_CLASS;H_FOOTNOTELAYOUT_CLASS;H_TABLEHEADINGLAYOUT_CLASS;H_TOCSUPERTABLELAYOUT_CLASS;H_CLICKHERE_CLASS',0)} [See list of classes](#)

(Read-write) Allows you to lock a CellLayout object to prevent it from being removed.

Data Type

[Integer](#)

Syntax

[objectreference].IsLocked = islockedvalue

islockedvalue = [objectreference].IsLocked

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

The IsLocked property applies only to cell layout objects.

When you enter a table cell in Word Pro, a cell layout object is created. If you leave the cell layout without modifying the layout or its content, Word Pro removes the cell layout object in order to conserve memory. A True setting prevents Word Pro from removing a specific cell layout object.

Word Pro: IsLotusChart property

{button ,AL('H_GRAPHICOLEBJECT_CLASS;H_GRAPHIC_CLASS;H_OLEOBJECT_CLASS',0)} [See list of classes](#)

{button ,AL('H_ISLOTUSCHART_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-only) Tests to see if a graphic's content is a Lotus chart object.

Data Type

[Integer](#)

Syntax

islotuschartvalue = [objectreference].IsLotusChart

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Word Pro: IsMarginSameAsParent property

```
{button ,AL('H_LAYOUT_CLASS;H_CELLGROUPLAYOUT_CLASS;H_CELLLAYOUT_CLASS;H_CONNECTEDLAYOUT_CLASS;H_COLUMNGROUPLAYOUT_CLASS;H_FOOTERLAYOUT_CLASS;H_FRAMELAYOUT_CLASS;H_FRAMEGROUPLAYOUT_CLASS;H_DROPDOWNLAYOUT_CLASS;H_GROUPLAYOUT_CLASS;H_HEADERLAYOUT_CLASS;H_NOTELAYOUT_CLASS;H_PAGELAYOUT_CLASS;H_ROWGROUPLAYOUT_CLASS;H_ROWLAYOUT_CLASS;H_RUBYLAYOUT_CLASS;H_SUPERTABLEGROUPLAYOUT_CLASS;H_SUPERTABLELAYOUT_CLASS;H_TABLELAYOUT_CLASS;H_ENDNOTELAYOUT_CLASS;H_FOOTNOTELAYOUT_CLASS;H_TABLEHEADINGLAYOUT_CLASS;H_TOCSUPERTABLELAYOUT_CLASS',0)} See list of classes
```

```
{button ,AL('H_ISMARGINSAMEASPARENT_PROPERTY_EXSCRIPT',1)} See example
```

(Read-write) Determines whether a layout object's margins are kept the same as the margins of its parent layout.

Data Type

[Integer](#)

Syntax

ismarginsameasparentvalue = [objectreference].IsMarginSameAsParent

[objectreference].IsMarginSameAsParent = ismarginsameasparentvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Allows you to control whether a child layout object maintains the same margin settings as its parent. For example, a header layout object usually adheres to its parent's margin values. A header layout object's parent is usually a page layout. If you set the header layout object's IsMarginSameAsParent property to False, the header layout object margins no longer correspond with the page layout object's margins.

Word Pro: IsMarginsInColor property

{button ,AL('H_WINVIEWPREFS_CLASS',0)} [See list of classes](#)

{button ,AL('H_ISMARGINSINCOLOR_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Determines whether Word Pro displays page margins in color.

Data Type

[Integer](#)

Syntax

ismarginsincolorvalue = [objectreference].IsMarginsInColor

[objectreference].IsMarginsInColor = ismarginsincolorvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Equivalent to the "Show margins in color" box on the Show panel of the View Preferences dialog box. If this property is set to True, Word Pro displays page margins in color. If set to False, Word Pro does not display page margins in color.

Word Pro: IsMarkerValid property

{button ,AL(^H_MARKER_CLASS;H_POWERFIELD_CLASS;H_CLICKHERE_CLASS;H_TEXTMARKER_CLASS;H_RUBYMARKER_CLASS;H_TABLEMARKER_CLASS',0)} [See list of classes](#)

(Read-only)

Data Type

Integer

Syntax

ismarkerValidvalue = [objectreference].IsMarkerValid

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Word Pro: IsMisspelled property

{button ,AL('H_ATTRIBUTES_CLASS',0)} [See list of classes](#)

(Read-write)

Data Type

Integer

Syntax

ismisspelledvalue = [objectreference].IsMisspelled

[objectreference].IsMisspelled = ismisspelledvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Word Pro: IsNoPrintZone property

{button ,AL('H_WINVIEWPREFS_CLASS',0)} [See list of classes](#)

{button ,AL('H_ISNOPRINTZONE_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write)

This language element has not yet been defined. You can check for a newer Help file on the Lotus web site.

Data Type

Integer

Syntax

isnoprntzonevalue = [objectreference].IsNoPrintZone

[objectreference].IsNoPrintZone = isnoprntzonevalue

Legal values

The legal values for this property are -1 and 0 but you may use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Word Pro: IsNotCopyable property

```
{button ,AL(^H_LAYOUT_CLASS;H_CELLGROUPLAYOUT_CLASS;H_CELLLAYOUT_CLASS;H_CONNECTEDLAY  
OUT_CLASS;H_COLUMNGROUPLAYOUT_CLASS;H_FOOTERLAYOUT_CLASS;H_FRAMELAYOUT_CLASS;H  
_FRAMEGROUPLAYOUT_CLASS;H_DROP_CAP_LAYOUT_CLASS;H_GROUPLAYOUT_CLASS;H_HEADERLAY  
OUT_CLASS;H_NOTELAYOUT_CLASS;H_PAGELAYOUT_CLASS;H_ROWGROUPLAYOUT_CLASS;H_ROWLA  
YOUT_CLASS;H_RUBY_LAYOUT_CLASS;H_SUPERTABLEGROUPLAYOUT_CLASS;H_SUPERTABLELAYOUT  
_CLASS;H_TABLELAYOUT_CLASS;H_ENDNOTE_LAYOUT_CLASS;H_FOOTNOTE_LAYOUT_CLASS;H_TABLE  
HEADING_LAYOUT_CLASS;H_TOCSUPERTABLELAYOUT_CLASS',0)} See list of classes
```

(Read-write) Determines whether or not a layout object can be copied.

Data Type

Integer

Syntax

isnotcopyablevalue = [objectreference].IsNotCopyable

[objectreference].IsNotCopyable = isnotcopyablevalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Setting a layout object's IsNotCopyable property to True prevents the layout from being copied. Setting it to False allows normal copy operations.

Word Pro: IsNotCopyImage property

{button ,AL(^H_GRAPHIC_CLASS',0)} [See list of classes](#)

(Read-only)

Data Type

Integer

Syntax

isnotcopyImagevalue = [objectreference].IsNotCopyImage

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Word Pro: IsNotGroupable property

```
{button ,AL(^H_LAYOUT_CLASS;H_CELLGROUPLAYOUT_CLASS;H_CELLLAYOUT_CLASS;H_CONNECTEDLAY  
OUT_CLASS;H_COLUMNGROUPLAYOUT_CLASS;H_FOOTERLAYOUT_CLASS;H_FRAMELAYOUT_CLASS;H  
_FRAMEGROUPLAYOUT_CLASS;H_DROP_CAPLAYOUT_CLASS;H_GROUPLAYOUT_CLASS;H_HEADERLAY  
OUT_CLASS;H_NOTELAYOUT_CLASS;H_PAGELAYOUT_CLASS;H_ROWGROUPLAYOUT_CLASS;H_ROWLA  
YOUT_CLASS;H_RUBYLAYOUT_CLASS;H_SUPERTABLEGROUPLAYOUT_CLASS;H_SUPERTABLELAYOUT  
_CLASS;H_TABLELAYOUT_CLASS;H_ENDNOTELAYOUT_CLASS;H_FOOTNOTELAYOUT_CLASS;H_TABLE  
HEADINGLAYOUT_CLASS;H_TOCSUPERTABLELAYOUT_CLASS',0)} See list of classes
```

(Read-write) Indicates whether or not a layout can be grouped with any sibling layout objects.

Data Type

Integer

Syntax

isnotgroupablevalue = [objectreference].IsNotGroupable

[objectreference].IsNotGroupable = isnotgroupablevalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

A True setting prevents a layout object from being grouped with sibling layout objects. For example, if you set this property to True for a certain frame layout object, you cannot group that frame with other frame layout objects.

Word Pro: IsNoUICommAllowed property

```
{button ,AL(^H_LAYOUT_CLASS;H_CELLGROUPLAYOUT_CLASS;H_CELLLAYOUT_CLASS;H_CONNECTEDLAY  
OUT_CLASS;H_COLUMNGROUPLAYOUT_CLASS;H_FOOTERLAYOUT_CLASS;H_FRAMELAYOUT_CLASS;H  
_FRAMEGROUPLAYOUT_CLASS;H_DROP_CAPLAYOUT_CLASS;H_GROUPLAYOUT_CLASS;H_HEADERLAY  
OUT_CLASS;H_NOTELAYOUT_CLASS;H_PAGELAYOUT_CLASS;H_ROWGROUPLAYOUT_CLASS;H_ROWLA  
YOUT_CLASS;H_RUBYLAYOUT_CLASS;H_SUPERTABLEGROUPLAYOUT_CLASS;H_SUPERTABLELAYOUT  
_CLASS;H_TABLELAYOUT_CLASS;H_ENDNOTELAYOUT_CLASS;H_FOOTNOTELAYOUT_CLASS;H_TABLE  
HEADINGLAYOUT_CLASS;H_TOCSUPERTABLELAYOUT_CLASS',0)} See list of classes
```

(Read-write) Indicates whether or not you are locked out of the user interface for a layout object.

Data Type

Integer

Syntax

isnouicomallowedvalue = [objectreference].IsNoUICommAllowed

[objectreference].IsNoUICommAllowed = isnouicomallowedvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

If a layout object's IsNoUICommAllowed property is set to True, you cannot open the InfoBox directly to the layout object's properties. However, you can access properties for a layout object if the InfoBox is already open when you enter the layout object.

For example, if a frame layout object's IsNoUICommAllowed property is set to True, the Frame Properties menu option is unavailable on both the right-click and pull-down menus. However, if the InfoBox is open when you enter the frame layout, you can change the focus of the InfoBox to the frame.

This property provides limited access control to the properties of a layout object. It is not meant to completely restrict access to the properties for a layout object.

Word Pro: IsOpen property

{button ,AL(^H_DOCUMENT_CLASS;H_TEXTDOCUMENT_CLASS',0)} [See list of classes](#)

(Read-only)

Data Type

Integer

Syntax

isopenvalue = [objectreference].IsOpen

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Word Pro: IsOverridden property

```
{button ,AL('H_LAYOUT_CLASS;H_CELLGROUPLAYOUT_CLASS;H_CELLLAYOUT_CLASS;H_CONNECTEDLAY  
OUT_CLASS;H_COLUMNGROUPLAYOUT_CLASS;H_FOOTERLAYOUT_CLASS;H_FRAMELAYOUT_CLASS;H  
_FRAMEGROUPLAYOUT_CLASS;H_DROP_CAPLAYOUT_CLASS;H_GROUPLAYOUT_CLASS;H_HEADERLAY  
OUT_CLASS;H_NOTELAYOUT_CLASS;H_PAGELAYOUT_CLASS;H_ROWGROUPLAYOUT_CLASS;H_ROWLA  
YOUT_CLASS;H_RUBYLAYOUT_CLASS;H_SUPERTABLEGROUPLAYOUT_CLASS;H_SUPERTABLELAYOUT  
_CLASS;H_TABLELAYOUT_CLASS;H_ENDNOTELAYOUT_CLASS;H_FOOTNOTELAYOUT_CLASS;H_TABLE  
HEADINGLAYOUT_CLASS;H_TOCSUPERTABLELAYOUT_CLASS;H_CLICKHERE_CLASS;H_TEXTMARKER_  
CLASS;H_TEXT_CLASS',0)} See list of classes
```

```
{button ,AL('H_ISOVERRIDDEN_PROPERTY_EXSCRIPT',1)} See example
```

(Read-only) Indicates whether or not the character styles or paragraph styles can be overridden by local attributes anywhere in the current paragraph.

Data Type

Integer

Syntax

isoverriddenvalue = [objectreference].IsOverridden

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

[Layout]

If this property indicates that the layout object is overridden, then the layout object either did not come from a SmartMaster, or the layout object was modified after the SmartMaster was last loaded. Check the StyleExceptions property to determine which attributes are overridden.

Word Pro: IsOverride property

```
{button ,AL(^H_LAYOUT_CLASS;H_CELLGROUPLAYOUT_CLASS;H_CELLLAYOUT_CLASS;H_CONNECTEDLAY  
OUT_CLASS;H_COLUMNGROUPLAYOUT_CLASS;H_FOOTERLAYOUT_CLASS;H_FRAMELAYOUT_CLASS;H  
_FRAMEGROUPLAYOUT_CLASS;H_DROP_CAPLAYOUT_CLASS;H_GROUPLAYOUT_CLASS;H_HEADERLAY  
OUT_CLASS;H_NOTELAYOUT_CLASS;H_PAGELAYOUT_CLASS;H_ROWGROUPLAYOUT_CLASS;H_ROWLA  
YOUT_CLASS;H_RUBYLAYOUT_CLASS;H_SUPERTABLEGROUPLAYOUT_CLASS;H_SUPERTABLELAYOUT  
_CLASS;H_TABLELAYOUT_CLASS;H_ENDNOTELAYOUT_CLASS;H_FOOTNOTELAYOUT_CLASS;H_TABLE  
HEADINGLAYOUT_CLASS;H_TOCSUPERTABLELAYOUT_CLASS',0)} See list of classes
```

(Read-write) Indicates whether or not a layout can override another layout.

Data Type

Integer

Syntax

isoverridevalue = [objectreference].IsOverride

[objectreference].IsOverride = isoverridevalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values. Default is False.

Usage

This property receives its instructions from the ConditionType property. IsOverride is currently only used when converting Ami Pro documents to Word Pro format.

Word Pro: IsPageBreakAfter property

{button ,AL('H_BREAKS_CLASS',0)} [See list of classes](#)

{button ,AL('H_ISPAGEBREAKAFTER_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Determines whether or not there will be a page break after the object from which this property is set.

Data Type

Integer

Syntax

ispagebreakaftervalue = [objectreference].IsPageBreakAfter

[objectreference].IsPageBreakAfter = ispagebreakaftervalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

For the breaks object in the Text class, this property represents the "Break page" setting in the "After paragraph" box in the Text InfoBox.

Word Pro: IsPageBreakBefore property

{button ,AL('H_BREAKS_CLASS',0)} [See list of classes](#)

{button ,AL('H_ISPAGEBREAKBEFORE_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Determines whether or not there will be a page break before the object from which this property is set.

Data Type

[Integer](#)

Syntax

ispagebreakbeforevalue = [objectreference].IsPageBreakBefore

[objectreference].IsPageBreakBefore = ispagebreakbeforevalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

For the breaks object in the Text class, this property represents the "Break page" setting in the "Before paragraph" box in the Text InfoBox.

Word Pro: IsPageBreakMarks property

{button ,AL('H_WINVIEWPREFS_CLASS',0)} [See list of classes](#)

{button ,AL('H_ISPAGEBREAKMARKS_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Determines whether Word Pro displays page break marks.

Data Type

[Integer](#)

Syntax

ispagebreakmarksvalue = [objectreference].IsPageBreakMarks

[objectreference].IsPageBreakMarks = ispagebreakmarksvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Equivalent to the "Page breaks" option in the "Show marks" box on the Show panel of the View Preferences dialog box. If this property is set to True, Word Pro displays page break marks. If set to False, Word Pro does not display page break marks.

Word Pro: IsPageBreakWithin property

{button ,AL('H_BREAKS_CLASS',0)} [See list of classes](#)

{button ,AL('H_ISPAGEBREAKWITHIN_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Determines whether or not a page break is allowed in the object from which this property is accessed.

Data Type

[Integer](#)

Syntax

ispagebreakwithinsidevalue = [objectreference].IsPageBreakWithin

[objectreference].IsPageBreakWithin = ispagebreakwithinsidevalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

For the breaks object in the Text class, the IsPageBreakWithin property represents the "Keep entire paragraph on same page" option in the Text InfoBox.

Word Pro: IsPageBreak property

```
{button ,AL('H_LAYOUT_CLASS;H_CELLGROUPLAYOUT_CLASS;H_CELLLAYOUT_CLASS;H_CONNECTEDLAY  
OUT_CLASS;H_COLUMNGROUPLAYOUT_CLASS;H_FOOTERLAYOUT_CLASS;H_FRAMELAYOUT_CLASS;H  
_FRAMEGROUPLAYOUT_CLASS;H_DROP_CAPLAYOUT_CLASS;H_GROUPLAYOUT_CLASS;H_HEADERLAY  
OUT_CLASS;H_NOTELAYOUT_CLASS;H_PAGELAYOUT_CLASS;H_ROWGROUPLAYOUT_CLASS;H_ROWLA  
YOUT_CLASS;H_RUBYLAYOUT_CLASS;H_SUPERTABLEGROUPLAYOUT_CLASS;H_SUPERTABLELAYOUT  
_CLASS;H_TABLELAYOUT_CLASS;H_ENDNOTELAYOUT_CLASS;H_FOOTNOTELAYOUT_CLASS;H_TABLE  
HEADINGLAYOUT_CLASS;H_TOCSUPERTABLELAYOUT_CLASS',0)} See list of classes
```

```
{button ,AL('H_ISPAGEBREAK_PROPERTY_EXSCRIPT',1)} See example
```

(Read-write) Indicates whether or not a page break can be forced to occur beneath a row layout.

Data Type

[Integer](#)

Syntax

ispagebreakvalue = [objectreference].IsPageBreak

[objectreference].IsPageBreak = ispagebreakvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Equivalent to the "Page break after row" setting on the Misc panel of the InfoBox for cell layout objects.

Word Pro: IsParagraphNumberingDown property

{button ,AL(`H_BASETABLE_CLASS;H_FOOTNOTETABLE_CLASS;H_PARALLELCOLUMNS_CLASS;H_GLOSSARY_CLASS;H_TABLEHEADING_CLASS;H_TABLE_CLASS',0)} [See list of classes](#)

{button ,AL(`H_ISPARAGRAPHNUMBERINGDOWN_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Indicates whether or not paragraph numbering restarts at the top of each table column.

Data Type

Integer

Syntax

isparagraphnumberingdownvalue = [objectreference].IsParagraphNumberingDown

[objectreference].IsParagraphNumberingDown = isparagraphnumberingdownvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Equivalent to the "Restart paragraph numbers on each column" setting on the Misc panel of the InfoBox for table objects.

Word Pro: IsParagraphParent property

{button ,AL(^H_CLICKHERE_CLASS;H_TEXTMARKER_CLASS;H_TEXT_CLASS',0)} [See list of classes](#)

(Read-only)

Data Type

Integer

Syntax

isparagraphparentvariable = [objectreference].IsParagraphParent

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Word Pro: IsParallelColumnsMenuEnabled property

{button ,AL('H_CONTEXTMENUOPTIONS_CLASS',0)} [See list of classes](#)

(Read-write) Controls whether or not a Columns menu displays when the cursor is within a parallel column.

Data Type

Integer

Syntax

isparallelcolumnsmenuenabledvalue = [objectreference].IsParallelColumnsMenuEnabled

[objectreference].IsParallelColumnsMenuEnabled = isparallelcolumnsmenuenabledvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

A True setting allows the Columns menu to display when the cursor is within a parallel columns object. A False setting prevents the display of the Columns menu when the cursor is within a parallel columns object.

Word Pro: IsPartOfGroup property

```
{button ,AL(^H_LAYOUT_CLASS;H_CELLGROUPLAYOUT_CLASS;H_CELLLAYOUT_CLASS;H_CONNECTEDLAY  
OUT_CLASS;H_COLUMNGROUPLAYOUT_CLASS;H_FOOTERLAYOUT_CLASS;H_FRAMELAYOUT_CLASS;H  
_FRAMEGROUPLAYOUT_CLASS;H_DROP_CAPLAYOÜT_CLASS;H_GROUPLAYOUT_CLASS;H_HEADERLAY  
OUT_CLASS;H_NOTELAYOUT_CLASS;H_PAGELAYOUT_CLASS;H_ROWGROUPLAYOUT_CLASS;H_ROWLA  
YOUT_CLASS;H_RUBYLAYOUT_CLASS;H_SUPERTABLEGROUPLAYOUT_CLASS;H_SUPERTABLELAYOUT  
_CLASS;H_TABLELAYOUT_CLASS;H_ENDNOTELAYOUT_CLASS;H_FOOTNOTELAYOUT_CLASS;H_TABLE  
HEADINGLAYOUT_CLASS;H_TOCSUPERTABLELAYOUT_CLASS',0)} See list of classes
```

(Read-only) Indicates whether or not a layout object is part of a group of layout objects.

Data Type

Integer

Syntax

ispartofgroupvalue = [objectreference].IsPartOfGroup

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Word Pro: IsPersistent property

{button ,AL('H_SCRIPTDATASET_CLASS;H_WPDATASET_CLASS',0)} [See list of classes](#)

(Read-only) Indicates whether the data set is saved with the document between sessions.

Data Type

Integer

Syntax

ispersistentvalue = [objectreference].IsPersistent

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

This property returns True if the data set is saved with the document between sessions. This property returns False if the data set is not saved with the document between sessions.

Usage

All data sets contained in a document are saved with the document. Data set objects in the WPAApplication object are not saved with the document.

Word Pro: IsPrePrintedForm property

{button ,AL(`H_PRINTSETTINGS_CLASS',0)} [See list of classes](#)

{button ,AL(`H_ISPREPRINTEDFORM_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Prevents protected text from being printed if "Allow editing of protected text" is deselected on the Other Protection panel of the TeamSecurity dialog box.

Data Type

[Integer](#)

Syntax

ispreprintedformvalue = [objectreference].IsPrePrintedForm

[objectreference].IsPrePrintedForm = ispreprintedformvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Equivalent to choosing File - Print, clicking Options and selecting "On preprinted form."

A True setting also prevents lines and background colors/patterns in column blocks, frames, headers/footers, pages, paragraphs, and table cells from printing.

{button ,AL(`H_ISPROTECTED_PROPERTY_MEMDEF',0)} [See related topics](#)

Word Pro: IsPrintable property

```
{button ,AL(^H_LAYOUT_CLASS;H_CELLGROUPLAYOUT_CLASS;H_CELLLAYOUT_CLASS;H_CONNECTEDLAY  
OUT_CLASS;H_COLUMNGROUPLAYOUT_CLASS;H_FOOTERLAYOUT_CLASS;H_FRAMELAYOUT_CLASS;H  
_FRAMEGROUPLAYOUT_CLASS;H_DROP_CAPLAYOUT_CLASS;H_GROUPLAYOUT_CLASS;H_HEADERLAY  
OUT_CLASS;H_NOTELAYOUT_CLASS;H_PAGELAYOUT_CLASS;H_ROWGROUPLAYOUT_CLASS;H_ROWLA  
YOUT_CLASS;H_RUBYLAYOUT_CLASS;H_SUPERTABLEGROUPLAYOUT_CLASS;H_SUPERTABLELAYOUT  
_CLASS;H_TABLELAYOUT_CLASS;H_ENDNOTELAYOUT_CLASS;H_FOOTNOTELAYOUT_CLASS;H_TABLE  
HEADINGLAYOUT_CLASS;H_TOCSUPERTABLELAYOUT_CLASS',0)} See list of classes
```

(Read-write) Indicates whether or not the layout prints.

Data Type

Integer

Syntax

isprintablevalue = [objectreference].IsPrintable

[objectreference].IsPrintable = isprintablevalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

This property does not affect the display or the flow of text. When a document prints, layout objects that have the IsPrintable property set to True leave a blank space equal to the size of the layout object.

Word Pro: IsPrintClickHereBlocks property

{button ,AL('H_PRINTSETTINGS_CLASS',0)} [See list of classes](#)

{button ,AL('H_ISPRINTCLICKHEREBLOCKS_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Determines whether or not unfilled ClickHere blocks that contain prompt text are printed.

Data Type

[Integer](#)

Syntax

isprintclickhereblocksvalue = [objectreference].IsPrintClickHereBlocks

[objectreference].IsPrintClickHereBlocks = isprintclickhereblocksvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Equivalent to choosing File - Print, clicking Options, and selecting "With unfilled click here blocks."

Word Pro: IsPrivate property

{button ,AL('H_CHARACTERSTYLE_CLASS;H_SILVERBULLET_CLASS;H_PARAGRAPHSTYLE_CLASS',0)} [See list of classes](#)

(Read-write) Indicates whether this character style can be seen as part of the selectable styles.

Data Type

[Integer](#)

Syntax

isprivatevalue = [objectreference].IsPrivate

[objectreference].IsPrivate = isprivatevalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Word Pro: IsPrompting property

{button ,AL(^H_CLICKHERE_CLASS',0)} [See list of classes](#)

(Read-write)

Data Type

Integer.

Syntax

[objectreference].IsPrompting = ispromptingvalue

ispromptingvalue = [objectreference].IsPrompting

Legal values**Usage**

Word Pro: IsProtected property

```
{button ,AL('H_DIVISION_CLASS;H_LAYOUT_CLASS;H_CELLGROUPLAYOUT_CLASS;H_CELLLAYOUT_CLASS;H_CONNECTEDLAYOUT_CLASS;H_COLUMNGROUPLAYOUT_CLASS;H_FOOTERLAYOUT_CLASS;H_FRAMELAYOUT_CLASS;H_FRAMEGROUPLAYOUT_CLASS;H_DROPAPLAYOUT_CLASS;H_GROUPLAYOUT_CLASS;H_HEADERLAYOUT_CLASS;H_NOTELAYOUT_CLASS;H_PAGELAYOUT_CLASS;H_ROWGROUPLAYOUT_CLASS;H_ROWLAYOUT_CLASS;H_RUBYLAYOUT_CLASS;H_SUPERTABLEGROUPLAYOUT_CLASS;H_SUPERTABLELAYOUT_CLASS;H_TABLELAYOUT_CLASS;H_ENDNOTELAYOUT_CLASS;H_FOOTNOTELAYOUT_CLASS;H_TABLEHEADINGLAYOUT_CLASS;H_TOCSUPERTABLELAYOUT_CLASS;H_TEXTDOCUMENT_CLASS',0)} See list of classes
```

```
{button ,AL('H_ISPROTECTED_PROPERTY_EXSCRIPT',1)} See example
```

[Layout]

(Read-write) Determines whether or not a layout object is marked as protected.

[Division]

(Read-write) Determines whether or not you can edit frames and table cells that are marked as protected.

Data Type

Integer

Syntax

isprotectedvalue = [objectreference].IsProtected

[objectreference].IsProtected = isprotectedvalue

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

[Layout]

This property is equivalent to protecting a layout object by selecting the "Protect" setting in the "Other options" box on the Misc panel of the InfoBox.

[Division]

This property is equivalent to the "Honor protectino on frames and table cells" option in the TeamSecurity dialog box.

Word Pro: IsRegistered property

{button ,AL(^H_MARKER_CLASS;H_POWERFIELD_CLASS;H_CLICKHERE_CLASS;H_TEXTMARKER_CLASS;H_RUBYMARKER_CLASS;H_TABLEMARKER_CLASS',0)} [See list of classes](#)

(Read-write)

Data Type

Integer

Syntax

isregisteredvalue = [objectreference].IsRegistered

[objectreference].IsRegistered = isregisteredvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Word Pro: IsRepeat property

{button ,AL('H_FOOTNOTEOPTIONS_CLASS',0)} [See list of classes](#)

{button ,AL('H_ISREPEAT_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Determines whether or not the footnote or endnote reference number is repeated when continued.

Data Type

[Integer](#)

Syntax

isrepeatvalue = [objectreference].IsRepeat

[objectreference].IsRepeat = isrepeatvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Equivalent to the "Repeat reference number when continued" option on the Continued Messages panel of the Footnote and Endnote Options dialog box.

Word Pro: IsReplaceable property

{button ,AL(^H_CONTENT_CLASS;H_GRAPHICOLEBJECT_CLASS;H_GRAPHIC_CLASS;H_OLEOBJECT_CLASSES;H_SUPERTABLE_CLASS;H_BASSETABLE_CLASS;H_FOOTNOTETABLE_CLASS;H_PARALLELCOLUMNS_CLASS;H_GLOSSARY_CLASS;H_TABLEHEADING_CLASS;H_TABLE_CLASS;H_TEXT_CLASS';0)} [See list of classes](#)

(Read-only) Indicates whether or not you can replace a content object with another content object.

Data Type

Integer

Syntax

isreplaceablevalue = [objectreference].IsReplaceable

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

This property is useful for importing content from other documents. Text is replaceable only if the content object is empty. If content is present, the content object is not replaceable. If only a ClickHere block, the content object can be replaced.

Word Pro: IsResetParagraphNumber property

{button ,AL(^H_BASETABLE_CLASS;H_FOOTNOTETABLE_CLASS;H_PARALLELCOLUMNS_CLASS;H_GLOSSARY_CLASS;H_TABLEHEADING_CLASS;H_TABLE_CLASS',0)} [See list of classes](#)

(Read-write) Indicates whether paragraph numbering is cumulative across cells within a table.

Data Type

Integer

Syntax

isresetparagraphnumbervalue = [objectreference].IsResetParagraphNumber

[objectreference].IsResetParagraphNumber = isresetparagraphnumbervalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

A True setting forces paragraph numbering to restart between each table cell. A False setting allows the IsParagraphNumberingDown property to control the circumstances in which paragraph numbering will reset.

Word Pro: IsRevisionMark property

{button ,AL(^H_CLICKHERE_CLASS;H_TEXTMARKER_CLASS;H_TEXT_CLASS',0)} [See list of classes](#)

(Read-only)

Data Type

Integer

Syntax

isrevisionmarkvalue = [objectreference].IsRevisionMark

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Word Pro: IsSame property

{button ,AL(^H_ICONBAR_CLASS',0)} [See list of classes](#)

(Write-only) Icon bar objects that have this flag checked are placed in the same location on the workspace as on the previous close. Moving just one of the bars with this flag set to True causes all bars that have this flag set to move.

Data Type

Integer

Syntax

[objectreference].IsSame = issamevalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Equivalent to the "Bars that will appear in the same location" setting in the SmartIcons Setup dialog box. All context icon bar objects have this flag set to True by default (except for the Universal set). For example, if you are working in text and move the Text icon bar object to the bottom of the workspace, and then start to work in a frame, the Frame icon bar object also displays on the bottom of the workspace.

Word Pro: IsSave property

{button ,AL(^H_ICONBAR_CLASS',0)} [See list of classes](#)

(Write-only) Tells you whether or not an icon bar object can be saved. If you move the icon bar to another location on the workspace or you add an icon to it, the icon bar object must be saved to a file in order to preserve your changes.

Data Type

[Integer](#)

Syntax

[objectreference].IsSave = issavevalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

If this flag is on when Word Pro closes, icon bar information is saved.

If the flag is turned off, no changes made to the icon bar will save. Therefore, if you move a bar or add or remove icons from a bar while Word Pro is active, none of this information is saved.

Word Pro: IsScalableBorder property

{button ,AL(^H_ICONBAR_CLASS',0)} [See list of classes](#)

(Write-only) Tells you whether or not you can change (scale) the size of an icon bar object.

Data Type

Integer

Syntax

[objectreference].IsScalableBorder = isscalablebordervalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values. Default is True (-1).

Usage

Set to True by default. If not turned on, you cannot drag the icon bar set to a new size.

Word Pro: IsScalable property

{button ,AL(^H_GRAPHICOLEBJECT_CLASS;H_GRAPHIC_CLASS;H_OLEOBJECT_CLASS',0)} [See list of classes](#)

(Read-only)

Data Type

[Integer](#)

Syntax

isscalablevalue = [objectreference].IsScalable

Legal values

True False

Usage

Word Pro: IsScrollable property

{button ,AL(^H_DIVISIONINFO_CLASS;H_LAYOUT_CLASS;H_CELLGROUPLAYOUT_CLASS;H_CELLAYOUT_CLASS;H_CONNECTEDLAYOUT_CLASS;H_COLUMNGROUPLAYOUT_CLASS;H_FOOTERLAYOUT_CLASS;H_FRAMELAYOUT_CLASS;H_FRAMEGROUPLAYOUT_CLASS;H_DROP_CAPLAYOUT_CLASS;H_GROUPLAYOUT_CLASS;H_HEADERLAYOUT_CLASS;H_NOTELAYOUT_CLASS;H_PAGELAYOUT_CLASS;H_ROWGROUPLAYOUT_CLASS;H_ROWLAYOUT_CLASS;H_RUBYLAYOUT_CLASS;H_SUPERTABLEGROUPLAYOUT_CLASS;H_SUPERTABLELAYOUT_CLASS;H_TABLELAYOUT_CLASS;H_ENDNOTELAYOUT_CLASS;H_FOOTNOTELAYOUT_CLASS;H_TABLEHEADINGLAYOUT_CLASS;H_TOCSUPERTABLELAYOUT_CLASS';0)} [See list of classes](#)

(Read-write) Determines whether or not a scroll bar is inserted in a comment note layout object.

Data Type

[Integer](#)

Syntax

isscrollablevalue = [objectreference].IsScrollable

[objectreference].IsScrollable = isscrollablevalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values. Default is False (0).

Usage

This property is only used with comment note layout objects.

If this property is set to False, the comment note does not have scroll bars. If set to True, the comment note does have scroll bars. Default is True.

Word Pro: IsShowing property

{button ,AL(^H_ICONBAR_CLASS;H_ICONBARMANAGER_CLASS;H_RULER_CLASS;H_SECTIONTABS_CLASS', 0)} [See list of classes](#)

(Read-write) Indicates whether a specific user interface object is currently visible.

[IconBar]

Indicates whether the icon bar object is currently visible.

[IconBarManager]

Indicates whether or not any icon bars are visible.

[Ruler]

Indicates whether or not the ruler object is currently visible.

[SectionTabs]

Indicates whether or not section tabs are visible within a specific document.

Data Type

[Integer](#)

Syntax

isshowingvalue = [objectreference].IsShowing

[objectreference].IsShowing = isshowingvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

[IconBarManager]

Equivalent to choosing View - Show/Hide - SmartIcons on the menu.

[SectionTabs]

Equivalent to clicking the tab icon at the right of the divider tab area to display or hide divider tabs. Assigning a value to this property affects only the current document. In order to display or hide divider tabs for all documents in Word Pro, modify the IsViewSectionTabs property of the WinViewPrefs class.

[Ruler]

Equivalent to choosing View - Show/Hide - Ruler.

Word Pro: IsSingleClickEntry property

```
{button ,AL('H_LAYOUT_CLASS;H_CELLGROUPLAYOUT_CLASS;H_CELLLAYOUT_CLASS;H_CONNECTEDLAY  
OUT_CLASS;H_COLUMNGROUPLAYOUT_CLASS;H_FOOTERLAYOUT_CLASS;H_FRAMELAYOUT_CLASS;H  
_FRAMEGROUPLAYOUT_CLASS;H_DROPDOWNLAYOUT_CLASS;H_GROUPLAYOUT_CLASS;H_HEADERLAY  
OUT_CLASS;H_NOTELAYOUT_CLASS;H_PAGELAYOUT_CLASS;H_ROWGROUPLAYOUT_CLASS;H_ROWLA  
YOUT_CLASS;H_RUBYLAYOUT_CLASS;H_SUPERTABLEGROUPLAYOUT_CLASS;H_SUPERTABLELAYOUT  
_CLASS;H_TABLELAYOUT_CLASS;H_ENDNOTELAYOUT_CLASS;H_FOOTNOTELAYOUT_CLASS;H_TABLE  
HEADINGLAYOUT_CLASS;H_TOCSUPERTABLELAYOUT_CLASS',0)} See list of classes
```

(Read-only) Indicates whether or not a layout can be entered with a single click.

Data Type

Integer

Syntax

issingleclickentryvalue = [objectreference].IsSingleClickEntry

[objectreference].IsSingleClickEntry = issingleclickentryvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Word Pro: IsSizable property

```
{button ,AL('H_LAYOUT_CLASS;H_CELLGROUPLAYOUT_CLASS;H_CELLLAYOUT_CLASS;H_CONNECTEDLAY  
OUT_CLASS;H_COLUMNGROUPLAYOUT_CLASS;H_FOOTERLAYOUT_CLASS;H_FRAMELAYOUT_CLASS;H  
_FRAMEGROUPLAYOUT_CLASS;H_DROP_CAPLAYOUT_CLASS;H_GROUPLAYOUT_CLASS;H_HEADERLAY  
OUT_CLASS;H_NOTELAYOUT_CLASS;H_PAGELAYOUT_CLASS;H_ROWGROUPLAYOUT_CLASS;H_ROWLA  
YOUT_CLASS;H_RUBYLAYOUT_CLASS;H_SUPERTABLEGROUPLAYOUT_CLASS;H_SUPERTABLELAYOUT  
_CLASS;H_TABLELAYOUT_CLASS;H_ENDNOTELAYOUT_CLASS;H_FOOTNOTELAYOUT_CLASS;H_TABLE  
HEADINGLAYOUT_CLASS;H_TOCSUPERTABLELAYOUT_CLASS',0)} See list of classes
```

```
{button ,AL('H_ISSIZABLE_PROPERTY_EXSCRIPT',1)} See example
```

(Read-write) Allows you to set whether or not sizing handles function on the layout object.

Data Type

[Integer](#)

Syntax

issizablevalue = [objectreference].IsSizable

[objectreference].IsSizable = issizablevalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

If a layout object's IsSizable property is set to False, you cannot size the layout object using the InfoBox or the mouse.

Word Pro: IsSizingViaMouse property

{button ,AL(^H_BASETABLE_CLASS;H_FOOTNOTETABLE_CLASS;H_PARALLELCOLUMNS_CLASS;H_GLOSSARY_CLASS;H_TABLEHEADING_CLASS;H_TABLE_CLASS',0)} [See list of classes](#)

(Read-write) Indicates whether or not you can use the mouse to size table columns.

Data Type

Integer

Syntax

issizingviamousevalue = [objectreference].IsSizingViaMouse

[objectreference].IsSizingViaMouse = issizingviamousevalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Word Pro: IsSmartCorrectEnabled property

{button ,AL('H_PREFERENCES_CLASS',0)} [See list of classes](#)

{button ,AL('H_ISSMARTCORRECTENABLED_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write)

Data Type

[Integer](#)

Syntax

issmartcorrectenabledvalue = [objectreference].IsSmartCorrectEnabled

[objectreference].IsSmartCorrectEnabled = issmartcorrectenabledvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Word Pro: IsSmartEditEnabled property

{button ,AL(^H_PREFERENCES_CLASS',0)} [See list of classes](#)

(Read-write)

Data Type

Integer.

Syntax

[objectreference].IsSmartEditEnabled = issmarteditenabledvalue

issmarteditenabledvalue = [objectreference].IsSmartEditEnabled

Legal values

Usage

Word Pro: IsSmartShadeEnabled property

{button ,AL('H_PREFERENCES_CLASS',0)} [See list of classes](#)

{button ,AL('H_ISSMARTSHADEENABLED_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write)

Data Type

[Integer](#)

Syntax

issmartshadeenabledvalue = [objectreference].IsSmartShadeEnabled

[objectreference].IsSmartShadeEnabled = issmartshadeenabledvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Word Pro: IsSnapTo property

```
{button ,AL('H_LAYOUT_CLASS;H_CELLGROUPLAYOUT_CLASS;H_CELLLAYOUT_CLASS;H_CONNECTEDLAY  
OUT_CLASS;H_COLUMNGROUPLAYOUT_CLASS;H_FOOTERLAYOUT_CLASS;H_FRAMELAYOUT_CLASS;H  
_FRAMEGROUPLAYOUT_CLASS;H_DROP_CAPLAYOUT_CLASS;H_GROUPLAYOUT_CLASS;H_HEADERLAY  
OUT_CLASS;H_NOTELAYOUT_CLASS;H_PAGELAYOUT_CLASS;H_ROWGROUPLAYOUT_CLASS;H_ROWLA  
YOUT_CLASS;H_RUBYLAYOUT_CLASS;H_SUPERTABLEGROUPLAYOUT_CLASS;H_SUPERTABLELAYOUT  
_CLASS;H_TABLELAYOUT_CLASS;H_ENDNOTELAYOUT_CLASS;H_FOOTNOTELAYOUT_CLASS;H_TABLE  
HEADINGLAYOUT_CLASS;H_TOCSUPERTABLELAYOUT_CLASS',0)} See list of classes
```

```
{button ,AL('H_ISSNAPTO_PROPERTY_EXSCRIPT',1)} See example
```

(Read-write) Indicates whether or not a child layout snaps to the grid of the current layout.

Data Type

[Integer](#)

Syntax

issnaptovalue = [objectreference].IsSnapTo

[objectreference].IsSnapTo = issnaptovalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Equivalent to the "Snap frames to grid" setting on the Misc panel of the InfoBox for certain layout objects.

```
{button ,AL('H_GRIDDISTANCE_PROPERTY_MEMDEF;H_GRIDTYPE_PROPERTY_MEMDEF',0)} See related  
topics
```

Word Pro: IsSortFromEnd property

{button ,AL(^H_SORTKEY_CLASS',0)} [See list of classes](#)

(Read-write) Specifies whether or not words or numbers are used in a sort level based on their relative position from the end of the field.

Data Type

[Integer](#)

Syntax

issortfromendvalue = [objectreference].IsSortFromEnd

[objectreference].IsSortFromEnd = issortfromendvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

A True value in this property indicates that the word used for the sort level is selected from the end of the field. A False value indicates that the word being used in the sort level is selected from the beginning of the field.

Usage

Equivalent to choosing Sort - Text, choosing "Other" from the "Word" box, and choosing "From end of field."

Word Pro: IsSpellBarUp property

{button ,AL(^H_DIVISION_CLASS;H_TEXTDOCUMENT_CLASS',0)} [See list of classes](#)

(Read-write)

Data Type

Integer

Syntax

isspellbarupvalue = [objectreference].IsSpellBarUp

[objectreference].IsSpellBarUp = isspellbarupvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values. Default is False (0).

Usage

Word Pro: IsSpellMode property

{button ,AL(^H_WINVIEWPREFS_CLASS',0)} [See list of classes](#)

(Read-write) Determines whether misspelled words are highlighted in your document.

Data Type

Integer

Syntax

isspellmodevalue = [objectreference].IsSpellMode

[objectreference].IsSpellMode = isspellmodevalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Displaying the spell check bar will automatically set this property to True. Closing the spell check bar will automatically set this property to False.

Word Pro: IsSqueeze property

{button ,AL(^H_KINSOKU_CLASS',0)} [See list of classes](#)

(Read-write)

Data Type

Integer

Syntax

issqueezevalue = [objectreference].IsSqueeze

[objectreference].IsSqueeze = issqueezevalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Word Pro: IsStyleSheet property

{button ,AL(^H_DIVISION_CLASS;H_TEXTDOCUMENT_CLASS',0)} [See list of classes](#)

(Read-write)

Data Type

Integer

Syntax

isstylesheetvalue = [objectreference].IsStyleSheet

[objectreference].IsStyleSheet = isstylesheetvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Word Pro: IsStyle property

```
{button ,AL(^H_LAYOUT_CLASS;H_CELLGROUPLAYOUT_CLASS;H_CELLLAYOUT_CLASS;H_CONNECTEDLAY  
OUT_CLASS;H_COLUMNGROUPLAYOUT_CLASS;H_FOOTERLAYOUT_CLASS;H_FRAMELAYOUT_CLASS;H  
_FRAMEGROUPLAYOUT_CLASS;H_DROP_CAPLAYOUT_CLASS;H_GROUPLAYOUT_CLASS;H_HEADERLAY  
OUT_CLASS;H_NOTELAYOUT_CLASS;H_PAGELAYOUT_CLASS;H_ROWGROUPLAYOUT_CLASS;H_ROWLA  
YOUT_CLASS;H_RUBYLAYOUT_CLASS;H_SUPERTABLEGROUPLAYOUT_CLASS;H_SUPERTABLELAYOUT  
_CLASS;H_TABLELAYOUT_CLASS;H_ENDNOTELAYOUT_CLASS;H_FOOTNOTELAYOUT_CLASS;H_TABLE  
HEADINGLAYOUT_CLASS;H_TOCSUPERTABLELAYOUT_CLASS',0)} See list of classes
```

(Read-only) Indicates whether a particular layout object is a style layout object.

Data Type

Integer

Syntax

[objectreference].IsStyle = isstylevalue

isstylevalue = [objectreference].IsStyle

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

You can reference this property to determine whether a particular layout object is a style layout object.

This property is useful when accessing layout object collections, since style layout objects are stored in collections, along with user-created layout objects. You should be aware that modifying style layout objects can sometimes cause unpredictable results.

For more information on style layout objects, see the Layout class description.

Word Pro: IsSymbolic property

{button ,AL(^H_FONTMETRICS_CLASS',0)} [See list of classes](#)

(Read-only)

Data Type

Integer.

Syntax

issymbolicvalue = [objectreference].IsSymbolic

Legal values**Usage**

Word Pro: IsTableHeading property

```
{button ,AL(^H_LAYOUT_CLASS;H_CELLGROUPLAYOUT_CLASS;H_CELLLAYOUT_CLASS;H_CONNECTEDLAY  
OUT_CLASS;H_COLUMNGROUPLAYOUT_CLASS;H_FOOTERLAYOUT_CLASS;H_FRAMELAYOUT_CLASS;H  
_FRAMEGROUPLAYOUT_CLASS;H_DROP_CAPLAYOUT_CLASS;H_GROUPLAYOUT_CLASS;H_HEADERLAY  
OUT_CLASS;H_NOTELAYOUT_CLASS;H_PAGELAYOUT_CLASS;H_ROWGROUPLAYOUT_CLASS;H_ROWLA  
YOUT_CLASS;H_RUBYLAYOUT_CLASS;H_SUPERTABLEGROUPLAYOUT_CLASS;H_SUPERTABLELAYOUT  
_CLASS;H_TABLELAYOUT_CLASS;H_ENDNOTELAYOUT_CLASS;H_FOOTNOTELAYOUT_CLASS;H_TABLE  
HEADINGLAYOUT_CLASS;H_TOCSUPERTABLELAYOUT_CLASS',0)} See list of classes
```

(Read-write) Allows you to make a row layout or row group layout object a table heading.

Data Type

Integer

Syntax

istableheadingvalue = [objectreference].IsTableHeading

[objectreference].IsTableHeading = istableheadingvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Equivalent to choosing Table - Mark as Repeated Heading.

Fulfillment Information

Refer to the following table when ordering the Application Developer's Documentation Set for SmartSuite 97.

- Identify the fulfillment center/centre for your country
- Copy the mailing address to the front of the fulfillment coupon
- Use the appropriate shipping and handling charge for your country

<i>Currency</i>	<i>Countries</i>	<i>Charge</i>	<i>Mailing address</i>
Australian Dollar	Australia	A\$35	Lotus Development Pty Ltd Customer Service Department Level 12, 321 Kent Street Sydney NSW 2000 Australia
Canadian Dollar	Canada	CS\$15	Lotus Development Corporation SmartSuite Documentation P.O. Box 670 Scarborough, Ontario M1K 5C5
Belgian Franc	Belgium	900BF	Lotus Assistance SARL
French Franc	France	150F	Parc Club Ariane
Lira	Italy	L. 4500	Bat. Neptune 5
Guilder	Netherlands	F 50	Bld des Chenes, BP 219
Escudo	Portugal	4500 Esc.	78051 St. Quentin en Yvelines Cedex
South African Rand	South Africa	R135	FRANCE
Peseta	Spain	3500Pts	
Austrian Schilling	Austria	300 OS	Lotus Development Gmbh
Deutchmark	Germany	45 DM	Baierbrunnerstrasse 35
Swiss Franc	Switzerland	SFr 35	Postfach 70 12 20 81379 Muenchen GERMANY
New Zealand Dollar	New Zealand	NZ\$40	Lotus Development New Zealand Ltd Customer Service Dept Level 20, ASB Bank Centre Cnr Albert & Wellesley Sts Auckland New Zealand
US Dollar	United States	\$10	Lotus Development Corporation SmartSuite Documentation P.O. Box 25367 Rochester NY 14625-0367 USA
Danish Krone	Denmark	Dkr 175	Lotus Development European Corporation
Markka	Finland	130 mk	Lotus Park
Punt	Ireland	IR£15	The Causeway
Norwegian Krone	Norway	Nkr 190	Staines Middlesex TW18 9AG
Krona	Sweden	200 Skr	ENGLAND
Sterling	United Kingdom	£15	
US Dollar	Others	US\$30.00	Lotus Development Corporation SmartSuite Documentation P.O. Box 25367 Rochester NY 14625-0367 USA

Australia

Lotus Development Pty Ltd
Customer Service Department
Level 12, 321 Kent Street
Sydney NSW 2000
Australia

Canada

Lotus Development Corporation
SmartSuite Documentation
P.O. Box 670
Scarborough, Ontario M1K 5C5

France

Lotus Assistance SARL
Parc Club Ariane
Bat. Neptune 5
Bld des Chenes, BP 219

78051 St. Quentin en Yvelines Cedex
FRANCE

Germany

Lotus Development Gmbh
Baierbrunnerstrasse 35
Postfach 70 12 20
81379 Muenchen
GERMANY

New Zealand

Lotus Development New Zealand Ltd
Customer Service Dept
Level 20, ASB Bank Centre
Cnr Albert & Wellesley Sts
Auckland New Zealand

United Kingdom (U.K.)

Lotus Development European Corporation
Lotus Park
The Causeway
Staines Middlesex TW18 9AG
ENGLAND

United States

Lotus Development Corporation
SmartSuite Documentation
P.O. Box 25367
Rochester NY 14625-0367

<i>Country</i>	<i>Currency</i>	<i>SHCharge</i>	<i>Center</i>
Australia	Australian Dollar	A\$30.00	Australia
Austria	Austrian Schilling	30.00 ÖS	Germany
Belgium	Belgian Franc	30.00 BF	France
Canada	Canadian Dollar	C\$10.00	Canada
Denmark	Danish Krone	Dkr 30.00	U.K.
Eastern Europe	US Dollar	\$30.00	
Finland	Markka	30.00 mk	U.K.
France	French Franc	30.00 F	France
Germany	Deutschmark	30.00 DM	Germany
Ireland	Punt	IR£30.00	U.K.
Italy	Lira	L. 30	France
Luxembourg	Luxembourg Franc	30.00 LF	France
Netherlands	Guilder	F 30.00	France
New Zealand	New Zealand Dollar	NZ\$30.00	New Zealand
Norway	Norwegian Krone	Nkr 30.00	U.K.
Portugal	Escudo	30.00 Esc.	France
South Africa	South African Rand	R30.00	France

Spain	Peseta	30 Pts	France
Sweden	Krona	30.00 Skr	U.K.
Switzerland	Swiss Franc	SFr 30.00	Germany
United States	US Dollar	\$10.00	United States
U.K.	Sterling	£30.00	U.K.

Developing SmartSuite Applications

Developing SmartSuite Applications Using LotusScript is available in the SmartSuite CD package as an online book. To install *Developing SmartSuite Applications Using LotusScript*, see the SmartSuite installation instructions.

To order a printed version of *Developing SmartSuite Applications Using LotusScript* and other LotusScript user assistance complete the [order form](#) and return it to Lotus.

Ordering the LotusScript Documentation Set

To order the *SmartSuite Application Developer's Kit*, complete the following form and mail or fax it to Lotus. You will receive the *SmartSuite Application Developer's Kit* within 21 days.

Ordering by US Mail

Mail the completed form to:

Lotus Development Corporation
55 Cambridge Parkway
Cambridge, MA 02142

Ordering by FAX

Fax the completed form to:

(617) 537-8500

Order Form

Name: _____

Title: _____

Mailing address: _____

E-mail address: _____

Product Registration Number: _____

LotusScript Documentation as Online Books

You can use the SmartSuite Custom Install to install the following online books about LotusScript:

- *Getting the Most Out of LotusScript in SmartSuite 97*
- *Developing SmartSuite Applications Using LotusScript*
- *LotusScript Language Reference*
- *LotusScript Programmer's Guide*

For more information about installing online books, see your SmartSuite installation documentation.

LotusScript Documentation on the Web

You can view updated versions of LotusScript documentation or download sample applications or Help files from the LotusScript home page.

Enter the following URL in the location field in your browser and press ENTER:

`www.lotus.com/home.nsf/welcome/sswin`

Script Design Basics

LotusScript provides a variety of tools and services to support you in developing applications for SmartSuite. Getting productive in a new programming environment often involves understanding how all the pieces work together -- the tools, the language conventions, the object dependencies, and so on. Understanding how to approach the problem and where to enter your script code is half the challenge in learning.

Choosing a place to begin

Lotus Notes, 1-2-3, Approach, Freelance Graphics, and Word Pro all use the same underlying LotusScript language. Each product implements LotusObjects on top of the LotusScript language. To determine what product best supports the goals for your script application, consider using each of the SmartSuite products and reviewing its features. Read *Developing SmartSuite Applications Using LotusScript* for overviews of what each product can bring to your programming effort. Implement a few simple procedures in each product to get a feel for its features and objects. In the long run, you'll be better able to determine what product provides strengths where you need them most, and how you can develop cross-product applications that take advantage of the strengths of each product.

Working the basics

LotusScript applications share the following common features.

- You need a Lotus product to run script applications.
- You need a Lotus product to store scripts in a product document such as a 1-2-3 workbook or Word Pro document.
- You need to run the Lotus Integrated Development Environment (IDE) to edit and debug scripts stored in a product document.
- You need to open an IDE window for each product document containing scripts that you want to modify.

To write a basic script application, therefore, you must run a Lotus product and load a document in that product. You can then write scripts for the product objects that you create in your product.

Writing scripts in the Integrated Development Environment (IDE)

Your primary tool for developing script applications is the Lotus Integrated Development Environment (IDE). Beyond providing the basic tools such as an editor, a debugger, a browser, and a dialog editor, the IDE provides a high degree of integration with each Lotus product. It is easy to move between tasks that you perform in a product and those that you perform in the IDE.

Writing global scripts

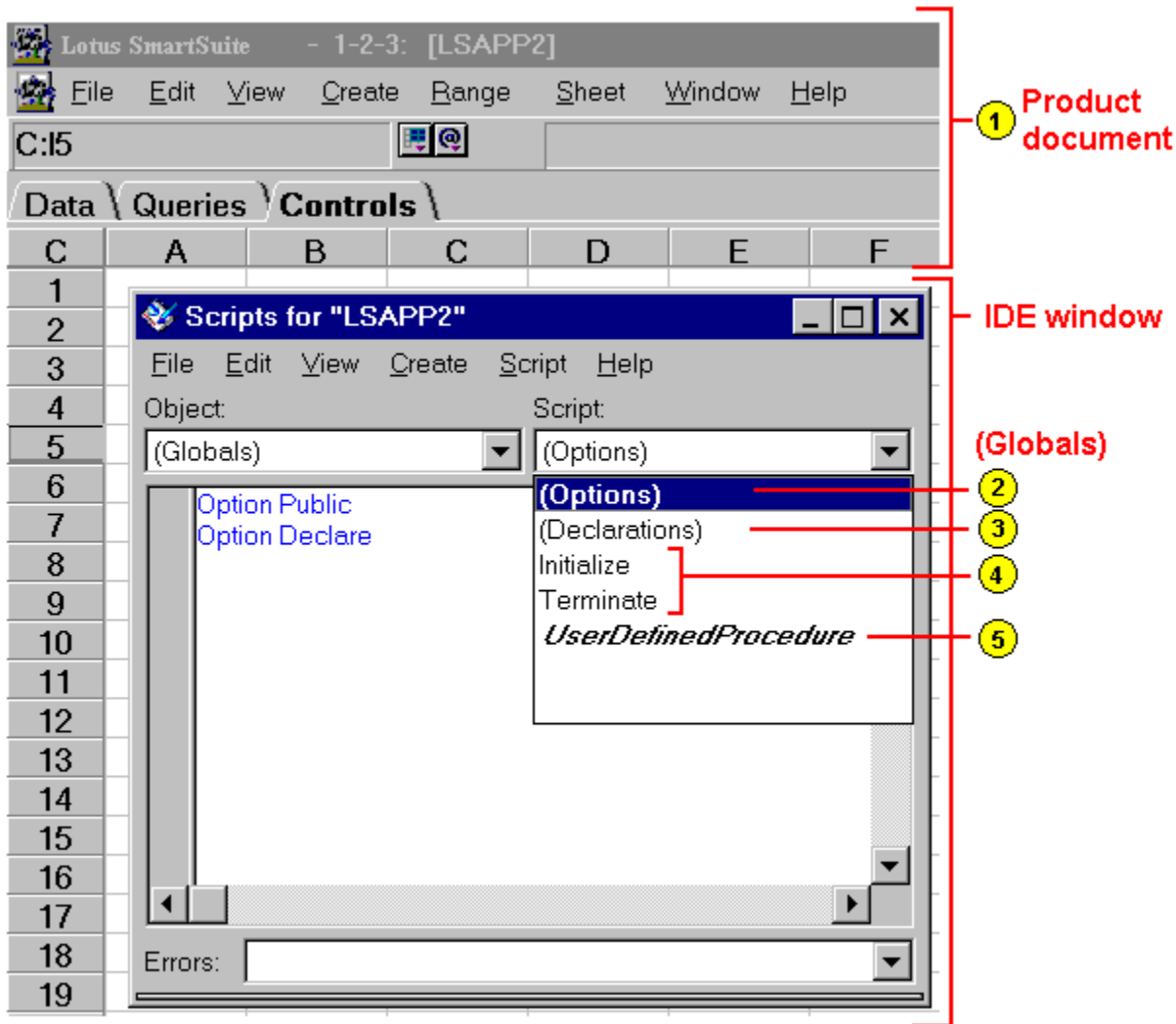
Global scripts make declarations, options, and procedures available to all scripts in your document. For example, to write global scripts for a 1-2-3 document named LSAPP2.123, you must first run 1-2-3, load the document LSAPP2.123, and then open an IDE window for that document. Choose Edit - Scripts & Macros - Show Script Editor in the 1-2-3 menu to activate an IDE window for your current document.

The IDE lists objects that you can script in the Object list and scripts for each of those objects in the Script list. You can add statements to predefined scripts in (Globals), such as (Options), (Declarations), Initialize, or Terminate, or you can create your own named procedures. You do not need to modify predefined scripts to write a basic script application.

The following illustration shows how to select a particular script for (Globals).

Click any item in the following list to learn more about it.

- | | | | |
|----------|--|----------|---|
| 1 | Product document | 4 | Initialize and Terminate subs |
| 2 | (Options) scripts | 5 | User-defined procedures |
| 3 | (Declarations) scripts | | |



Writing scripts for product objects

You can also write scripts for product objects in your document. As with (Globals), you can add statements in the predefined scripts for an object, or create new procedures for that object. Unlike scripts that you write in (Globals), the declarations, options statements, and procedures that you write for a product object are not generally available to scripts attached to a different product object.

The predefined scripts for product objects include object event procedures. Script statements in an object event procedure are executed when an object, such as a button, receives a particular event in your product, such as being clicked, double-clicked, or moved. For example, if you add a button named Button 5 to the 1-2-3 document LSAPP2.123, and you want it to run some script when you click it, you must add script statements to the Click procedure for Button 5. To select this event procedure, choose the Button 5 object in the IDE Object list and choose Click in the Script list.

The following illustration shows how to select a predefined or user-defined script for a 1-2-3 product object named Button 5.

Click any item in the following list to learn more about it.

- | | |
|--|--|
| 1 User-defined procedures | 4 Event procedures |
| 2 (Options) scripts | 5 Initialize and Terminate subs |
| 3 (Declarations) scripts | |

The screenshot shows the Lotus SmartSuite interface with a spreadsheet containing a 'Button 5' object. A dialog box titled 'Scripts for "LSAPP2"' is open, displaying the script for the selected object. The script is as follows:

```

Sub UserDefinedSub
  Dim AppName As String
  AppName = "WeeklyBud
  Call AppSetup1(AppNam
End Sub
  
```

The dialog box also shows a list of event handlers on the right side, with red lines and yellow circles with numbers 1-5 pointing to specific elements:

- 1: Points to the 'UserDefinedSub' script name in the 'Script' dropdown.
- 2: Points to the '(Options)' section.
- 3: Points to the '(Declarations)' section.
- 4: Points to the 'Click' event handler.
- 5: Points to the 'Initialize' event handler.

Red lines also connect the 'Button 5' object in the spreadsheet to the 'Object' dropdown in the dialog box, and the 'Click' event handler to the script code.

Working with external script files

In many cases, the one-application-per-document approach is sufficient for working with objects and data in isolated documents. To develop more sophisticated applications that reuse important scripts or use multiple products, consider using the following types of external script files:

[LotusScript Script \(LSS\) files](#)

[LotusScript Object \(LSO\) files](#)

[LotusScript Extension \(LSX\) files](#)

[OLE Custom Control \(OCX\) files](#)

[Dynamic-link Library \(DLL\) files](#)

Dynamic-link Library (DLL) files

If you develop useful functions in C and compiled them in a Dynamic-link Library (DLL), you can call them from your LotusScript application. For example, the following procedure declares and calls a LotusScript function named SendDLL, corresponding to a C function named _SendExportedRoutine in the DLL file named MYEXPORTS.DLL.

```
Declare Function SendDLL Lib _  
    "C:\LOTUS\ADDINS\MYEXPORTS.DLL" _  
    Alias "_SendExportedRoutine" (i1 As Long, i2 As Long)  
SendDLL(5, 10)
```

For more information on using Dynamic-link Libraries, see *LotusScript Language Reference*.

(Declarations) scripts in (Globals)

The (Declarations) script is designed to contain the following statements:

- Dim statements for variables that you want to be available to all scripts in your document
- Public, Private, Type, Class, and Declare Lib statements (external C calls)
- Const statements for those constants that you want to be available to all scripts in your document and are not needed for Use or UseLSX statements in (Options)

By default, the (Declarations) script is initially empty.

If you enter Type, Class, or Declare Lib statements in any other script in (Globals), the IDE moves them to (Declarations) automatically. If you enter Dim, Public, Private, or Const statements outside the scope of a procedure in another script, the IDE moves them to (Declarations) automatically. Const statements in (Options) are the exception to this rule.

Initialize and Terminate subs in (Globals)

Initialize script

Use the Initialize sub in (Globals) to initialize variables that you declared in (Declarations). The Initialize sub executes before any of these variables are accessed and before any other scripts in (Globals) are executed. By default, the Initialize script is empty.

Terminate script

Use the Terminate sub in (Globals) to clean up variables that you declared in (Declarations) when you close your document, or when you modify a script and execute it again. For example, you can use an Open statement to open a file containing data in Initialize, and use a Close statement in Terminate to close it. By default, the Terminate script is empty.

(Options) scripts in (Globals)

The (Options) script in (Globals) is designed to contain these the following statements:

- Option statements
Note (Options) contains the statement, Option Public, by default. This makes Const, Dim, Type, Class, Sub, Function, and Property statements public by default. You can use the Public form of these statements to make them public explicitly, or the Private form to make them unavailable to other scripts outside (Globals).
- Deftype statements
- Use and UseLSX statements
- Const statements needed for Use and UseLSX statements

If you enter any of these statements, except for Const, in any other script in (Globals), the IDE automatically moves them to (Options).

Option and Deftype statements that you enter in (Options) apply only to scripts for the current object. To make certain that an option is applied consistently throughout your document, enter the appropriate statement in the (Options) script for every object for which you are writing scripts.

User-defined procedures in (Globals)

While you are working in (Globals), you can add procedures to make them available throughout your document. There are three ways to add procedures to (Globals) in the IDE:

- *Using the IDE menu:* Choose Create - New Sub or Create - New Function in the IDE menu to create new subs and functions in (Globals). The IDE automatically adds the name of the new procedure to the Script list.
- *Entering statements:* Enter a Sub, Function, or Property statement anywhere in (Globals) except within a class. The IDE automatically adds the name of the new procedure to the Script list for (Globals).
- *Importing procedures from a file:* Use File - Import Script in the IDE menu to import scripts when you are working in (Globals). These imported scripts will be available to all scripts in your document. The IDE automatically adds the name of any new procedures contained in the imported script to the Script list.

LotusScript User Assistance

To help you learn how to develop LotusScript applications for SmartSuite, Lotus provides a complete library of user assistance.

Getting the Most Out of LotusScript in SmartSuite 97

This publication explains how SmartSuite 97 products use the LotusScript programming language and how your business can take advantage of LotusScript in developing applications for SmartSuite.

Getting the Most Out of LotusScript in SmartSuite 97 is available in hard copy, Adobe Acrobat, or HTML formats in your SmartSuite 97 package, in the [SmartSuite Application Developer's Kit](#), or on the [Worldwide Web](#).

Developing SmartSuite Applications Using LotusScript

This publication provides comprehensive information on key concepts and techniques for developing LotusScript applications. *Developing SmartSuite Applications Using LotusScript* focuses on programming tools, cross-application programming, Lotus Notes integration, and product-specific application development.

Developing SmartSuite Applications Using LotusScript is available in hard copy, Adobe Acrobat, or HTML formats in your SmartSuite 97 package, in the [SmartSuite Application Developer's Kit](#), or on the [Worldwide Web](#).

LotusScript Language Reference

This publication provides a comprehensive summary of conventions and basic commands for the LotusScript language. *LotusScript Language Reference* provides the foundation for programming any product that supports the LotusScript programming language.

LotusScript Language Reference is available in hard copy, Adobe Acrobat, Help, or HTML formats in your SmartSuite package, in the [SmartSuite Application Developer's Kit](#), or on the [Worldwide Web](#).

LotusScript Programmer's Guide

This publication is a general introduction to LotusScript that describes basic building blocks in the language and explains how to use them to create powerful applications.

LotusScript Programmer's Guide is available in hard copy, Adobe Acrobat, or HTML formats in your SmartSuite package, in the [SmartSuite Application Developer's Kit](#), or on the [Worldwide Web](#).

Class Reference Help and Frequently-asked Questions

Each product provides comprehensive Help on product classes, frequently-asked questions about programming, and code examples. All this is delivered in an innovative Help system designed to enhance your work as a programmer.

Class reference Help and frequently-asked questions are available in Help or HTML formats in your SmartSuite 97 package, in the [SmartSuite Application Developer's Kit](#), or on the [Worldwide Web](#).

Example code and sample applications

Most products also provide working code to illustrate important programming techniques. You can reuse and modify this code as you develop your own applications.

Example code is available in the SmartSuite CD-ROM package, in the [SmartSuite Application Developer's Kit](#), and on the [Worldwide Web](#).

LotusScript Object (LSO) files

LotusScript Object (LSO) files contain public definitions that you can use in your script applications. If you develop a library of commonly-used declarations or procedures that you want to reuse across multiple script applications, you can collect them in a product document, and use the File - Export Globals as LSO menu command to create a compiled LotusScript Object file. If this file were named WKREPORT.LSO, you would make these public definitions available to your script application by entering the following statement in the appropriate (Options) script:

```
Use "C:\LOTUS\ADDINS\WKREPORT.LSO"
```

For more information on using LotusScript Object files, see *LotusScript Language Reference*.

LotusScript Script (LSS) files

LotusScript Script (LSS) files are text files that contain LotusScript statements. You can create LSS files in any text editor. Use the %Include directive anywhere in a script to reference the contents of an LSS file. For example, to include the contents of a LotusScript Script file named STDSETUP.LSS in your application, enter the following statement:

```
%Include "C:\MYSCRIPTS\STDSETUP.LSS"
```

By default, LotusScript assumes that the LotusScript Script files referenced have an LSS file extension. You can actually use any extension for your text file or no extension at all.

For more information on using LotusScript Script files, see *LotusScript Language Reference*.

LotusScript Extension (LSX) files

LotusScript Extension (LSX) files are Dynamic-link Libraries (DLLs) that contain public class definitions. LSX files are developed using the Lotus LSX Toolkit. To obtain a version of the LSX Toolkit for your operating system, connect to the Lotus home page on the World Wide Web. Lotus ships LSX files for Lotus Notes and Approach; other LSX files are being developed for SmartSuite products by Lotus and by third-party developers. These extension files expand the range of classes that you can use in your LotusScript applications.

Tip You can enter a UseLSX statement in any script; the IDE automatically moves it to (Options).

Loading and using class definitions in LSX files

There are two ways to load and use the public class definitions in an LSX file.

- If the LSX file that you want to load is not registered in the Registry, you must refer to the LSX file directly in your UseLSX statement.

```
UseLSX "C:\MYSRIPTS\LSX4DB2.DLL"
```

- If an LSX is registered and you want to reference a class definition directly, you can enter the name of the class definition.

```
UseLSX "ObjectName"
```

In this example, LotusScript searches all entries under "LotusScriptExtensions" in the Registry for the specified class definition, and loads that definition.

Note If the LSX file you want to load is registered in the Windows Registry, you can reference its Registry name and have Windows provide the appropriate DLL name and file path. SmartSuite 97 registers an LSX file that contains Notes public class definitions. To use these Notes class definitions in your cross-product script applications, enter the following statement:

```
UseLSX "*Notes"
```

Viewing class definitions

Once you run a script containing a UseLSX statement and loaded an LSX file, you can browse its class definitions in the IDE Browser panel.

For more information on using LotusScript Extension files, see *LotusScript Language Reference*.

(Declaration) scripts in object scripts

The (Declarations) script for an object is designed to contain the following statements:

- Dim statements for variables that you want to be available to all scripts for the current object
- Const statements for those constants that you want to be available to all scripts for the current object and that are not needed for Use or UseLSX statements in (Options)

By default, the (Declarations) script is initially empty.

Event procedures in object scripts

If you are writing a script for an object, the Script list displays default event procedures for the selected object. In the IDE, you cannot create new event procedures for an existing product object because valid events for that object are defined by the product.

Initialize and Terminate subs in object scripts

Initialize sub

Use the Initialize sub to set up variables declared in the object's (Declarations) script. The Initialize sub for an object executes before any of its event procedures. By default, the Initialize script is empty.

Note Scripts for controls created in the Lotus Dialog Editor do not have Initialize subs.

Terminate sub

Use the Terminate sub to clean up variables that you declared in the object's (Declarations) script. By default, the Terminate script is empty.

Note Scripts for controls created in the Lotus Dialog Editor do not have Terminate subs.

(Options) scripts in object scripts

The (Options) script for an object is designed to contain these the following statements:

- Option statements
- *Deftype* statements
- Use and UseLSX statements
- Const statements needed for Use and UseLSX statements

User-defined procedures in object scripts

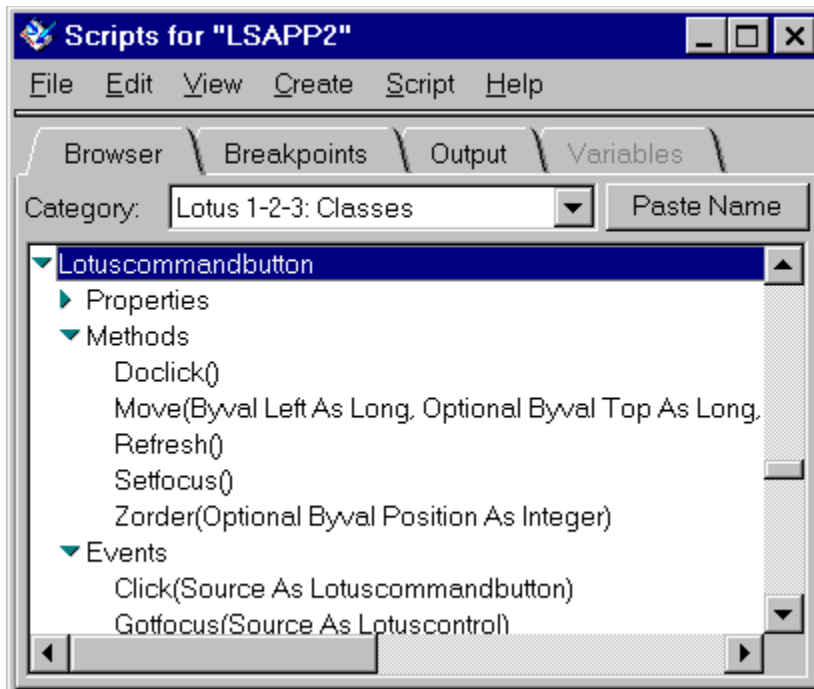
You can create other named subs, functions, and properties for objects, in addition to the predefined scripts or event procedures. Because these procedures are not in (Globals), they can be called only from other scripts for the object.

There are three ways to create object scripts in the IDE:

- *Using the IDE menu:* Use Create - New Sub and Create - New Function to create new subs and functions for an object. The IDE automatically adds the name of the new procedure to the Script list for that object.
- *Entering statements:* Enter a Sub, Function, or Property statement anywhere in a script for the current object. The IDE automatically adds the name of the new procedure to the Script list for that object.
- *Importing procedures from a file:* Use File - Import Script when you are working with object scripts to import scripts for that object. The IDE automatically adds the name of any new procedures contained in the imported script to the Script list.

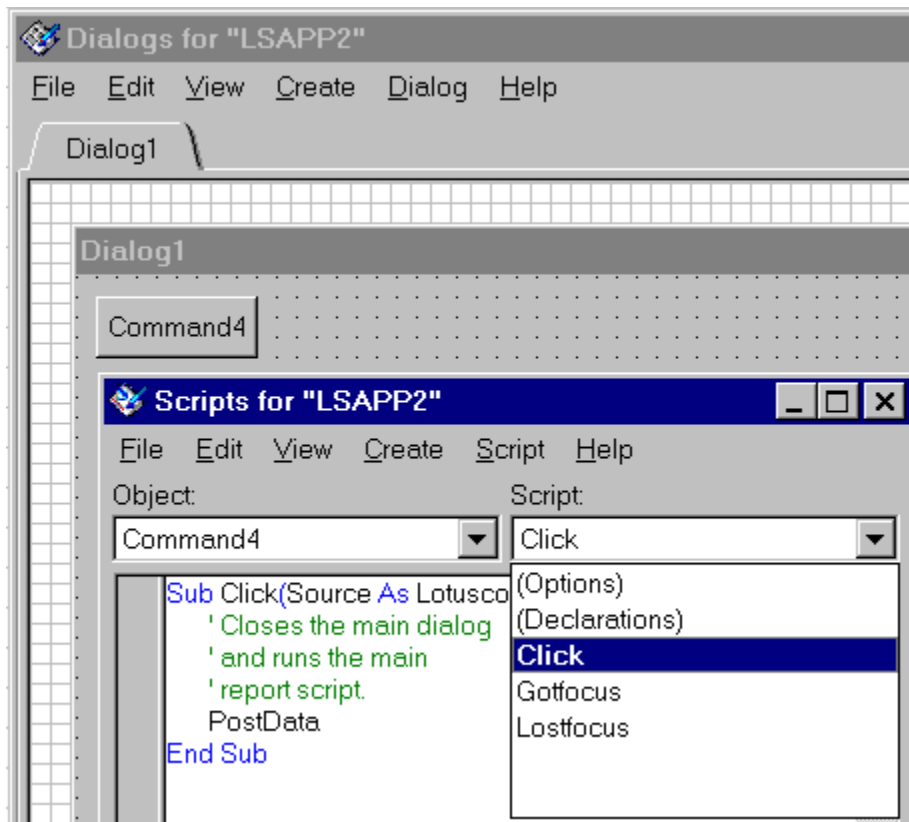
OLE Custom Control (OCX) files

OLE Custom Controls extend the number of objects that you can script in Lotus products. For example, the Lotus dialog controls listed under product classes in the IDE Browser panel are OCX controls that you can add to the Lotus Dialog Editor.



Once you add an OCX control to your product, you can script its properties, methods, and events in the IDE Script Editor.

The following illustration shows how the properties, methods, and events of an Lotus CommandButton OCX named Command4 are available to you in the IDE.



Tip You can add OCX controls registered on your system to the Lotus Dialog Editor Toolbox by choosing File - Toolbox Setup in the Lotus Dialog Editor menu.

Product Document

To edit scripts in the IDE or to execute them in one or more products, you must create or use a document in your product that contains the scripts. Lotus products supporting LotusScript use the following document extensions:

<i>Lotus Product</i>	<i>Document extension(s)</i>
1-2-3	123
Approach	APR
Freelance Graphics	PRZ
	SMC
Notes	NSF
Word Pro	LWP

Using LotusScript Examples

Code examples provide working models for the scripts that you write. Whether the example is listed in a Help example or available as a product document on disk, you can copy statements or entire scripts from the examples and use them in your own script applications.

There are three types of LotusScript examples, each designed to illustrate a different aspect of the LotusScript language or the classes available for each SmartSuite product.

Examples in reference Help

Most examples appear in reference Help for the LotusScript language and for product classes. These brief examples focus on individual elements in the language or members of a product class. They illustrate how to use correct syntax for a working example, how to enter appropriate values for parameters, and how dependencies between elements operate.

Note Although you can copy examples from reference Help and paste them into your scripts, they are not designed primarily to be self-contained. Sometimes there are dependencies between a piece of example code and the larger sample application from which it is derived.

Examples in Frequently Asked Questions (FAQs) Help

Frequently Asked questions (FAQs) illustrate how to complete common programming tasks using LotusScript. Examples in FAQs not only illustrate how individual statements work, but they also illustrate how these statements form a complete application or procedure. Most examples in FAQs are designed to be self-sufficient; you can copy one or more procedures from Help, paste them into your own scripts in the Script Editor, and execute them.

Note When there are dependencies in an example that would require you to modify the example to make it run, these dependencies are documented in the Help topic or at the beginning of the first script in the example.

Sample applications

The *Developing SmartSuite Applications Using LotusScript* book includes numerous sample applications for SmartSuite and for individual products. These examples are designed to illustrate more sophisticated tasks for an individual product or tasks that utilize more than one product. They illustrate how to develop script applications that take advantage of embedded OLE objects, OLE automation, Notes, Visual Basic, the Worldwide Web, and custom Dynamic-link libraries (DLLs). Lotus develops new sample applications for SmartSuite on an ongoing basis; these new samples and updated versions of the ones in *Developing SmartSuite Applications Using LotusScript* are available on the [World Wide Web](#).

To copy scripts from these sample applications and paste them into your own script applications, you must first open the sample application document and then display its scripts by opening the IDE window for that document.

Note All sample applications in *Developing SmartSuite Applications Using LotusScript* are designed to run without modification.

Using LotusScript Help

The design for LotusScript Help supports three of the most frequent activities that you perform as a programmer:

- Searching for objects and elements to use in your scripts
- Writing scripts
- Debugging scripts

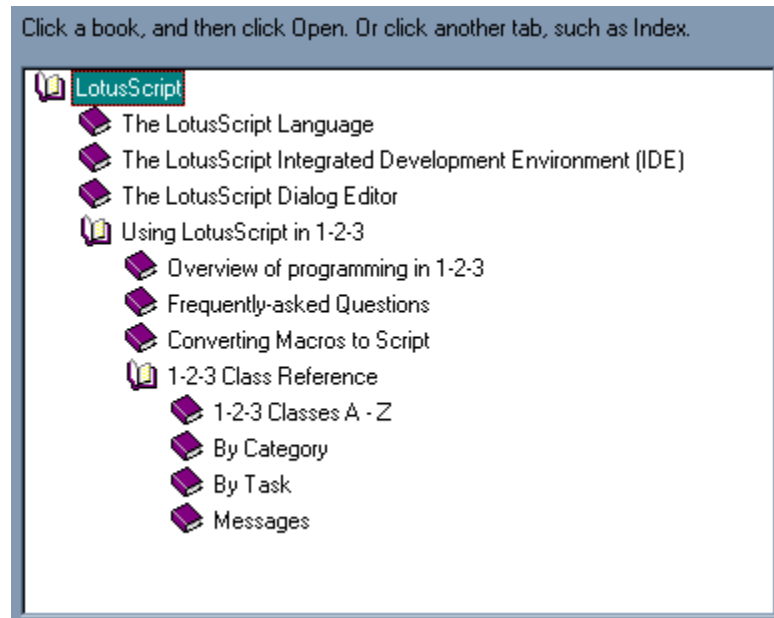
LotusScript Help uses different types of windows to display different types of information, so it is important to know what each type of window contains and how to navigate between them.

Using Help to search for objects and elements

There are areas in Help designed to help you search for objects and language elements to use in your scripts:

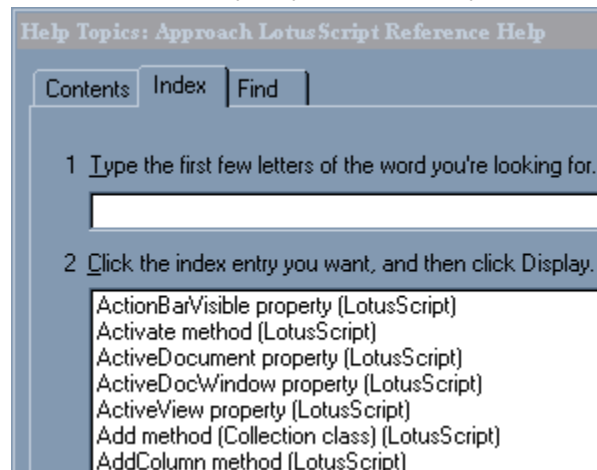
LotusScript Help Contents

You can use Contents in Help to examine the overall structure of Help and to browse for Help topics relevant to your current script.



LotusScript Index

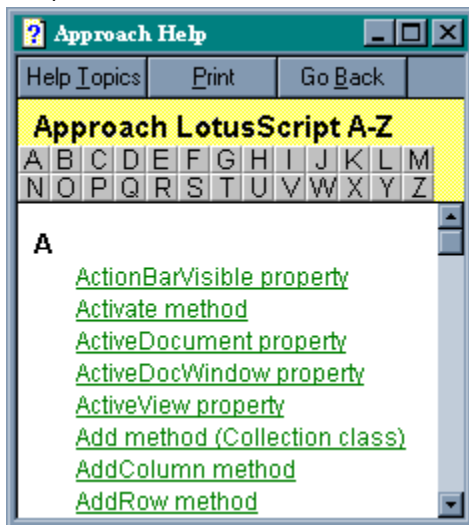
Indexes are one of the most popular ways that programmers search for information. Topics in LotusScript Help are indexed alphabetically, so you can enter key phrases or keywords and navigate to the corresponding Help topics.



LotusScript A - Z lists

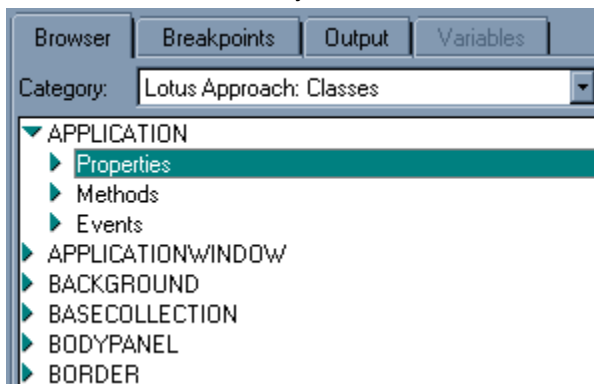
LotusScript Help for each product provides A - Z lists of its classes, properties, methods, and events, including a

comprehensive list of all the elements in the product.



IDE Browser Help

The Browser panel in the Integrated Development Environment (IDE) displays lists of LotusScript language elements and classes for products. You can expand and collapse entries in the Browser to view the associated properties, methods, and events for objects.



Highlight an element in the Browser panel and press F1 (HELP) to get context-sensitive Help on that element.

Using Help to write scripts

Help focuses on objects. As you are writing scripts, you explore the relationships between product classes and the behaviors of objects in that product.

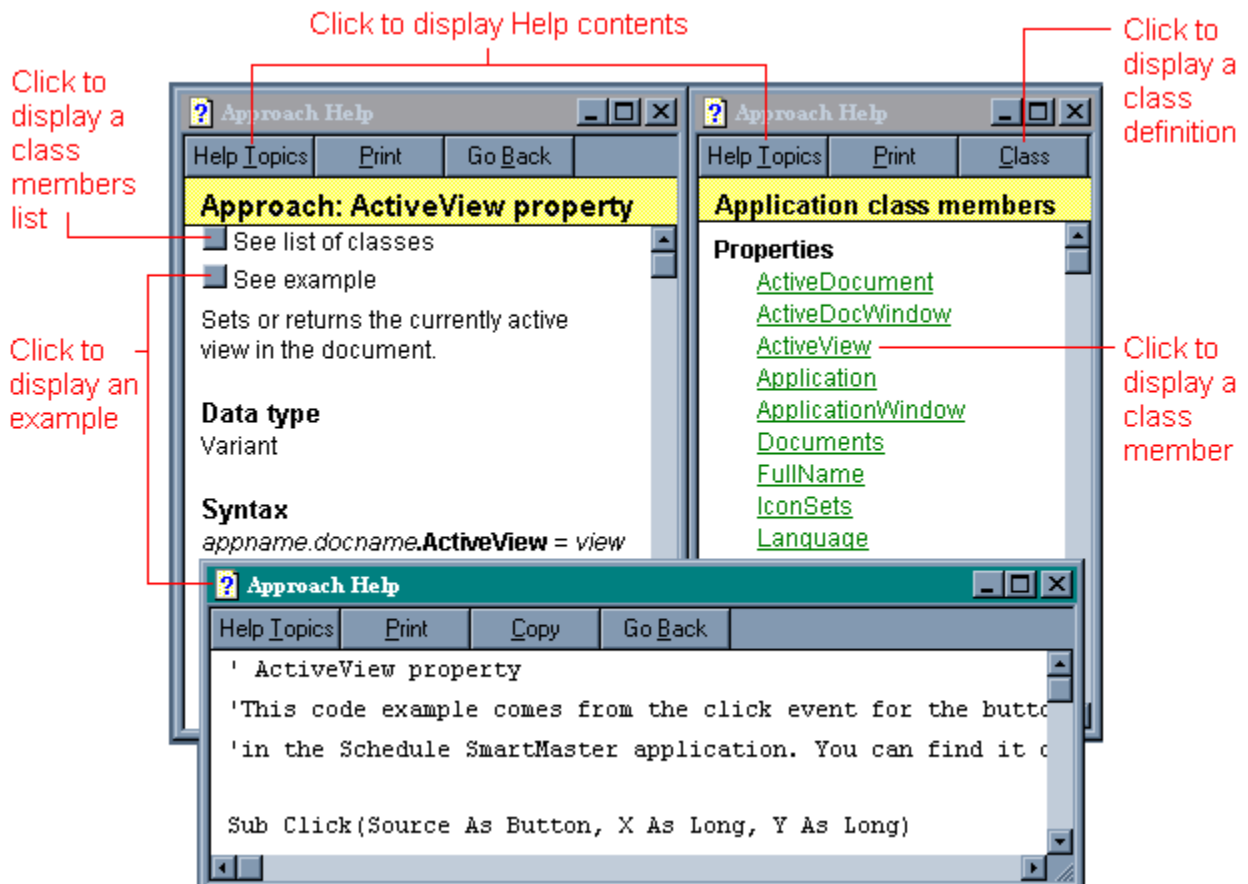
Types of Help windows

To support this exploration, Help separates information about classes into four types of windows:

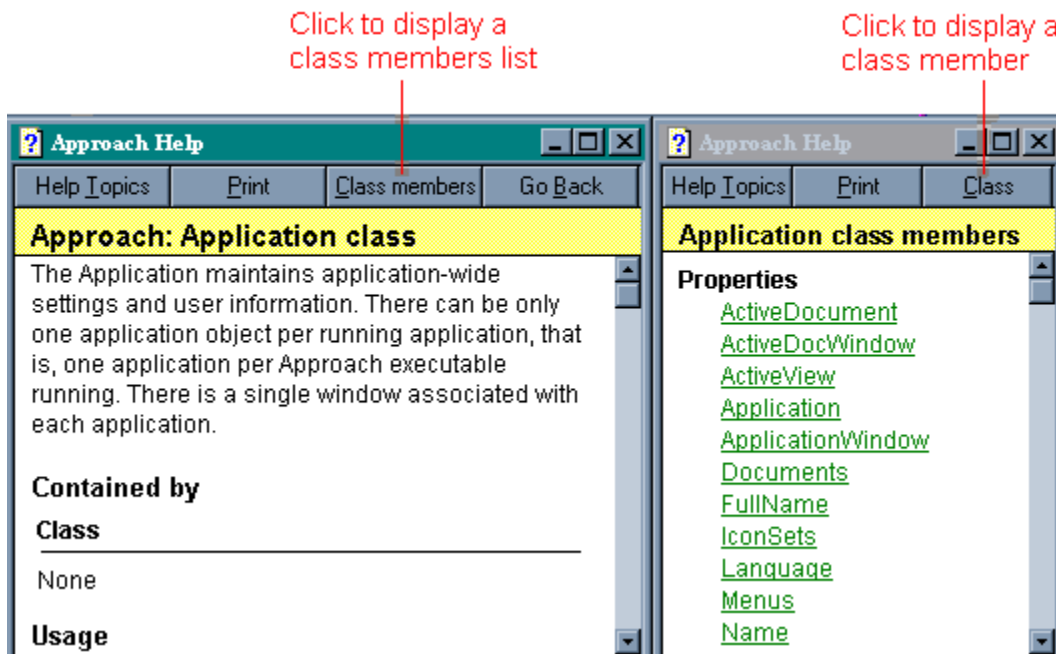
- Class definition windows define what a class does in a product and how it works in the product's containment hierarchy. The class definition topic for the 1-2-3 Range object describes what ranges do in 1-2-3, how they are contained by larger objects, and how they contain smaller objects.
- Class member list windows list all the properties, methods, and events that are members of a particular class.
- Class member windows focus on particular properties, methods, or events.
- Example windows contain one or more scripts for a particular property, method, or event. You can copy and paste script statements from these example windows into the IDE Script Editor.

Displaying Help windows

To display different types of LotusScript Help windows, use buttons in Help topics and in the Help window that are labeled by the type of Help window. The following illustration shows how to use buttons to display class member, class member list, and example windows in Help.



The following illustration shows how to display class definition and class member list windows in Help.



Help for editing and debugging scripts

You can also get context-sensitive Help about keywords and messages when you are editing or debugging your

scripts in the IDE.

Context-sensitive Help in the Script Editor and Script Debugger

If you need help on a keyword while you are writing or debugging a script in the Script Editor and Script Debugger, place the insertion point on the keyword and press F1 (HELP) to get context-sensitive Help on that keyword.

Context-sensitive Help on messages

You can also get context-sensitive Help on two types of messages in the IDE. In the Script Editor, you can get context-sensitive Help on syntax errors. Navigate to the statement that caused the error and press F1 (HELP). When you are debugging your scripts and the IDE reports a run-time error, press F1 (HELP) to display information about that error and suggestions about fixing it.

'Example: IsContinuedFrom property

'This example asks the user whether to print a message above a footnote

'which continues from the previous page, and then sets the appropriate option.

```
stat = MessageBox ("Do you want to print a continued from message?", 36, "Example  
Script")
```

```
If stat = 6 Then ' user said yes
```

```
    .Division.FootnoteOptions.IsContinuedFrom = True
```

```
    .Division.FootnoteOptions.ContinuedFromAlignment = $LtsAlignmentHorizCenter
```

```
    .Division.FootnoteOptions.ContinuedFromMessage = "Continued from previous page..."
```

```
Else
```

```
    .Division.FootnoteOptions.IsContinuedFrom = False
```

```
End If
```



```
'Example: IsContinuedOn property
'This example asks the user whether to print a message below a footnote
'which will continue to the next page, and then sets the appropriate option.

stat = MessageBox ("Do you want to print a continued on message?", 36, "Example
Script")
If stat = 6 Then ' user said yes
    .Division.FootnoteOptions.IsContinuedOn = True
    .Division.FootnoteOptions.ContinuedOnAlignment = $LtsAlignmentRight
    .Division.FootnoteOptions.ContinuedOnMessage = "Continued on next page..."
Else
    .Division.FootnoteOptions.IsContinuedOn = False
End If
```

'Example: IsDisableWarningMessages property
'This example asks the user whether to disable warning messages,
'and then sets the appropriate option.

```
stat = MessageBox ("Do you want to disable warning messages?", 36, "Example Script")
If stat = 6 Then ' user said yes
    .ApplicationWindow.UserInterfacePrefs.IsDisableWarningMessages = True
Else
    .ApplicationWindow.UserInterfacePrefs.IsDisableWarningMessages = False
End If
```

```
'Example: IsDisplayMisspelled property
'This example displays misspelled words, and types a misspelled word as an example.
.ActiveDocWindow.WinViewPrefs.IsDisplayMisspelled = True
.Type "Misspelled Word"
MessageBox "Click OK to hide Misspelled Words", MB_OK, "Example Script"
.ActiveDocWindow.WinViewPrefs.IsDisplayMisspelled = False
```

'Example: IsDivisionExternal property

'This example examines the current document, looking for external divisions.

'A message box is displayed indicating the status of each division.

ForAll Div in .ActiveDocument.Divisions

If Div.**IsDivisionExternal** then

 MessageBox Div.Divisioninfo.Name & " is an external division.", MB_OK, "Example
Script"

Else

 MessageBox Div.Divisioninfo.Name & " is an internal division.", MB_OK, "Example
Script"

End If

End Forall

```
'Example: IsDraw property
'This example checks the contents of a graphic frame, and displays a
'message box indicating the contents.

If .Graphic Is Nothing Then ' make sure a graphic is selected
    Messagebox "No graphic is selected.", MB_OK, "Example Script"
    Exit Sub
End If
If .Graphic.IsWordProChart Then
    Messagebox "A Word Pro 16 bit chart is selected", MB_OK, "Example script"
Elseif .Graphic.IsLotusChart Then
    Messagebox "A Lotus chart is selected", MB_OK, "Example script"

Elseif .Graphic.IsDraw Then
    Messagebox "A Word Pro drawing is selected", MB_OK, "Example script"

Elseif .Graphic.IsEquation Then
    Messagebox "A Word Pro equation is selected", MB_OK, "Example script"
Else
    Messagebox "A different type of graphic is selected", MB_OK, "Example script"
End If
```

```
'Example: IsEquation property
'This example checks the contents of a graphic frame, and displays a
'message box indicating the contents.

If .graphic Is Nothing Then ' make sure a graphic is selected
    Messagebox "No graphic is selected.", MB_OK, "Example Script"
    Exit Sub
End If
If .Graphic.IsWordProChart Then
    Messagebox "A Word Pro 16 bit chart is selected", MB_OK, "Example script"
Elseif .Graphic.IsLotusChart Then
    Messagebox "A Lotus chart is selected", MB_OK, "Example script"
Elseif .Graphic.IsDraw Then
    Messagebox "A Word Pro drawing is selected", MB_OK, "Example script"

Elseif .Graphic.IsEquation Then
    Messagebox "A Word Pro equation is selected", MB_OK, "Example script"

Else
    Messagebox "A different type of graphic is selected", MB_OK, "Example script"
End If
```

'Example: IsExportedAsNotesFX method

' This example determines whether or not the DocInfo fields will be exported
' to Lotus Notes fields.
' RUNTIME DEPENDENCIES: You must have a document open for this script to work.

```
Print .ActiveDocument.DocInfo.IsExportedAsNotesFX($LwpDocVarCreatedby )  
Print .ActiveDocument.DocInfo.IsExportedAsNotesFX($LwpDocVarDescription )  
Print .ActiveDocument.DocInfo.IsExportedAsNotesFX($LwpDocVarDocsize )
```

' This example inserts the 'created by' docinfo field. The field is made
' Lotus Notes FX aware.

```
.ActiveDocument.DocInfo.IsExportedAsNotesFX $LwpDocVarCreatedby  
.InsertDocInfo $LwpDocVarCreatedby
```

Note OLE is not supported under OS/2.

```
'Example: IsFilterTypePresent method
' This example prints all the text and table filter types to the Lotus Script
' Output panel.
' RUNTIME DEPENDENCIES: You must have a document open for this script to work.

Forall TextAndTable In .ApplicationWindow.Filter.TextAndTableImports
    FilterType = TextAndTable
    Print FilterType & " " &
        .ApplicationWindow.Filter.IsFilterTypePresent(FilterType,1)
End Forall
```


'Example: IsFixedLength property

'This example sets options for footnote separator lines, including

'setting a variable length line of 6 1/2 inches (9360 twips).

.Division.FootnoteOptions.FootnoteSeparator.UseSeparatorLine = True

.Division.FootnoteOptions.FootnoteSeparator.**IsFixedLength** = False

.Division.FootnoteOptions.FootnoteSeparator.CustomLength = 9360

.Division.FootnoteOptions.FootnoteSeparator.IndentFromLeft = 1440

.Division.FootnoteOptions.FootnoteSeparator.SpaceAbove = 40

.Division.FootnoteOptions.FootnoteSeparator.SpaceBelow = 60

.Division.FootnoteOptions.FootnoteContSep.UseSeparatorLine = True

```
'Example: IsFooter property
'This displays a message box indicating whether the insertion point is in the
'footer while the script is being run.

If .Page.IsFooter = 1 then
    MsgBox "The insertion point is in the footer.", MB_OK, "Example Script"
Else
    MsgBox "The insertion point is not in the footer.", MB_OK, "Example Script"
End If
```

'Example: IsHeader property

'This displays a message box indicating whether the insertion point is in the
'header while the script is being run.

If .Page.**IsHeader** = 1 then

 MessageBox "The insertion point is in the header.", MB_OK, "Example Script"

Else

 MessageBox "The insertion point is not in the header.", MB_OK, "Example Script"

End If

'Example: IsHiddenMark property

'This example sets revision marking options for the current editor to hidden with deleted marker, then demonstrates the feature.

```
.ActiveDocument.EditorManager.CurrentEditor.TextAttributes.IsHiddenMark = True  
.ActiveDocument.EditorManager.CurrentEditor.TextAttributes.HiddenMode = False  
.ActiveDocument.DivisionOptions.ShowHiddenText = False  
.Text.InsertText "This is some deleted text. "  
.ActiveDocument.RevisionMarkMode = 1  
.Type "[Left][Left][Left][Left][Left][Left][Left][shiftLeft][shiftLeft]"  
.Type "[shiftLeft][shiftLeft][shiftLeft][shiftLeft][shiftLeft][Del][Down]"  
.ActiveDocument.RevisionMarkMode = 0
```

'Example: IsHideHeaderFooter property

'This example shows, then hides header/footer areas.

```
.ActiveDocWindow.WinViewPrefs.IsHideHeaderFooter = True
```

```
MessageBox "Click OK to show headers and footers", MB_OK, "Example Script"
```

```
.ActiveDocWindow.WinViewPrefs.IsHideHeaderFooter = False
```

```
'Example: IsHorizontalScrollBarCleanScrn property
'This example asks the user whether to display the horizontal scroll bar while
'in Clean Screen view, and then sets the appropriate option.

stat = MessageBox ("Do you want to view the horizontal scroll bar in Clean Screen?",
36, "Example Script")
If stat = 6 Then ' user said yes
    .ActiveDocWindow.WinViewPrefs.IsHorizontalScrollBarCleanScrn = True
Else
    .ActiveDocWindow.WinViewPrefs.IsHorizontalScrollBarCleanScrn = False
End If
```

'Example: IsInDraft property

'This example displays the current document in draft view, then in the previous view.

.ActiveDocWindow.WinViewPrefs.**IsInDraft** = True

Msgbox "Click OK to turn off draft view." , MB_OK, "Example Script"

.ActiveDocWindow.WinViewPrefs.**IsInDraft** = False

'Example: IsInOutline property

'This example displays the current document in outline view.

.ActiveDocWindow.WinViewPrefs.**IsInOutline** = False

MessageBox "Click OK to display outline view", MB_OK, "Example Script"

.ActiveDocWindow.WinViewPrefs.**IsInOutline** = True

'Example: IsInPageSort property

'This example displays the current document in page sorter view.

```
.ActiveDocWindow.WinViewPrefs.IsInPageSort = False
```

```
MessageBox "Click OK to display page sorter view", MB_OK, "Example Script"
```

```
.ActiveDocWindow.WinViewPrefs.IsInPageSort = True
```

```
'Example: IsLotusChart property
'This example checks the contents of a graphic frame, and displays a
'message box indicating the contents.

If .graphic Is Nothing Then ' make sure a graphic is selected
    Messagebox "No graphic is selected.", MB_OK, "Example Script"
    Exit Sub
End If
If .Graphic.IsWordProChart Then
    Messagebox "A Word Pro 16 bit chart is selected", MB_OK, "Example script"

Elseif .Graphic.IsLotusChart Then
    Messagebox "A Lotus chart is selected", MB_OK, "Example script"

Elseif .Graphic.IsDraw Then
    Messagebox "A Word Pro drawing is selected", MB_OK, "Example script"
Elseif .Graphic.IsEquation Then
    Messagebox "A Word Pro equation is selected", MB_OK, "Example script"
Else
    Messagebox "A different type of graphic is selected", MB_OK, "Example script"
End If
```

'Example: IsMarginSameAsParent property

'This example creates a new document, and then changes the header margins.

.NewDocument

.Layout.Header.**IsMarginSameAsParent** = False ' this lets us change the margins.

.Layout.Header.MarginLeft = 2880 ' 2 inches

.Layout.Header.MarginRight = 720 ' 1/2 inch

```
'Example: IsMarginsInColor property
'This example displays margins in color.
.ActiveDocWindow.WinViewPrefs.IsMarginsInColor = False
MessageBox "Click OK to display margins in color", MB_OK, "Example Script"
.ActiveDocWindow.WinViewPrefs.IsMarginsInColor = True
```

'Example: IsOverridden property

'This example types a paragraph of text, and tests the value of the
'IsOverridden property. It then bolds the paragraph, and then retests
'the value.

.Text.InsertText "Now is the time for all good men to come to the aid of their party."

If .Text.**IsOverridden** = False Then

 MessageBox "No local attributes have been set."

End If

.Text.Select \$LwpSelectObjectTypeParagraph

.Text.Font.Bold = True

If .Text.**IsOverridden** = True Then

 MessageBox "Local attributes have been set."

End If

```
'Example: IsPageBreakAfter property
'This example sets options for page breaks before and after paragraphs.
.Type "This is some text in the on the first page.[Enter]"
.Text.Breaks.IsPageBreakBefore = True
.Type "There is a page break before this paragraph[Enter]"
.Text.Breaks.IsPageBreakBefore = False
.Text.Breaks.IsPageBreakAfter = True ' force a page break after this paragraph.
.Type "There is a page break after this paragraph[Enter]"
.Text.Breaks.IsPageBreakAfter = False
.Type "This is text on the third page."
```

```
'Example: IsPageBreakBefore property
'This example sets options for page breaks before and after paragraphs.
.Type "This is some text in the on the first page.[Enter]"
.Text.Breaks.IsPageBreakBefore = True ' force a break before this paragraph.
.Type "There is a page break before this paragraph[Enter]"
.Text.Breaks.IsPageBreakBefore = False
.Text.Breaks.IsPageBreakAfter = True
.Type "There is a page break after this paragraph[Enter]"
.Text.Breaks.IsPageBreakAfter = False
.Type "This is text on the third page."
```

'Example: IsPageBreakMarks property
'This example displays page break marks.

```
.ActiveDocWindow.WinViewPrefs.IsPageBreakMarks = False  
.Text.InsertText "There will be a page break below this line."  
.Text.InsertBreak $LwpBreakTypePage ' put in a break so we can see it.  
MessageBox "Click OK to display page break marks", MB_OK, "Example Script"  
.ActiveDocWindow.WinViewPrefs.IsPageBreakMarks = True
```


'Example: IsPageBreakWithin property

'This example asks the user whether to allow a page break in the current
'paragraph, and then sets the appropriate option.

```
stat = MessageBox ("Do you want to allow a page break in this paragraph?", 36,  
"Example Script")
```

```
If stat = 6 Then ' user said yes
```

```
    .Text.Breaks.IsPageBreakWithin = True
```

```
Else
```

```
    .Text.Breaks.IsPageBreakWithin = False
```

```
End If
```

```
'Example: IsPageBreak property
'This example creates a table, asks the user whether to force a
'page break after the current row, and then sets the appropriate option.

.CreateTable False, "Default Table", 3, 2
stat = MessageBox ("Do you want a page break after this row?", 36, "Example Script")
If stat = 6 Then ' user said yes
    .Table.CurrentRow.IsPageBreak = True
Else
    .Table.CurrentRow.IsPageBreak = False
End If
```

```
'Example: IsParagraphNumberingDown property
'This example asks the user whether to reset numbering at the top of each column
'in a newly created table, and then sets the appropriate option.
'The style is then changed to a numbered list, and the text is copied to
'several cells to illustrate the example

.CreateTable False, "Default Table", 3, 2
stat = MessageBox ("Do you numbering to reset on each column?", 36, "Example Script")
If stat = 6 Then ' user said yes
    .Table.IsParagraphNumberingDown = True
Else
    .Table.IsParagraphNumberingDown = False
End If
.Text.ParagraphStyleName = "Number List"
.Type "A Numbered list.[shiftHome]"
.CopySelection
.Type "[Down]"
.Paste
.Type "[Tab][Up]"
.Paste
.Type "[Down]"
.Paste
```

```
'Example: IsPrePrintedForm property
'This example asks the user whether to print form prompt text, sets the
'appropriate option, and prints the document.

stat = MessageBox ("Is this document a preprinted form?", 36, "Example Script")
If stat = 6 Then ' user said yes
    .ActiveDocument.PrintSettings.IsPrePrintedForm = True
Else
    .ActiveDocument.PrintSettings.IsPrePrintedForm = False
End If
.Print
```

```
'Example: IsPrintClickHereBlocks property
'This example asks the user whether to print unfilled Click Here blocks,
'sets the appropriate option, and prints the document.

stat = MessageBox ("Do you want to print empty Click Here blocks?", 36, "Example
Script")
If stat = 6 Then ' user said yes
    .ActiveDocument.PrintSettings.IsPrintClickHereBlocks = True
Else
    .ActiveDocument.PrintSettings.IsPrintClickHereBlocks = False
End If
.Print
```

'Example: IsProtected property
'This example creates a table, then protects the current cell.
'Then the option to honor protection of cells is enabled for the division.

```
.CreateTable False, "Default Table", 2, 2  
.Type "This cell will be protected."  
.Table.CurrentCell.IsProtected = True ' enable cell protection  
.Division.IsProtected = True ' enable division protection
```

```
'Example: IsRepeat property
'This example asks the user whether to repeate footnote numbers for
'footnotes which continue to the next page, and then sets the appropriate option.

stat = MessageBox ("Do you want to repeat footnote numbers?", 36, "Example Script")
If stat = 6 Then ' user said yes
    .Division.FootnoteOptions.IsRepeat = True
Else
    .Division.FootnoteOptions.IsRepeat = False
End If
```

```
'Example: ShowExportWarningMessages property
'This example asks the user whether to display warning messages when saving
'a document to a type other than Word Pro, and then sets the appropriate option.

stat = MessageBox ("Do you want to display an alert when saving files to a different
type?", 36, "Example Script")
If stat = 6 Then ' user said yes
    .ApplicationWindow.UserInterfacePrefs.ShowExportWarningMessages = True
Else
    .ApplicationWindow.UserInterfacePrefs.ShowExportWarningMessages = False
End If
```



```
'Example: IsShowing property
' This example hides all visible SmartIcons bars if any are showing
IsItShowing = .ApplicationWindow.IconBarManager.IsShowing
If IsItShowing = True Then
    .ToggleIconBar
End If
```

'Example: IsSizable property
'This example creates a table, then disables auto row height. Finally,
'the row height is locked at its current height.

```
.CreateTable False, "Default Table", 3, 2  
.Table.CurrentRow.DirectionDown = &H0 ' disable auto row height  
.Table.CurrentRow.Height = 288  
.Table.CurrentRow.IsSizable = False ' lock sizing with the mouse
```

'Example: IsSmartCorrectEnabled property

'This example asks the user whether to disable SmartCorrect,

'and then sets the appropriate option.

```
stat = MessageBox ("Do you want to disable SmartCorrect?", 36, "Example Script")
```

```
If stat = 6 Then ' user said yes
```

```
    .Application.Preferences.IsSmartCorrectEnabled = False
```

```
Else
```

```
    .Application.Preferences.IsSmartCorrectEnabled = True
```

```
End If
```

'Example: IsSmartShadeEnabled property
'This example asks the user whether to disable SmartSelect,
'and then sets the appropriate option.

```
stat = MessageBox ("Do you want to disable SmartSelect?", 36, "Example Script")  
If stat = 6 Then ' user said yes  
    .Application.Preferences.IsSmartShadeEnabled = False  
Else  
    .Application.Preferences.IsSmartShadeEnabled = True  
End If
```

'Example: IsSnapTo property

'This example creates a frame, then displays a grid with lines spaced one
'inch apart. The contents of the frame is then set to snap to the grid.

```
.CreateFrame
```

```
.Frame.Layout.GridType = $LwpGridTypeLines
```

```
.Frame.Layout.GridDistance = 1440
```

```
.Frame.Layout.IsSnapTo = True
```

```
'Example: IsTextLocked property
'This example protects some text, then enables protection for the text.
.Text.Attributes.ProtectedMode = True ' turn on protect attribute
.Division.DivisionOptions.IsTextLocked = False ' disable protection for text
.Type "Now is the time for all good men to come to the aid of their party."
.Text.Attributes.ProtectedMode = False
.Division.DivisionOptions.IsTextLocked = True ' now honor protection
```

```
'Example: IsUndoOn property
''This example turns the Undo recorder off in the currently
active division and then turns it on again.
'Paste this example in the Sub Main section of your script.
.Foundry.IsUndoOn = False
.Foundry.IsUndoOn = True
```

```
'Example: IsUpdateIndex property
'This example asks the user whether to update the Index before printing,
'sets the appropriate option, and prints the document.

stat = MessageBox ("Do you want to update the index before printing?", 36, "Example
Script")
If stat = 6 Then ' user said yes
    .ActiveDocument.PrintSettings.IsUpdateIndex = True
Else
    .ActiveDocument.PrintSettings.IsUpdateIndex = False
End If
.Print
```


'Example: IsUpdateTOC property
'This example asks the user whether to update the Table of Contents before printing,
'sets the appropriate option, and prints the document.

```
stat = MessageBox ("Do you want to update the Table of Contents?", 36, "Example  
Script")
```

```
If stat = 6 Then ' user said yes
```

```
    .ActiveDocument.PrintSettings.IsUpdateTOC = True
```

```
Else
```

```
    .ActiveDocument.PrintSettings.IsUpdateTOC = False
```

```
End If
```

```
.Print
```

```
'Example: IsVerticalScrollBarCleanScrn property
'This example asks the user whether to display the vertical scroll bar while
'in Clean Screen view, and then sets the appropriate option.

stat = MessageBox ("Do you want to view the vertical scroll bar in Clean Screen?", 36,
"Example Script")
If stat = 6 Then ' user said yes
    .ActiveDocWindow.WinViewPrefs.IsVerticalScrollBarCleanScrn = True
Else
    .ActiveDocWindow.WinViewPrefs.IsVerticalScrollBarCleanScrn = False
End If
```

'Example: IsViewAnchor property

'This example asks the user whether to display the location of frame and table
'anchors in the current document, and then sets the appropriate option.

```
stat = MessageBox ("Do you want to view anchor marks?", 36, "Example Script")
```

```
If stat = 6 Then ' user said yes
```

```
    .ActiveDocWindow.WinViewPrefs.IsViewAnchor = True
```

```
Else
```

```
    .ActiveDocWindow.WinViewPrefs.IsViewAnchor = False
```

```
End If
```

'Example: IsViewBookmarks property
'This example asks the user whether to display bookmarks in the current document,
'and then sets the appropriate option.

```
stat = MessageBox ("Do you want to view bookmarks?", 36, "Example Script")
If stat = 6 Then ' user said yes
    .ActiveDocWindow.WinViewPrefs.IsViewBookmarks = True
Else
    .ActiveDocWindow.WinViewPrefs.IsViewBookmarks = False
End If
```

```
'Example: IsViewClickHereBlocks property
'This example asks the user whether to display Click Here blocks in the current
document,
'and then sets the appropriate option.

stat = MessageBox ("Do you want to view Click Here blocks?", 36, "Example Script")
If stat = 6 Then  ' user said yes
    .ActiveDocWindow.WinViewPrefs.IsViewClickHereBlocks = True
Else
    .ActiveDocWindow.WinViewPrefs.IsViewClickHereBlocks = False
End If
```

'Example: IsViewColGuides property
'This example asks the user whether to display column guides in the current document,
'and then sets the appropriate option.

```
stat = MessageBox ("Do you want to view column guides?", 36, "Example Script")
If stat = 6 Then ' user said yes
    .ActiveDocWindow.WinViewPrefs.IsViewColGuides = True
Else
    .ActiveDocWindow.WinViewPrefs.IsViewColGuides = False
End If
```

```
'Example: IsViewColumnBreakMarks property
'This example asks the user whether to display column break marks in the current
document,
'and then sets the appropriate option.

stat = MessageBox ("Do you want to view column break marks?", 36, "Example Script")
If stat = 6 Then ' user said yes
    .ActiveDocWindow.WinViewPrefs.IsViewColumnBreakMarks = True
Else
    .ActiveDocWindow.WinViewPrefs.IsViewColumnBreakMarks = False
End If
```

```
'Example: IsViewDDELinks property
'This example asks the user whether to display marks indicating DDE links
'in the current document, and then sets the appropriate option.

stat = MessageBox ("Do you want to view DDE link marks?", 36, "Example Script")
If stat = 6 Then ' user said yes
    .ActiveDocWindow.WinViewPrefs.IsViewDDELinks = True
Else
    .ActiveDocWindow.WinViewPrefs.IsViewDDELinks = False
End If
```



```
'Example: IsViewHorzRuler property
'This example asks the user whether to display the horizontal tab ruler
'in the current document, and then sets the appropriate option.

stat = MessageBox ("Do you want to view the tab ruler?", 36, "Example Script")
If stat = 6 Then ' user said yes
    .ActiveDocWindow.WinViewPrefs.IsViewHorzRuler = True
Else
    .ActiveDocWindow.WinViewPrefs.IsViewHorzRuler = False
End If
```

```
'Example: IsViewMenuCleanScrn property
'This example asks the user whether to display the menu bar while
'in Clean Screen view, and then sets the appropriate option.

stat = MessageBox ("Do you want to view the menu bar in Clean Screen?", 36, "Example
Script")
If stat = 6 Then ' user said yes
    .ActiveDocWindow.WinViewPrefs.IsViewMenuCleanScrn = True
Else
    .ActiveDocWindow.WinViewPrefs.IsViewMenuCleanScrn = False
End If
```

```
'Example: IsViewNotes property
'This example asks the user whether to display marks at the location of
'comment notes in the current document, and then sets the appropriate option.

stat = MessageBox ("Do you want to view comment note marks?", 36, "Example Script")
If stat = 6 Then ' user said yes
    .ActiveDocWindow.WinViewPrefs.IsViewNotes = True
Else
    .ActiveDocWindow.WinViewPrefs.IsViewNotes = False
End If
```

```
'Example: IsViewOutlineFlowToScreen property
'This example asks the user whether to wrap outline text within the document window,
'and then sets the appropriate option.

stat = MessageBox ("Do you want to wrap outline text within the window?", 36, "Example
Script")
If stat = 6 Then ' user said yes
    .ActiveDocWindow.WinViewPrefs.IsViewOutlineFlowToScreen = True
Else
    .ActiveDocWindow.WinViewPrefs.IsViewOutlineFlowToScreen = False
End If
```

```
'Example: IsViewOutlineIndent property
'This example asks the user whether to display indented outline levels in outline
view,
'and then sets the appropriate option.

stat = MessageBox ("Do you want to view indented outlines?", 36, "Example Script")
If stat = 6 Then ' user said yes
    .ActiveDocWindow.WinViewPrefs.IsViewOutlineIndent = True
Else
    .ActiveDocWindow.WinViewPrefs.IsViewOutlineIndent = False
End If
```

```
'Example: IsViewPageLayoutMarks property
'This example asks the user whether to display marks at the location of
'inserted page layouts in the current document, and then sets the appropriate option.

stat = MessageBox ("Do you want to view page layout marks?", 36, "Example Script")
If stat = 6 Then ' user said yes
    .ActiveDocWindow.WinViewPrefs.IsViewPageLayoutMarks = True
Else
    .ActiveDocWindow.WinViewPrefs.IsViewPageLayoutMarks = False
End If
```

```
'Example: IsViewParallelColumnBorder property
'This example asks the user whether to display parallel column guides in the current
document,
'and then sets the appropriate option.

stat = MessageBox ("Do you want to view parallel column guides?", 36, "Example
Script")
If stat = 6 Then ' user said yes
    .ActiveDocWindow.WinViewPrefs.IsViewParallelColumnBorder = True
Else
    .ActiveDocWindow.WinViewPrefs.IsViewParallelColumnBorder = False
End If
```

'Example: IsViewPictures property
'This example asks the user whether to display graphics in the current document,
'and then sets the appropriate option.

```
stat = MessageBox ("Do you want to view pictures?", 36, "Example Script")
If stat = 6 Then ' user said yes
    .ActiveDocWindow.WinViewPrefs.IsViewPictures = True
Else
    .ActiveDocWindow.WinViewPrefs.IsViewPictures = False
End If
```



```
'Example: IsViewReturnIconCleanScrn property
'This example asks the user whether to display the Return icon while in
'clean screen view, and then sets the appropriate option.

stat = MessageBox ("Do you want to display the Return icon?", 36, "Example Script")
If stat = 6 Then ' user said yes
    .ActiveDocWindow.WinViewPrefs.IsViewReturnIconCleanScrn = True
Else
    .ActiveDocWindow.WinViewPrefs.IsViewReturnIconCleanScrn = False
End If
```

'Example: IsViewReturns property
'This example asks the user whether to display return symbols in the current document,
'and then sets the appropriate option.

```
stat = MessageBox ("Do you want to view return symbols?", 36, "Example Script")
If stat = 6 Then ' user said yes
    .ActiveDocWindow.WinViewPrefs.IsViewReturns = True
Else
    .ActiveDocWindow.WinViewPrefs.IsViewReturns = False
End If
```

'Example: IsViewRulerMarks property
'This example asks the user whether to display ruler marks in the current document,
'and then sets the appropriate option.

```
stat = MessageBox ("Do you want to view ruler marks?", 36, "Example Script")
If stat = 6 Then ' user said yes
    .ActiveDocWindow.WinViewPrefs.IsViewRulerMarks = True
Else
    .ActiveDocWindow.WinViewPrefs.IsViewRulerMarks = False
End If
```

```
'Example: IsViewSectionBreakMarks property
'This example asks the user whether to display marks at section breaks
'in the current document, and then sets the appropriate option.

stat = MessageBox ("Do you want to view section marks?", 36, "Example Script")
If stat = 6 Then ' user said yes
    .ActiveDocWindow.WinViewPrefs.IsViewSectionBreakMarks = True
Else
    .ActiveDocWindow.WinViewPrefs.IsViewSectionBreakMarks = False
End If
```

'Example: IsViewSectionTabs property
'This example asks the user whether to display divider tabs in the current document,
'and then sets the appropriate option.

```
stat = MessageBox ("Do you want to view divider tabs?", 36, "Example Script")
If stat = 6 Then ' user said yes
    .ActiveDocWindow.WinViewPrefs.IsViewSectionTabs = True
Else
    .ActiveDocWindow.WinViewPrefs.IsViewSectionTabs = False
End If
```

```
'Example: IsViewSmartIconsCleanScrn property
'This example asks the user whether to display SmartIcons while
'in Clean Screen view, and then sets the appropriate option.

stat = MessageBox ("Do you want to view SmartIcons in Clean Screen?", 36, "Example
Script")
If stat = 6 Then ' user said yes
    .ActiveDocWindow.WinViewPrefs.IsViewSmartIconsCleanScrn = True
Else
    .ActiveDocWindow.WinViewPrefs.IsViewSmartIconsCleanScrn = False
End If
```

'Example: IsViewStatusBarCleanScrn property

'This example asks the user whether to display the Status bar while
'in Clean Screen view, and then sets the appropriate option.

```
stat = MessageBox ("Do you want to view the Status bar in Clean Screen?", 36, "Example  
Script")
```

```
If stat = 6 Then ' user said yes
```

```
    .ActiveDocWindow.WinViewPrefs.IsViewStatusBarCleanScrn = True
```

```
Else
```

```
    .ActiveDocWindow.WinViewPrefs.IsViewStatusBarCleanScrn = False
```

```
End If
```

'Example: IsViewTableHeadings property
'This example asks the user whether to display row and column headings for tables
'in the current document, and then sets the appropriate option.

```
stat = MessageBox ("Do you want to view table headings?", 36, "Example Script")
If stat = 6 Then ' user said yes
    .ActiveDocWindow.WinViewPrefs.IsViewTableHeadings = True
Else
    .ActiveDocWindow.WinViewPrefs.IsViewTableHeadings = False
End If
```


'Example: IsViewTabs property
'This example asks the user whether to display tab marks in the current document,
'and then sets the appropriate option.

```
stat = MessageBox ("Do you want to view tab marks?", 36, "Example Script")
If stat = 6 Then ' user said yes
    .ActiveDocWindow.WinViewPrefs.IsViewTabs = True
Else
    .ActiveDocWindow.WinViewPrefs.IsViewTabs = False
End If
```

'Example: IsViewThumbBarCleanScrn property
'This example asks the user whether to display the page gauge in the current document,
'and then sets the appropriate option.

```
stat = MessageBox ("Do you want to view the page gauge?", 36, "Example Script")
If stat = 6 Then ' user said yes
    .ActiveDocWindow.WinViewPrefs.IsViewThumbBarCleanScrn = True
Else
    .ActiveDocWindow.WinViewPrefs.IsViewThumbBarCleanScrn = False
End If
```

```
'Example: IsViewTitleBarCleanScrn property
'This example asks the user whether to display the Windows title bar while
'in Clean Screen view, and then sets the appropriate option.

stat = MessageBox ("Do you want to view the title bar in Clean Screen?", 36, "Example
Script")
If stat = 6 Then ' user said yes
    .ActiveDocWindow.WinViewPrefs.IsViewTitleBarCleanScrn = True
Else
    .ActiveDocWindow.WinViewPrefs.IsViewTitleBarCleanScrn = False
End If
```

```
'Example: IsViewVertRuler property
'This example asks the user whether to display the vertical ruler in the current
document,
'and then sets the appropriate option.

stat = MessageBox ("Do you want to view the vertical ruler?", 36, "Example Script")
If stat = 6 Then ' user said yes
    .ActiveDocWindow.WinViewPrefs.IsViewVertRuler = True
Else
    .ActiveDocWindow.WinViewPrefs.IsViewVertRuler = False
End If
```

```
'Example: IsWMCommandValid method
' This example uses the IsWmCommandvalid to ensure that a table is selected
' before querying for table information.
' RUNTIME DEPENDENCIES: You must have a document open for this script to work.

.CreateTable False, "Default Table", 5,5
' check to see if the table infobox could be displayed. If so, we're in a table.
If (.IsWmCommandvalid(LwpMenuMtTableinfobox)) Then
    Dim MyTable As Table
    Set MyTable = .Table
    Print "Table Name: " & MyTable.Name
    Print "Table Rows:" & Str$(MyTable.NumRows)
    Print "Table Cols:" & Str$(MyTable.NumCols)
End If
```

```
'Example: IsWordProChart property
'This example checks the contents of a graphic frame, and displays a
'message box indicating the contents.

If .graphic Is Nothing Then ' make sure a graphic is selected
    Messagebox "No graphic is selected.", MB_OK, "Example Script"
    Exit Sub
End If

If .Graphic.IsWordProChart Then
    Messagebox "A Word Pro 16 bit chart is selected", MB_OK, "Example script"

Elseif .Graphic.IsLotusChart Then
    Messagebox "A Lotus chart is selected", MB_OK, "Example script"
Elseif .Graphic.IsDraw Then
    Messagebox "A Word Pro drawing is selected", MB_OK, "Example script"
Elseif .Graphic.IsEquation Then
    Messagebox "A Word Pro equation is selected", MB_OK, "Example script"
Else
    Messagebox "A different type of graphic is selected", MB_OK, "Example script"
End If
```

'Example: IsWorkingDir property

'This example asks the user whether to use the last folder chosen when

'displaying a Browse dialog box, and then sets the appropriate option.

```
stat = MessageBox ("Do you want to use the working directory?", 36, "Example Script")
```

```
If stat = 6 Then ' user said yes
```

```
    .ApplicationWindow.UserInterfacePrefs.IsWorkingDir = True
```

```
Else
```

```
    .ApplicationWindow.UserInterfacePrefs.IsWorkingDir = False
```

```
End If
```

'Example: Italic method

' This example first inserts sample text in the current document and selects
' the paragraph. The script then uses the Italic method to toggle the Italic
' attribute.

' RUNTIME DEPENDENCIES: You must have a document open with selected text
' for this script to work.

.Text.InsertText "This is some sample text."
.SelectParagraph

.Italic

'Example: Italic property

'This example enables the italic attribute, types some text, then disables the attribute.

.Text.Font.**Italic** = True

.Text.InsertText "This is italicized text."

.Text.Font.**Italic** = False

```
'Example: Item method
' This example creates a bookmark from the current marker and then prints
' the bookmark's marker name by indexing into the Bookmarks collection using
' the Item method
' RUNTIME DEPENDENCIES: You must have a document open for this script to work.
```

```
Dim MarkerName As String
MarkerName = .Mark($LwpMarkerTypeBookmark)
.Division.BookmarkManager.AddBookmark "ExampleBookmark", MarkerName
Print .Division.BookmarkManager.Bookmarks.Item("ExampleBookmark").MarkerName
```

'Example: JoinName property

'This example creates an custom border for the current page layout.

'The border is then changed to another pattern.

```
.Page.Layout.RightPage.BorderLines.LinePlacement = &Hf
.Page.Layout.RightPage.BorderLines.AllBorders.Pattern = -32768
.Page.Layout.RightPage.BorderLines.AllBorders.NameOfExternalBorder =
"1998327121627.EDB"
.Page.Layout.RightPage.BorderLines.AllBorders.WidthInTwips = 720
.Page.Layout.RightPage.BorderLines.AllBorders.Color.Red = 176
.Page.Layout.RightPage.BorderLines.AllBorders.Color.Blue = 0
.Page.Layout.RightPage.BorderLines.AllBorders.Color.Green = 198
.Page.Layout.RightPage.BorderLines.AllBorders.Color.Override =
$LwpColorOverrideBlack
.Page.Layout.RightPage.Join.JoinType = -32768
.Page.Layout.RightPage.Join.Clear
.Page.Layout.RightPage.Join.JoinName = "1998326125038.EDB"
.Page.Layout.RightPage.Join.ScaleMode = $LwpJoinScaleMatchborder
.Page.Layout.RightPage.Join.Percentage = 30

MessageBox "Click OK to change the designer border.", MB_OK, "Example Script"
.Page.Layout.RightPage.BorderLines.AllBorders.NameOfExternalBorder = "\Borders\
emf\1713.emf"
.Page.Layout.RightPage.Join.Clear
.Page.Layout.RightPage.Join.JoinName = "\Borders\emf\17c013.emf"
```

'Example: Justifiable property

'This example creates a table, then prevents the first column from automatically
'changing width if the width of the table changes.

```
.CreateTable False, "Default Table", 3, 2
```

```
.Table.CurrentColumn.Justifiable = False
```

'Example: KeepWithNext property

'This example inserts two paragraphs of text into the current document.

'The first paragraph is set to be kept with the next paragraph, and

'the second paragraph is set to be kept with the first.

.Text.Breaks.**KeepWithNext** = True

.Text.InsertText "This paragraph of text will be kept with the next paragraph.", True

.Text.Breaks.KeepWithPrev = True

.Text.InsertText "This paragraph of text will be kept with the previous paragraph.",

True

'Example: KeepWithPrev property

'This example inserts two paragraphs of text into the current document.

'The first paragraph is set to be kept with the next paragraph, and

'the second paragraph is set to be kept with the first.

.Text.Breaks.KeepWithNext = True

.Text.InsertText "This paragraph of text will be kept with the next paragraph.", True

.Text.Breaks.**KeepWithPrev** = True

.Text.InsertText "This paragraph of text will be kept with the previous paragraph.",

True

'Example: Keywords property

'This example prints the keywords for the current document to the LotusScript output panel.

Print "Document keywords: " & .ActiveDocument.DocInfo.**Keywords**

'Example: LandscapeMode property

'This changes the current page layout from portrait to landscape mode.

NewHeight = .Page.Layout.RightPage.Width

NewWidth = .Page.Layout.RightPage.Height

.Page.Layout.RightPage.Width = NewWidth

.Page.Layout.RightPage.Height = NewHeight

.Page.Layout.RightPage.**LandscapeMode** = True

'Example: Language property

'This example sets language options for the division and text.

' set the division's language to Japanese

.Division.DivisionOptions.**Language.Language** = \$LwpLanguagesJapanese

' set the text's language to French Canadian

.Text.**Language.Language** = \$LwpLanguagesFrenchCanadian

'Example: LastEditorName property

'This example displays a message box with the name of the last editor
'of the text the insertion point is in. Place the insertion point in
'different text areas in the document to see who last edited them.

```
MessageBox "The last editor of this text was " & .Text.LastEditorName, MB_OK, "Example  
Script"
```

```
'Example: LastInsertSymbolFont property
'This example displays an input box asking for the name of the font to
'use for inserting symbols. The current font name is proposed as
'the default.

OldFont = .ApplicationWindow.UserInterfacePrefs.LastInsertSymbolFont
NewFont = Inputbox ("Which font do you want to use for inserting symbols?", "Example
Script", OldFont)
If NewFont <> "" Then ' user did not cancel
    .ApplicationWindow.UserInterfacePrefs.LastInsertSymbolFont = NewFont
End If
```

'Example: LastName property

'This example inserts a new division into the document, and then displays

'the names of the first and last divisions in a message box.

```
.ApplicationWindow.SectionTabs.AddNewSectionTabs
```

```
FirstDiv = .ActiveDocument.Divisions(.ActiveDocument.FirstName).DivisionInfo.Name
```

```
LastDiv = .ActiveDocument.Divisions(.ActiveDocument.LastName).DivisionInfo.Name
```

```
MessageBox "The first division in this document is " & FirstDiv & _
```

```
    " and the last division is " & LastDiv & ".", MB_OK, "Example Script"
```

'Example: LastPage property

'This example inserts several pages into the current document, then displays

'a message box with the total number of pages in the document.

```
.Text.InsertText "This is some sample text.", True
```

```
.InsertPageBreak
```

```
.Text.InsertText "This is some sample text.", True
```

```
.InsertPageBreak
```

```
.Text.InsertText "This is some sample text.", True
```

```
MessageBox "There are " & .ActiveDocument.LastPage & " pages in this document.",  
MB_OK, "Example Script"
```

'Example: LeadingText property

'This example sets the leading text option for footnotes and endnotes
'to the open parentheses character.

.Division.FootnoteOptions.FootnoteNumbering.**LeadingText** = "("

.Division.FootnoteOptions.EndnoteDivisionNum.**LeadingText** = "("

.Division.FootnoteOptions.EndnoteDivisionGroupNum.**LeadingText** = "("

.Division.FootnoteOptions.EndnoteDocNum.**LeadingText** = "("

'Example: LeftExternalMargin property

'This example creates a frame, then changes the padding around the frame's
'border to 1/4 inch (360 twips).

```
.NewFrame 4320, 4320, 3387, 1992, "Default Frame"
```

```
.Frame.Layout.LeftExternalMargin = 360
```

```
.Frame.Layout.TopExternalMargin = 360
```

```
.Frame.Layout.RightExternalMargin = 360
```

```
.Frame.Layout.BottomExternalMargin = 360
```

Word Pro: IsTemp property

{button ,AL('H_CHARACTERSTYLE_CLASS;H_PARAGRAPHSTYLE_CLASS',0)} [See list of classes](#)

(Read-write) A value of True indicates that this style is removed when the InUseCount property value drops to 0.

Data Type

Integer

Syntax

istempvalue = [objectreference].IsTemp

[objectreference].IsTemp = istempvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Word Pro: IsTextLocked property

{button ,AL('H_DIVISIONOPTIONS_CLASS',0)} [See list of classes](#)

{button ,AL('H_IStEXTLOCKED_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Determines whether or not protected text in a document can be edited.

Data Type

[Integer](#)

Syntax

istextlockedvalue = [objectreference].IsTextLocked

[objectreference].IsTextLocked = istextlockedvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

This property is equivalent to the "Allow editing of protected text" option on the Other Protection panel of the TeamSecurity dialog box.

{button ,AL('H_PROTECTEDMODE_PROPERTY_MEMDEF',0)} [See related topics](#)

Word Pro: IsTextMenuEnabled property

{button ,AL('H_CONTEXTMENUOPTIONS_CLASS',0)} [See list of classes](#)

(Read-write) Controls whether or not the Text menu displays when the cursor is within a text object.

Data Type

Integer

Syntax

istextmenuenabledvalue = [objectreference].IsTextMenuEnabled

[objectreference].IsTextMenuEnabled = istextmenuenabledvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

A setting of True allows the Text menu to display when the focus is in a text object. A setting of False prevents the display of the Text menu when the focus is in a text object.

Word Pro: IsTOCMarkAble property

```
{button ,AL(^H_LAYOUT_CLASS;H_CELLGROUPLAYOUT_CLASS;H_CELLLAYOUT_CLASS;H_CONNECTEDLAY  
OUT_CLASS;H_COLUMNGROUPLAYOUT_CLASS;H_FOOTERLAYOUT_CLASS;H_FRAMELAYOUT_CLASS;H  
_FRAMEGROUPLAYOUT_CLASS;H_DROP_CAPLAYOUT_CLASS;H_GROUPLAYOUT_CLASS;H_HEADERLAY  
OUT_CLASS;H_NOTELAYOUT_CLASS;H_PAGELAYOUT_CLASS;H_ROWGROUPLAYOUT_CLASS;H_ROWLA  
YOUT_CLASS;H_RUBYLAYOUT_CLASS;H_SUPERTABLEGROUPLAYOUT_CLASS;H_SUPERTABLELAYOUT  
_CLASS;H_TABLELAYOUT_CLASS;H_ENDNOTELAYOUT_CLASS;H_FOOTNOTELAYOUT_CLASS;H_TABLE  
HEADINGLAYOUT_CLASS;H_TOCSUPERTABLELAYOUT_CLASS',0)} See list of classes
```

(Read-only)

This language element has not yet been defined. You can check for a newer Help file on the Lotus web site.

Data Type

Integer

Syntax

istocmarkablevalue = [objectreference].IsTOCMarkAble

Legal values

This property is read-only. The value of this property cannot be set by a script.

Usage

Word Pro: IsTOC property

```
{button ,AL(^H_LAYOUT_CLASS;H_CELLGROUPLAYOUT_CLASS;H_CELLLAYOUT_CLASS;H_CONNECTEDLAY  
OUT_CLASS;H_COLUMNGROUPLAYOUT_CLASS;H_FOOTERLAYOUT_CLASS;H_FRAMELAYOUT_CLASS;H  
_FRAMEGROUPLAYOUT_CLASS;H_DROP_CAPLAYOUT_CLASS;H_GROUPLAYOUT_CLASS;H_HEADERLAY  
OUT_CLASS;H_NOTELAYOUT_CLASS;H_PAGELAYOUT_CLASS;H_ROWGROUPLAYOUT_CLASS;H_ROWLA  
YOUT_CLASS;H_RUBY_LAYOUT_CLASS;H_SUPERTABLEGROUPLAYOUT_CLASS;H_SUPERTABLELAYOUT  
_CLASS;H_TABLELAYOUT_CLASS;H_ENDNOTELAYOUT_CLASS;H_FOOTNOTELAYOUT_CLASS;H_TABLE  
HEADINGLAYOUT_CLASS;H_TOCSUPERTABLELAYOUT_CLASS',0)} See list of classes
```

(Read-only) Indicates whether or not a table is a special TOC table.

Data Type

Integer

Syntax

istocvalue = [objectreference].IsTOC

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Word Pro: IsTrueType property

{button ,AL(^H_FONT_CLASS;H_FONTMETRICS_CLASS',0)} [See list of classes](#)

(Read-only)

Data Type

Integer

Syntax

istruetypevalue = [objectreference].IsTrueType

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Word Pro: IsUndoOn property

{button ,AL('H_DIVISION_CLASS;H_FOUNDRY_CLASS;H_TEXTDOCUMENT_CLASS',0)} [See list of classes](#)

{button ,AL('H_ISUNDOON_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Turns the Undo recorder on (True) or off (False) for the objects in the Foundry object.

Data Type

Integer

Syntax

isundoonvalue = [objectreference].IsUndoOn

[objectreference].IsUndoOn = isundoonvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Word Pro is capable of recording all your actions and allowing you to undo those actions in succession. Word Pro does this by recording any changes made to an object and keeping track of each object's status before and after each change.

Use this property to turn the Undo recorder on or off for all the objects in a particular Foundry object. This is particularly useful when you want to run a lengthy script that makes many changes to a document. By turning the Undo recorder off for each active division's Foundry, you can save memory, prevent users from undoing your script's changes, preserve any changes made by the user, and possibly improve the overall performance of your script.

Word Pro: IsUpdateAutomatic property

{button ,AL('H_DDELINK_CLASS',0)} [See list of classes](#)

(Read-write) Determines whether DDE link updates are automatically sent to Word Pro.

Data Type

Integer

Syntax

isupdateautomaticvalue = [objectreference].IsUpdateAutomatic

[objectreference].IsUpdateAutomatic = isupdateautomaticvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Equivalent to the "Automatic update" option in the Manage Links dialog box. If this property is set to True, DDE links are updated automatically. If set to False, DDE links are not updated automatically.

Word Pro: IsUpdateIndex property

{button ,AL('H_PRINTSETTINGS_CLASS',0)} [See list of classes](#)

{button ,AL('H_ISUPDATEINDEX_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Determines whether or not Word Pro updates the index while printing a document.

Data Type

[Integer](#)

Syntax

isupdateindexvalue = [objectreference].IsUpdateIndex

[objectreference].IsUpdateIndex = isupdateindexvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

A setting of True is equivalent to choosing File - Print, clicking Options, and selecting "Index" in the Update section of the Print Options dialog box.

Word Pro: IsUpdateTOC property

{button ,AL('H_PRINTSETTINGS_CLASS',0)} [See list of classes](#)

{button ,AL('H_ISUPDATETOC_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Determines whether or not Word Pro updates the table of contents while printing a document.

Data Type

[Integer](#)

Syntax

isupdatetocvalue = [objectreference].IsUpdateTOC

[objectreference].IsUpdateTOC = isupdatetocvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Equivalent to choosing File - Print, clicking Options, and selecting "Table of contents" in the Update section of the Print Options dialog box.

Word Pro: IsValid property

{button ,AL(^H_BASEOBJECT_CLASS',0)} [See list of classes](#)

(Read-only) Indicates if the object from which you call this property is available to LotusScript at the time the script is run. This is particularly useful when you need to determine if a MenuItem object is available from the current context.

Data Type

[Integer](#)

Syntax

isvalidvalue = [objectreference].IsValid

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Word Pro often hides or deactivates certain objects under specific circumstances. For example, the Frame menu and its menu items are marked as invalid (IsValid = 0), unless a frame is active in the focus. Use this property when you want to check to see if an object is available to LotusScript from the current context.

Some objects, such as WPAApplication, are always valid.

Word Pro: IsVerticalScrollBarCleanScrn property

{button ,AL('H_WINVIEWPREFS_CLASS',0)} [See list of classes](#)

{button ,AL('H_ISVERTICALSCROLLBARCLEANSCRN_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Determines whether Word Pro displays vertical scroll bars in clean screen mode.

Data Type

[Integer](#)

Syntax

isverticalscrollbarcleanscrnvalue = [objectreference].IsVerticalScrollBarCleanScrn

[objectreference].IsVerticalScrollBarCleanScrn = isverticalscrollbarcleanscrnvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Equivalent to the "Show vertical scroll bar" box on the Clean Screen panel of the View Preferences dialog box. If this property is set to True, Word Pro displays vertical scroll bars in clean screen mode. If set to False, Word Pro does not display vertical scroll bars in clean screen mode.

{button ,AL('H_ISVIEWVERTSCROLLBAR_PROPERTY_MEMDEF',0)} [See related topics](#)

Word Pro: IsViewAnchor property

{button ,AL('H_WINVIEWPREFS_CLASS',0)} [See list of classes](#)

{button ,AL('H_ISVIEWANCHOR_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Determines whether Word Pro displays anchor marks.

Data Type

[Integer](#)

Syntax

isviewanchorvalue = [objectreference].IsViewAnchor

[objectreference].IsViewAnchor = isviewanchorvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Equivalent to the "Anchors" option in the "Show marks" box on the Show panel of the View Preferences dialog box. If this property is set to True, Word Pro displays anchor marks. If set to False, Word Pro does not display anchor marks.

Word Pro: IsViewBookmarks property

{button ,AL('H_WINVIEWPREFS_CLASS',0)} [See list of classes](#)

{button ,AL('H_ISVIEWBOOKMARKS_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Determines whether Word Pro displays bookmark marks.

Data Type

[Integer](#)

Syntax

isviewbookmarksvalue = [objectreference].IsViewBookmarks

[objectreference].IsViewBookmarks = isviewbookmarksvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Equivalent to the "Bookmark" option in the "Show marks" box on the Show panel of the View Preferences dialog box. If this property is set to True, Word Pro displays bookmark marks. If set to False, Word Pro does not display bookmark marks.

Word Pro: IsViewClickHereBlocks property

{button ,AL('H_WINVIEWPREFS_CLASS',0)} [See list of classes](#)

{button ,AL('H_ISVIEWCLICKHEREBLOCKS_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Determines whether Word Pro displays ClickHere blocks.

Data Type

[Integer](#)

Syntax

isviewclickhereblocksvalue = [objectreference].IsViewClickHereBlocks

[objectreference].IsViewClickHereBlocks = isviewclickhereblocksvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Equivalent to the "Show Click Here Blocks" box on the Show panel of the View Preferences dialog box. If this property is set to True, Word Pro displays ClickHere blocks. If set to False, Word Pro does not display ClickHere blocks.

Word Pro: IsViewColGuides property

{button ,AL('H_WINVIEWPREFS_CLASS',0)} [See list of classes](#)

{button ,AL('H_ISVIEWCOLGUIDES_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Determines whether Word Pro displays margin guides.

Data Type

[Integer](#)

Syntax

isviewcolguidesvalue = [objectreference].IsViewColGuides

[objectreference].IsViewColGuides = isviewcolguidesvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Equivalent to the "Show margin guides" box on the Show panel of the View Preferences dialog box. If this property is set to True, Word Pro displays lines that mark the left and right margins. These lines do not print. If set to False, Word Pro does not display the lines that mark the left and right margins.

Word Pro: IsViewColumnBreakMarks property

{button ,AL('H_WINVIEWPREFS_CLASS',0)} [See list of classes](#)

{button ,AL('H_ISVIEWCOLUMNBREAKMARKS_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Determines whether Word Pro displays column break marks.

Data Type

[Integer](#)

Syntax

isviewcolumnbreakmarksvalue = [objectreference].IsViewColumnBreakMarks

[objectreference].IsViewColumnBreakMarks = isviewcolumnbreakmarksvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Equivalent to the "Column breaks" option in the "Show marks" box on the Show panel of the View Preferences dialog box. If this property is set to True, Word Pro displays column break marks. If set to False, Word Pro does not display column break marks.

Word Pro: IsViewDDELinks property

{button ,AL('H_WINVIEWPREFS_CLASS',0)} [See list of classes](#)

{button ,AL('H_ISVIEWDDELINKS_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Determines whether Word Pro displays OLE/DDE link marks.

Note OLE is not supported under OS/2.

Data Type

Integer

Syntax

isviewddelinksvalue = [objectreference].IsViewDDELinks

[objectreference].IsViewDDELinks = isviewddelinksvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Equivalent to the "OLE/DDE marks" option in the "Show marks" box on the Show panel of the Word Pro preferences dialog box. If this property is set to True, Word Pro displays OLE/DDE marks. If set to False, Word Pro does not display OLE/DDE marks.

Word Pro: IsViewHorzRuler property

{button ,AL('H_WINVIEWPREFS_CLASS',0)} [See list of classes](#)

{button ,AL('H_ISVIEWHORZRULER_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Determines whether Word Pro displays the horizontal ruler

Data Type

[Integer](#)

Syntax

isviewhorzrulervalue = [objectreference].IsViewHorzRuler

[objectreference].IsViewHorzRuler = isviewhorzrulervalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Equivalent to the "Show ruler" box on the Show panel of the View Preferences dialog box. If this property is set to True, Word Pro displays the horizontal ruler. If set to False, Word Pro does not display the horizontal ruler.

Word Pro: IsViewHorzScrollBar property

{button ,AL(^H_WINVIEWPREFS_CLASS',0)} [See list of classes](#)

(Read-write) Determines whether Word Pro displays the horizontal scroll bar.

Data Type

[Integer](#)

Syntax

isviewhorzscrollBarvalue = [objectreference].IsViewHorzScrollBar

[objectreference].IsViewHorzScrollBar = isviewhorzscrollBarvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

If this property is set to True, Word Pro displays the horizontal scroll bar. If set to False, Word Pro does not display the horizontal scroll bar.

{button ,AL(^H_ISHORIZONTALSCROLLBARCLEANSCRN_PROPERTY_MEMDEF',0)} [See related topics](#)

Word Pro: IsViewMenuCleanScrn property

{button ,AL('H_WINVIEWPREFS_CLASS',0)} [See list of classes](#)

{button ,AL('H_ISVIEWMENUCLEANSRN_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Determines whether Word Pro displays pull-down menus in clean screen mode.

Data Type

[Integer](#)

Syntax

isviewmenucleanscrnvalue = [objectreference].IsViewMenuCleanScrn

[objectreference].IsViewMenuCleanScrn = isviewmenucleanscrnvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Equivalent to the "Show menu" box on the Clean Screen panel of the View Preferences dialog box. If this property is set to True, Word Pro displays pull-down menus in clean screen mode. If set to False, Word Pro does not display pull-down menus in clean screen mode.

Word Pro: IsViewNotes property

{button ,AL('H_WINVIEWPREFS_CLASS',0)} [See list of classes](#)

{button ,AL('H_ISVIEWNOTES_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Determines whether Word Pro displays comment note marks.

Data Type

[Integer](#)

Syntax

isviewnotesvalue = [objectreference].IsViewNotes

[objectreference].IsViewNotes = isviewnotesvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Equivalent to the "Comment marks" option in the "Show marks" box on the Show panel of the View Preferences dialog box. If this property is set to True, Word Pro displays comment note marks. If set to False, Word Pro does not display comment note marks.

Word Pro: IsViewOutlineFlowToScreen property

{button ,AL('H_WINVIEWPREFS_CLASS',0)} [See list of classes](#)

{button ,AL('H_ISVIEWOUTLINEFLOWTOSCREEN_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Determines whether Word Pro wraps text to fit the size of the window in outline view.

Data Type

[Integer](#)

Syntax

isviewoutlineflowtoscreenvalue = [objectreference].IsViewOutlineFlowToScreen

[objectreference].IsViewOutlineFlowToScreen = isviewoutlineflowtoscreenvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Equivalent to the "Wrap within window" box on the Outline panel of the View Preferences dialog box. If this property is set to True, Word Pro wraps text to fit the size of the window in outline view. If set to False, Word Pro does not wrap text to fit the size of the window in outline view.

Word Pro: IsViewOutlineIndent property

{button ,AL('H_WINVIEWPREFS_CLASS',0)} [See list of classes](#)

{button ,AL('H_ISVIEWOUTLINEINDENT_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Determines whether Word Pro indents text in outline view according to outline level.

Data Type

[Integer](#)

Syntax

isviewoutlineindentvalue = [objectreference].IsViewOutlineIndent

[objectreference].IsViewOutlineIndent = isviewoutlineindentvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Equivalent to the "Show level indents" box on the Outline panel of the View Preferences dialog box. If this property is set to True, Word Pro indents text in outline view according to outline level. If set to False, Word Pro does not indent text in outline view according to outline level.

Word Pro: IsViewPageLayoutMarks property

{button ,AL('H_WINVIEWPREFS_CLASS',0)} [See list of classes](#)

{button ,AL('H_ISVIEWPAGELAYOUTMARKS_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Determines whether Word Pro displays new page style marks.

Data Type

[Integer](#)

Syntax

isviewpagelayoutmarksvalue = [objectreference].IsViewPageLayoutMarks

[objectreference].IsViewPageLayoutMarks = isviewpagelayoutmarksvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Equivalent to the "New page styles" option in the "Show marks" box on the Show panel of the View Preferences dialog box. If this property is set to True, Word Pro displays new page style marks. If set to False, Word Pro does not display new page style marks.

Word Pro: IsViewParallelColumnBorder property

{button ,AL('H_WINVIEWPREFS_CLASS',0)} [See list of classes](#)

{button ,AL('H_ISVIEWPARALLELCOLUMNBORDER_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Determines whether Word Pro displays parallel column grid lines.

Data Type

[Integer](#)

Syntax

isviewparallelcolumnbordervalue = [objectreference].IsViewParallelColumnBorder

[objectreference].IsViewParallelColumnBorder = isviewparallelcolumnbordervalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Equivalent to the "Show parallel column grid lines" box on the Show panel of the View Preferences dialog box. If this property is set to True, Word Pro displays parallel column grid lines. If set to False, Word Pro does not display parallel column grid lines.

Word Pro: IsViewPictures property

{button ,AL('H_WINVIEWPREFS_CLASS',0)} [See list of classes](#)

{button ,AL('H_ISVIEWPICTURES_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Determines whether Word Pro displays graphics.

Data Type

[Integer](#)

Syntax

isviewpicturesvalue = [objectreference].IsViewPictures

[objectreference].IsViewPictures = isviewpicturesvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Equivalent to the "Show graphics" box on the Show panel of the View Preferences dialog box. If this property is set to True, Word Pro displays graphics. If set to False, Word Pro does not display graphics.

Word Pro: IsViewReturnIconCleanScrn property

{button ,AL('H_WINVIEWPREFS_CLASS',0)} [See list of classes](#)

{button ,AL('H_ISVIEWRETURNICONCLEANSCRN_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Determines whether Word Pro displays the return icon to leave clean screen mode.

Data Type

[Integer](#)

Syntax

isviewreturniconcleanscrnvalue = [objectreference].IsViewReturnIconCleanScrn

[objectreference].IsViewReturnIconCleanScrn = isviewreturniconcleanscrnvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Equivalent to the "Show return icon" box on the Clean Screen panel of the View Preferences dialog box. If this property is set to True, Word Pro displays the return icon to leave clean screen mode. If set to False, Word Pro does not display the return icon to leave clean screen mode.

Word Pro: IsViewReturns property

{button ,AL('H_WINVIEWPREFS_CLASS',0)} [See list of classes](#)

{button ,AL('H_ISVIEWRETURNS_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Determines whether Word Pro displays paragraph return marks.

Data Type

Integer

Syntax

isviewreturnsvalue = [objectreference].IsViewReturns

[objectreference].IsViewReturns = isviewreturnsvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Equivalent to the "Returns" option in the "Show marks" box on the Show panel of the View Preferences dialog box. If this property is set to True, Word Pro displays paragraph return marks. If set to False, Word Pro does not display paragraph return marks.

Word Pro: IsViewRubyMarks property

{button ,AL(^H_WINVIEWPREFS_CLASS',0)} [See list of classes](#)

(Read-write) Used only in the Japanese version of Word Pro. Determines whether Ruby marks are displayed in your document.

Data Type

[Integer](#)

Syntax

isviewrubymarksvalue = [objectreference].IsViewRubyMarks

[objectreference].IsViewRubyMarks = isviewrubymarksvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

This property corresponds to "Show Ruby Marks" in the Show Marks box of the View Preferences dialog box.

Word Pro: IsViewRulerMarks property

{button ,AL('H_WINVIEWPREFS_CLASS',0)} [See list of classes](#)

{button ,AL('H_ISVIEWRULERMARKS_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Determines whether Word Pro displays ruler marks.

Data Type

[Integer](#)

Syntax

isviewrulermarksvalue = [objectreference].IsViewRulerMarks

[objectreference].IsViewRulerMarks = isviewrulermarksvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Equivalent to the "Rulers" option in the "Show marks" box on the Show panel of the View Preferences dialog box. If this property is set to True, Word Pro displays ruler marks. If set to False, Word Pro does not display ruler marks.

Word Pro: IsViewSectionBreakMarks property

{button ,AL('H_WINVIEWPREFS_CLASS',0)} [See list of classes](#)

{button ,AL('H_ISVIEWSECTIONBREAKMARKS_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Determines whether Word Pro displays section break marks.

Data Type

[Integer](#)

Syntax

isviewsectionbreakmarksvalue = [objectreference].IsViewSectionBreakMarks

[objectreference].IsViewSectionBreakMarks = isviewsectionbreakmarksvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Equivalent to the "Section breaks" option in the "Show marks" box on the Show panel of the View Preferences dialog box. If this property is set to True, Word Pro displays section break marks. If set to False, Word Pro does not display section break marks.

Word Pro: IsViewSectionTabs property

{button ,AL('H_WINVIEWPREFS_CLASS',0)} [See list of classes](#)

{button ,AL('H_ISVIEWSECTIONTABS_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Determines whether Word Pro displays divider tabs.

Data Type

[Integer](#)

Syntax

isviewsectiontabsvalue = [objectreference].IsViewSectionTabs

[objectreference].IsViewSectionTabs = isviewsectiontabsvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Equivalent to the "Show divider tabs" box on the Show panel of the View Preferences dialog box. If this property is set to True, Word Pro displays divider tabs. If set to False, Word Pro does not display divider tabs.

Word Pro: IsViewSmartIconsCleanScrn property

{button ,AL('H_WINVIEWPREFS_CLASS',0)} [See list of classes](#)

{button ,AL('H_ISVIEWSMARTICONSCLEANSKRN_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Determines whether Word Pro displays SmartIcons in clean screen mode.

Data Type

[Integer](#)

Syntax

isviewsmarticonscleanscrnvalue = [objectreference].IsViewSmartIconsCleanScrn

[objectreference].IsViewSmartIconsCleanScrn = isviewsmarticonscleanscrnvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Equivalent to the "Show SmartIcons" box on the Clean Screen panel of the View Preferences dialog box. If this property is set to True, Word Pro displays SmartIcons in clean screen mode. If set to False, Word Pro does not display SmartIcons in clean screen mode.

Word Pro: IsViewStatusBarCleanScrn property

{button ,AL('H_WINVIEWPREFS_CLASS',0)} [See list of classes](#)

{button ,AL('H_ISVIEWSTATUSBARCLEANSCRN_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Determines whether Word Pro displays the status bar in clean screen mode.

Data Type

[Integer](#)

Syntax

isviewstatusbarcleanscrnvalue = [objectreference].IsViewStatusBarCleanScrn

[objectreference].IsViewStatusBarCleanScrn = isviewstatusbarcleanscrnvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Equivalent to the "Show status bar" box on the Clean Screen panel of the View Preferences dialog box. If this property is set to True, Word Pro displays the status bar in clean screen mode. If set to False, Word Pro does not display the status bar in clean screen mode.

Word Pro: IsViewStatusSpell property

{button ,AL(^H_WINVIEWPREFS_CLASS',0)} [See list of classes](#)

(Read-write) This property is for internal use only.

Data Type

Integer.

Syntax

[objectreference].IsViewStatusSpell = isviewstatusspellvalue

isviewstatusspellvalue = [objectreference].IsViewStatusSpell

Legal values**Usage**

This property is for internal use only.

Word Pro: IsViewTableCellSelection property

{button ,AL('H_WINVIEWPREFS_CLASS',0)} [See list of classes](#)

(Read-write) Determines whether Word Pro surrounds selected table cells with a thick, non-printing border.

Data Type

Integer

Syntax

isviewtablecellselectionvalue = [objectreference].IsViewTableCellSelection

[objectreference].IsViewTableCellSelection = isviewtablecellselectionvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

If this property is set to True, Word Pro surrounds selected table cells with a thick, non-printing border. If set to False, Word Pro does not surround the selected table cells with a border. Only the contents of selected table cells appear as selected.

Word Pro: IsViewTableGridLines property

{button ,AL(^H_WINVIEWPREFS_CLASS',0)} [See list of classes](#)

(Read-write) Determines whether Word Pro displays grid lines around all table cells.

Data Type

Integer

Syntax

isviewtablegridlinesvalue = [objectreference].IsViewTableGridLines

[objectreference].IsViewTableGridLines = isviewtablegridlinesvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Equivalent to the "Show table guides" box on the Show panel of the View Preferences dialog box. If this property is set to True, Word Pro displays table grid lines. If set to False, Word Pro does not display table grid lines.

Word Pro: IsViewTableHeadings property

{button ,AL('H_WINVIEWPREFS_CLASS',0)} [See list of classes](#)

{button ,AL('H_ISVIEWTABLEHEADINGS_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Determines whether Word Pro displays table row and column headings.

Data Type

[Integer](#)

Syntax

isviewtableheadingsvalue = [objectreference].IsViewTableHeadings

[objectreference].IsViewTableHeadings = isviewtableheadingsvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Equivalent to the "Show table row/col. headings" box on the Show panel of the View Preferences dialog box. If this property is set to True, Word Pro displays table row and column headings. If set to False, Word Pro does not display table row and column headings.

Word Pro: IsViewTabs property

{button ,AL('H_WINVIEWPREFS_CLASS',0)} [See list of classes](#)

{button ,AL('H_ISVIEWTABS_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Determines whether Word Pro displays tab marks.

Data Type

[Integer](#)

Syntax

isviewtabsvalue = [objectreference].IsViewTabs

[objectreference].IsViewTabs = isviewtabsvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Equivalent to the "Tabs" option in the "Show marks" box on the Show panel of the View Preferences dialog box. If this property is set to True, Word Pro displays tab marks. If set to False, Word Pro does not display tab marks.

Word Pro: IsViewThumbBarCleanScrn property

{button ,AL('H_WINVIEWPREFS_CLASS',0)} [See list of classes](#)

{button ,AL('H_ISVIEWTHUMBBARCLEANSCRN_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Determines whether or not the page gauge displays.

Data Type

[Integer](#)

Syntax

isviewthumbbarcleanscrnvalue = [objectreference].IsViewThumbBarCleanScrn

[objectreference].IsViewThumbBarCleanScrn = isviewthumbbarcleanscrnvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

If this property is set to True, the page gauge displays. If this property is set to False, the page gauge doesn't display.

This property is equivalent to the "Show page gauge" box on the Show panel of the View Preferences dialog box.

Word Pro: IsViewTitleBarCleanScrn property

{button ,AL(`H_WINVIEWPREFS_CLASS',0)} [See list of classes](#)

{button ,AL(`H_ISVIEWTITLEBARCLEANSCRN_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Determines whether Word Pro displays the application title bar in clean screen mode.

Data Type

[Integer](#)

Syntax

isviewtitlebarcleanscrnvalue = [objectreference].IsViewTitleBarCleanScrn

[objectreference].IsViewTitleBarCleanScrn = isviewtitlebarcleanscrnvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Equivalent to the "Show title bar" box on the Clean Screen panel of the View Preferences dialog box.

When setting this property to True, the IsViewMenuCleanScrn property should also be set to True. If only the IsViewTitleBarCleanScrn property is set to True, the menu bar displays in clean screen mode but the application title bar does not. If set to False, the application title bar does not display in clean screen mode.

{button ,AL(`H_ISVIEWMENUCLEANSCRN_PROPERTY_MEMDEF',0)} [See related topics](#)

Word Pro: IsViewVertRuler property

{button ,AL('H_WINVIEWPREFS_CLASS',0)} [See list of classes](#)

{button ,AL('H_ISVIEWVERTRULER_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Determines whether Word Pro displays the vertical ruler.

Data Type

[Integer](#)

Syntax

isviewvertrulervalue = [objectreference].IsViewVertRuler

[objectreference].IsViewVertRuler = isviewvertrulervalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Equivalent to the "Show vertical ruler" box on the Show panel of the View Preferences dialog box.

If this property is set to True, Word Pro displays the vertical ruler. If set to False, Word Pro does not display the vertical ruler.

Word Pro: IsViewVertScrollBar property

{button ,AL(^H_WINVIEWPREFS_CLASS',0)} [See list of classes](#)

(Read-write) Determines whether Word Pro displays the vertical scroll bar.

Data Type

[Integer](#)

Syntax

isviewvertscrollbarvalue = [objectreference].IsViewVertScrollBar

[objectreference].IsViewVertScrollBar = isviewvertscrollbarvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

If this property is set to True, Word Pro displays the vertical scroll bar. If set to False, Word Pro does not display the vertical scroll bar.

{button ,AL(^H_ISVERTICALSCROLLBARCLEANSCRN_PROPERTY_MEMDEF',0)} [See related topics](#)

Word Pro: IsWordProChart property

{button ,AL('H_GRAPHIC_CLASS',0)} [See list of classes](#)

{button ,AL('H_ISWORDPROCHART_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-only) Tests to see if the chart is a 16-bit Word Pro or Ami Pro chart.

Data Type

[Integer](#)

Syntax

iswordprochartvalue = [objectreference].IsWordProChart

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Word Pro: IsWorkingDir property

{button ,AL('H_USERINTERFACEPREFS_CLASS',0)} [See list of classes](#)

{button ,AL('H_ISWORKINGDIR_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Indicates if "Use working directory" is enabled in Word Pro Preferences.

Data Type

Integer (Bool)

Syntax

isworkingdirvalue = [objectreference].IsWorkingDir

[objectreference].IsWorkingDir =isworkingdirvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values. Default value is False (0).

Usage

Equivalent to "Use working directory" on the Locations panel of the Word Pro Preferences dialog box. If this property is set to True, Word Pro continues to use the folder you chose when opening or saving a document. Word Pro uses this folder when it displays file names in the Open and Save As dialog boxes. If set to False, Word Pro uses the default document directory.

Word Pro: Italic property

{button ,AL('H_FONT_CLASS;H_FONTMETRICS_CLASS',0)} [See list of classes](#)

{button ,AL('H_ITALIC_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write)

Data Type

Integer

Syntax

italicvalue = [objectreference].Italic

[objectreference].Italic = italicvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Word Pro: Items property

{button ,AL('H_MENUITEM_CLASS',0)} [See list of classes](#)

(Read-only) A collection item used by the current parent menu item to hold submenu items.

Data Type

[MenuItemCollection](#)

Syntax

itemsvalue = [objectreference].Items

Legal values

Always contains an instance of the MenuItemCollection class.

Usage

This property allows you to access submenu items for a specific parent menu item. For example, the Word Pro main menu resides in the LWPMenuBar property in the ApplicationWindow class. If you want to access the Word Pro File menu, you must use the Items property to return the File menu in the LWPMenuBar. Menu items in the Items property are enumerated by their Caption property.

In order to access the Close menu item in the File menu, you can use the following statement:

```
.ApplicationWindow.LWPMenuBar.Items("&File").Items("&Close")
```

Word Pro: JoinCorners property

{button ,AL(^H_JOIN_CLASS',0)} [See list of classes](#)

(Read-write) Sets or returns the corners of a layout object to which a particular join type should be applied.

Data Type

Data type is Variant which allows the value of this property to be one of the constants listed below or its numeric equivalent (in parentheses).

Syntax

joincornersvalue = [objectreference].JoinCorners

[objectreference].JoinCorners = joincornersvalue

Legal values

<u>Value</u>	<u>Effect</u>
\$LwpJoinCornerPositionAll (426)	Applies a specific join type to all corners of a layout object.
\$LwpJoinCornerPositionBottomleft (425)	Applies a specific join type to the bottom left corner of a layout object.
\$LwpJoinCornerPositionBottomright (424)	Applies a specific join type to the bottom right corner of a layout object.
\$LwpJoinCornerPositionTopleft (422)	Applies a specific join type to top left corner of a layout object.
\$LwpJoinCornerPositionTopright (423)	Applies a specific join type to top right corner of a layout object.

Usage

Word Pro: JoinHeight property

{button ,AL('H_JOIN_CLASS',0)} [See list of classes](#)

(Read-write) Defines the height of the bounding rectangular box of a scaleable join object.

Data Type

Long

Syntax

joinheightvalue = [objectreference].JoinHeight

[objectreference].JoinHeight = joinheightvalue

Legal values

Data type is Long but the unit of measurement used for this property is Twips. There are 1440 Twips per inch.

Usage

A join object can be scaleable or fixed.

Scaleable join objects

You can set the ScaleMode property of a scaleable join object to scaling or no scaling. Setting this property to scaling causes the join object's width and height to be a function of the layout object's height and width, and also the Percentage property. (For more information, see ScaleMode property and Percentage property.)

Setting this property to no scaling causes the width and height of a join object to be the same as the width and height properties of the join object. The width and height of the join object does not change as the layout object's width and height changes.

The width and height of a scaleable join with a no scaling setting are fixed. However, you can change the width and height properties of a join object.

Fixed join objects

A fixed join object has predefined Word Pro width and height properties. Therefore, you cannot change its width or height.

Word Pro: JoinName property

{button ,AL('H_JOIN_CLASS',0)} [See list of classes](#)

{button ,AL('H_JOINNAME_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write)

This language element has not yet been defined. You can check for a newer Help file on the Lotus web site.

Data Type

String

Syntax

joinnamevalue = [objectreference].JoinName

[objectreference].JoinName = joinnamevalue

Legal values

Any value of type String.

Usage

{button ,AL('H_NAMEOFEXTERNALBORDER_PROPERTY_MEMDEF',0)} [See related topics](#)

Word Pro: JoinType property

{button ,AL(^H_JOIN_CLASS',0)} [See list of classes](#)

(Read-write) Defines the type or style of the join object.

Data Type

Data type is [Variant](#), which allows the value of this property to be one of the constants listed below or its numeric equivalent (in parentheses).

Syntax

jointypevalue = [objectreference].JoinType

[objectreference].JoinType = jointypevalue

Legal values

A join type object can be scaleable or fixed. For more information on scaleable and fixed join type objects, see the Usage section below.

<u>Value</u>	<u>Effect</u>
\$LwpJoinTypeDeco1 (543)	<i>Fixed.</i> Word Pro automatically sets the join type to no scaling since this join type cannot be scaled.
\$LwpJoinTypeDeco2 (544)	<i>Fixed.</i> Word Pro automatically sets the join type to no scaling since this join type cannot be scaled.
\$LwpJoinTypeDeco3 (549)	<i>Fixed.</i> Word Pro automatically sets the join type to no scaling since this join type cannot be scaled.
\$LwpJoinTypeDesktop (539)	<i>Scaleable.</i> This join type can either be fixed or scaled.
\$LwpJoinTypeDogear (538)	<i>Scaleable.</i> This join type can either be fixed or scaled.
\$LwpJoinTypeGirder (2389)	
\$LwpJoinTypeNone (530)	<i>Scaleable.</i> This join type can either be fixed or scaled.
\$LwpJoinTypePin (546)	<i>Fixed.</i> Word Pro automatically sets the join type to no scaling since this join type cannot be scaled.
\$LwpJoinTypeRain (545)	<i>Fixed.</i> Word Pro automatically sets the join type to no scaling since this join type cannot be scaled.
\$LwpJoinTypeRect (534)	<i>Scaleable.</i> This join type can either be fixed or scaled.
\$LwpJoinTypeRope (542)	<i>Fixed.</i> Word Pro automatically sets the join type to no scaling since this join type cannot be scaled.
\$LwpJoinTypeRose (547)	<i>Fixed.</i> Word Pro automatically sets the join type to no scaling since this join type cannot be scaled.
\$LwpJoinTypeRounded (533)	<i>Scaleable.</i> This join type can either be fixed or scaled.
\$LwpJoinTypeSunf (548)	<i>Fixed.</i> Word Pro automatically sets the join type to no scaling since this join type cannot be scaled.
\$LwpJoinTypeWarning (2386)	

Usage

Scaleable join objects

You can set the ScaleMode property of a scaleable join object to scaling or no scaling. Setting this property to scaling causes the join object's width and height to be a function of the layout object's height and width, and also the Percentage property. (For more information, see ScaleMode property and Percentage property.)

Setting this property to no scaling causes the width and height of a join object to be the same as the width and height properties of the join object. The width and height of the join object does not change as the layout object's width and height changes.

The width and height of a scaleable join with a no scaling setting are fixed. However, you can change the width and height properties of a join object.

Fixed join objects

A fixed join object has predefined Word Pro width and height properties. Therefore, you cannot change its width or height.

Word Pro: JoinWidth property

{button ,AL(^H_JOIN_CLASS',0)} [See list of classes](#)

(Read-write) Defines the width of the bounding rectangular box of a scaleable join object.

Data Type

Long

Syntax

joinwidthvalue = [objectreference].JoinWidth

[objectreference].JoinWidth = joinwidthvalue

Legal values

Data type is Long but the unit of measurement used for this property is Twips. There are 1440 Twips per inch.

Usage

Use the JoinWidth property to define the width of the bounding rectangular box of a scalable join object. A join object can be scalable or fixed.

Scaleable join objects

You can set the ScaleMode property of a scaleable join object to scaling or no scaling. Setting this property to scaling causes the join object's width and height to be a function of the layout object's height and width, and also the Percentage property. (For more information, see ScaleMode property and Percentage property.)

Setting this property to no scaling causes the width and height of a join object to be the same as the width and height properties of the join object. The width and height of the join object does not change as the layout object's width and height changes.

The width and height of a scaleable join with a no scaling setting are fixed. However, you can change the width and height properties of a join object.

Fixed join objects

A fixed join object has predefined Word Pro width and height properties. Therefore, you cannot change its width or height.

Word Pro: Join property

```
{button ,AL('H_LAYOUT_CLASS;H_CELLGROUPLAYOUT_CLASS;H_CELLLAYOUT_CLASS;H_CONNECTEDLAY  
OUT_CLASS;H_COLUMNGROUPLAYOUT_CLASS;H_FOOTERLAYOUT_CLASS;H_FRAMELAYOUT_CLASS;H  
_FRAMEGROUPLAYOUT_CLASS;H_DROP_CAPLAYOUT_CLASS;H_GROUPLAYOUT_CLASS;H_HEADERLAY  
OUT_CLASS;H_NOTELAYOUT_CLASS;H_PAGELAYOUT_CLASS;H_ROWGROUPLAYOUT_CLASS;H_ROWLA  
YOUT_CLASS;H_RUBYLAYOUT_CLASS;H_SUPERTABLEGROUPLAYOUT_CLASS;H_SUPERTABLELAYOUT  
_CLASS;H_TABLELAYOUT_CLASS;H_ENDNOTELAYOUT_CLASS;H_FOOTNOTELAYOUT_CLASS;H_TABLE  
HEADINGLAYOUT_CLASS;H_TOCSUPERTABLELAYOUT_CLASS',0)} See list of classes
```

(Read-only) Allows you to access the join object for a specific layout object.

Data Type

[Join](#)

Syntax

joinvalue = [objectreference].Join

Legal values

Always contains an instance of the Join class.

Usage

Word Pro: Justifiable property

```
{button ,AL('H_LAYOUT_CLASS;H_CELLGROUPLAYOUT_CLASS;H_CELLLAYOUT_CLASS;H_CONNECTEDLAY  
OUT_CLASS;H_COLUMNGROUPLAYOUT_CLASS;H_FOOTERLAYOUT_CLASS;H_FRAMELAYOUT_CLASS;H  
_FRAMEGROUPLAYOUT_CLASS;H_DROP_CAPLAYOUT_CLASS;H_GROUPLAYOUT_CLASS;H_HEADERLAY  
OUT_CLASS;H_NOTELAYOUT_CLASS;H_PAGELAYOUT_CLASS;H_ROWGROUPLAYOUT_CLASS;H_ROWLA  
YOUT_CLASS;H_RUBYLAYOUT_CLASS;H_SUPERTABLEGROUPLAYOUT_CLASS;H_SUPERTABLELAYOUT  
_CLASS;H_TABLELAYOUT_CLASS;H_ENDNOTELAYOUT_CLASS;H_FOOTNOTELAYOUT_CLASS;H_TABLE  
HEADINGLAYOUT_CLASS;H_TOCSUPERTABLELAYOUT_CLASS',0)} See list of classes
```

```
{button ,AL('H_JUSTIFIABLE_PROPERTY_EXSCRIPT',1)} See example
```

(Read-write) Expands the layout object to match the width of the parent content area, not the content itself.

Data Type

[Integer](#)

Syntax

justifiablevalue = [objectreference].Justifiable

[objectreference].Justifiable = justifiablevalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values. Default is False.

Usage

Word Pro: KeepWithNext property

{button ,AL('H_KEEPWITHNEXT_PROPERTY_EXSCRIPT',1)} [See example](#)

{button ,AL('H_BREAKS_CLASS',0)} [See list of classes](#)

(Read-write)

Data Type

[Integer](#)

Syntax

keepwithnextvalue = [objectreference].KeepWithNext

[objectreference].KeepWithNext = keepwithnextvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Word Pro: KeepWithPrev property

{button ,AL('H_BREAKS_CLASS',0)} [See list of classes](#)

{button ,AL('H_KEEPPREV_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write)

Data Type

[Integer](#)

Syntax

keepwithprevvalue = [objectreference].KeepWithPrev

[objectreference].KeepWithPrev = keepwithprevvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Word Pro: KeyboardLanguage property

{button ,AL(^H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

(Read-write) The current text language being used by Word Pro. Word Pro uses the language specified in this property to set the language for typed text. The value of this property does not indicate the language setting for text in the focus.

Data Type

Integer

Syntax

keyboardlanguagevalue = [objectreference].KeyboardLanguage

[objectreference].KeyboardLanguage = keyboardlanguagevalue

Legal values

You can change the value of this property but you must use one of the Windows 95 language codes listed in the table below. Note that this table contains the codes twice: first sorted by name on the left and then sorted by code on the right.

<u>Language Codes By Name</u>		<u>Language Codes by Number</u>	
AFRIKAANS	1078	0	SYSTEM
ALBANIAN	1052	10241	ARABIC SYRIA
AMERICAN	1033	1025	ARABIC
ARABIC EGYPT	3073	10250	SPANISH PERU
ARABIC MOROCCO	6145	1029	CZECH
ARABIC	1025	1037	HEBREW
ARABIC BAHRAIN	15361	1038	HUNGARIAN
ARABIC JORDAN	11265	1030	DANISH
ARABIC QATAR	16385	1039	ICELANDIC
ARABIC IRAQ	2049	1040	ITALIAN
ARABIC SYRIA	10241	1026	BULGARIAN
ARABIC TUNISIA	7169	1031	GERMAN
ARABIC LEBANON	12289	1032	GREEK
ARABIC LIBYA	4097	1027	CATALAN
ARABIC OMAN	8193	1033	AMERICAN
ARABIC KUWAIT	13313	1034	SPANISH
ARABIC ALGERIA	5121	1028	CHINESE TRADITIONAL
ARABIC YEMEN	9217	1035	FINNISH
ARABIC UAE	14337	1036	FRENCH
AUSTRALIAN	3081	1041	JAPANESE
BASQUE	1069	1042	KOREAN
BRAZILIAN	1046	1043	DUTCH
BRITISH	2057	1044	NORWEGIAN
BRITISHMEDIZE	35849	1045	POLISH
BRMEDICAL	34825	1046	BRAZILIAN
BULGARIAN	1026	1047	RHAETO ROMAN
BYELORUSSIAN	1059	1048	ROMANIAN
CATALAN	1027	1049	RUSSIAN
CHINESE PRCHINA	2052	1053	SWEDISH

CHINESE HONGKONG	3076	1050	CROATIAN
CHINESE SINGAPORE	4100	1051	SLOVAK
CHINESE TRADITIONAL	1028	1052	ALBANIAN
CROATIAN CYRILLIC	2074	1054	THAI
CROATIAN SERBIAN	3098	1055	TURKISH
CROATIAN	1050	1056	URDU
CZECH	1029	1057	INDONESIAN
DANISH	1030	1058	UKRAINIAN
DUTCH BELGIAN	2067	1059	BYELORUSSIAN
DUTCH	1043	1060	SLOVENE
ENGLISH CANADIAN	4105	1061	ESTONIAN
ENGLISH NEWZEALAND	5129	1062	LATVIAN
ENGLISH IRELAND	6153	1063	LITHUANIAN
ENGLISH SAFRICA	7177	1065	FARSI
ENGLISH JAMAICA	8201	1069	BASQUE
ENGLISH CARRIBEAN	9225	1070	SORBIAN
ESTONIAN	1061	1078	AFRIKAANS
FAEROESE	1080	1080	FAEROESE
FARSI	1065	11265	ARABIC JORDAN
FINNISH	1035	11274	SPANISH ARGENTINA
FRENCH BELGIAN	2060	12289	ARABIC LEBANON
FRENCH CANADIAN	3084	12298	SPANISH ECUADOR
FRENCH SWISS	4108	13313	ARABIC KUWAIT
FRENCH LUXEMBOURG	5132	13322	SPANISH CHILE
FRENCH	1036	14337	ARABIC UAE
GERMAN SWISS	2055	16394	SPANISH BOLIVIA
GERMAN AUSTRIAN	3079	14346	SPANISH URUGUAY
GERMAN	1031	15361	ARABIC BAHRAIN
GERMAN LIECHTENSTEIN	5127	15370	SPANISH PARAGUAY
GERMAN LUXEMBOURG	4103	16385	ARABIC QATAR
GREEK	1032	2049	ARABIC IRAQ
HEBREW	1037	2052	CHINESE PRCHINA
HUNGARIAN	1038	2055	GERMAN SWISS
ICELANDIC	1039	2057	BRITISH
INDONESIAN	1057	2058	SPANISH MEXICAN
ITALIAN SWISS	2064	2060	FRENCH BELGIAN
ITALIAN	1040	2064	ITALIAN SWISS
JAPANESE	1041	2066	KOREAN JOHAB
KOREAN	1042	2067	DUTCH BELGIAN
KOREAN JOHAB	2066	2068	NYNORSK
LATVIAN	1062	2070	PORTUGUESE
LITHUANIAN	1063	2074	CROATIAN CYRILLIC
MEDICAL	33801	3073	ARABIC EGYPT
NORWEGIAN	1044	3076	CHINESE HONGKONG
NYNORSK	2068	3079	GERMAN AUSTRIAN

POLISH	1045	3081	AUSTRALIAN
PORTUGUESE	2070	3082	SPANISH MODERN
RHAETO ROMAN	1047	3084	FRENCH CANADIAN
ROMANIAN	1048	3091	VOORKEUR
RUSSIAN	1049	3098	CROATIAN SERBIAN
RUSSIANIO	33817	33801	MEDICAL
SLOVAK	1051	33817	RUSSIANIO
SLOVENE	1060	34825	BRMEDICAL
SORBIAN	1070	35849	BRITISHMEDIZE
SPANISH ECUADOR	12298	4100	CHINESE SINGAPORE
SPANISH DOMINICAN	7178	5130	SPANISH COSTARICA
SPANISH MODERN	3082	4108	FRENCH SWISS
SPANISH ARGENTINA	11274	65535	UNIVERSAL
SPANISH PANAMA	6154	4106	SPANISH GUATEMALA
SPANISH PERU	10250	6154	SPANISH PANAMA
SPANISH VENEZUELA	8202	4103	GERMAN LUXEMBOURG
SPANISH MEXICAN	2058	7169	ARABIC TUNISIA
SPANISH CHILE	13322	4105	ENGLISH CANADIAN
SPANISH COLOMBIA	9226	6145	ARABIC MOROCCO
SPANISH BOLIVIA	16394	5121	ARABIC ALGERIA
SPANISH URUGUAY	14346	4097	ARABIC LIBYA
SPANISH COSTARICA	5130	5127	GERMAN LIECHTENSTEIN
SPANISH	1034	5129	ENGLISH NEWZEALAND
SPANISH GUATEMALA	4106	5132	FRENCH LUXEMBOURG
SPANISH PARAGUAY	15370	6153	ENGLISH IRELAND
SWEDISH	1053	7177	ENGLISH SAFRICA
SYSTEM	0	7178	SPANISH DOMINICAN
THAI	1054	8193	ARABIC OMAN
TURKISH	1055	8201	ENGLISH JAMAICA
UKRAINIAN	1058	8202	SPANISH VENEZUELA
UNIVERSAL	65535	9217	ARABIC YEMEN
URDU	1056	9225	ENGLISH CARRIBEAN
VOORKEUR	3091	9226	SPANISH COLOMBIA

Usage

When you type text in a Word Pro document, Word Pro assigns a language to that text. Any text you type is assigned the language shown in this property. If you change the language code in this property, all typed text is assigned the new language, regardless of where you type the text. When you perform a Spell Check, Word Pro uses the text's language to select the appropriate dictionary.

Word Pro does not use this property for language settings, unless you specify this option in the Word Pro Preferences dialog box by choosing File - User Setup - Word Pro Preferences. In the Word Pro Preferences dialog box, click the General tab and, in the Keyboard options, specify "language sets text's language."

Word Pro: Keywords property

{button ,AL('H_DOCINFO_CLASS',0)} [See list of classes](#)

{button ,AL('H_KEYWORDS_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) The user-defined keywords associated with a document.

Data Type

String

Syntax

keywordsvalue = [objectreference].Keywords

[objectreference].Keywords = keywordsvalue

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Word Pro: Kinsoku property

{button ,AL(^H_CLICKHERE_CLASS;H_TEXTMARKER_CLASS;H_PARAGRAPHSTYLE_CLASS;H_TEXT_CLASS',
0)} [See list of classes](#)

(Read-only)

Data Type

[Kinsoku](#)

Syntax

kinsokuvalue = [objectreference].Kinsoku

Legal values

Always contains an instance of the Kinsoku class.

Usage

Word Pro: LandscapeMode property

{button ,AL('H_PAGELAYOUT_CLASS',0)} [See list of classes](#)

{button ,AL('H_LANDSCAPEMODE_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Allows you to set the width of the layout object to its height and set the height of the layout object to its width.

Data Type

[Integer](#)

Syntax

landscapemodevalue = [objectreference].LandscapeMode

[objectreference].LandscapeMode = landscapemodevalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Word Pro: Language property

{button ,AL('H_APPLICATION_CLASS;H_WPAPPLICATION_CLASS;H_LANGUAGE_CLASS;H_CHARACTERSTYLE_CLASS;H_DIVISIONOPTIONS_CLASS;H_CLICKHERE_CLASS;H_TEXTMARKER_CLASS;H_PARAGRAPHS_TYLE_CLASS;H_TEXT_CLASS';0)} [See list of classes](#)

{button ,AL('H_LANGUAGE_PROPERTY_EXSCRIPT',1)} [See example](#)

[Language]

(Read-write) Determines the specific language type of an object.

[CharacterStyle] [ClickHere] [DivisionOptions] [FormatPreferences] [ParagraphStyle] [Text] [TextMarker]

(Read-only) Contains an instance of the Language class.

Data Type

[Language](#)

Syntax

languagevalue = [objectreference].Language

Legal values

[CharacterStyle] [ClickHere] [DivisionOptions] [FormatPreferences] [ParagraphStyle] [Text] [TextMarker]

Always contains an instance of the Language class.

[Language]

\$LwpLanguagesAfrikaans (474)

\$LwpLanguagesAlbanian (475)

\$LwpLanguagesAmerican (442)

\$LwpLanguagesArabic (1025)

\$LwpLanguagesArabicAlgeria (5121)

\$LwpLanguagesArabicBahrain (15361)

\$LwpLanguagesArabicEgypt (3073)

\$LwpLanguagesArabicIraq (2049)

\$LwpLanguagesArabicJordan (11265)

\$LwpLanguagesArabicKuwait (13313)

\$LwpLanguagesArabicLebanon (12289)

\$LwpLanguagesArabicLibya (4097)

\$LwpLanguagesArabicMorocco (6145)

\$LwpLanguagesArabicOman (8193)

\$LwpLanguagesArabicQatar (16385)

\$LwpLanguagesArabicSyria (10241)

\$LwpLanguagesArabicTunisia (7169)

\$LwpLanguagesArabicUAE (14337)

\$LwpLanguagesArabicYemen (9217)

\$LwpLanguagesAustralian (444)

\$LwpLanguagesBasque (1069)

\$LwpLanguagesBrazilian (468)

\$LwpLanguagesBritish (443)

\$LwpLanguagesBritishize (12297)

\$LwpLanguagesBritishmedize (13321)

\$LwpLanguagesBrmedical (11273)

\$LwpLanguagesBulgarian (478)

\$LwpLanguagesByelorussian (1059)

\$LwpLanguagesCatalan (436)
\$LwpLanguagesChineseHongKong (3076)
\$LwpLanguagesChinesePRChina (481)
\$LwpLanguagesChineseSingapore (4100)
\$LwpLanguagesChineseTraditional (479)
\$LwpLanguagesCroatian (1050)
\$LwpLanguagesCroatianCyrillic (2074)
\$LwpLanguagesCroatianLatin (1050)
\$LwpLanguagesCroatianSerbian (3098)
\$LwpLanguagesCzech (437)
\$LwpLanguagesDanish (438)
\$LwpLanguagesDutch (439)
\$LwpLanguagesDutchBelgian (440)
\$LwpLanguagesEnglishCanadian (445)
\$LwpLanguagesEnglishCaribbean (9225)
\$LwpLanguagesEnglishIreland (447)
\$LwpLanguagesEnglishJamaica (8201)
\$LwpLanguagesEnglishNewzealand (446)
\$LwpLanguagesEnglishSAfrica (7177)
\$LwpLanguagesEstonian (1061)
\$LwpLanguagesFaeroese (1080)
\$LwpLanguagesFarsi (1081)
\$LwpLanguagesFinnish (452)
\$LwpLanguagesFrench (453)
\$LwpLanguagesFrenchBelgian (454)
\$LwpLanguagesFrenchCanadian (455)
\$LwpLanguagesFrenchLuxembourg (5132)
\$LwpLanguagesFrenchSwiss (456)
\$LwpLanguagesGerman (457)
\$LwpLanguagesGermanAustrian (459)
\$LwpLanguagesGermanLiechtenstein (5127)
\$LwpLanguagesGermanLuxembourg (4103)
\$LwpLanguagesGermanSwiss (458)
\$LwpLanguagesGreek (460)
\$LwpLanguagesHebrew (483)
\$LwpLanguagesHungarian (461)
\$LwpLanguagesIcelandic (484)
\$LwpLanguagesIndonesian (1057)
\$LwpLanguagesItalian (462)
\$LwpLanguagesItalianSwiss (463)
\$LwpLanguagesJapanese (485)
\$LwpLanguagesKorean (486)
\$LwpLanguagesKoreanJohab (2066)
\$LwpLanguagesLatvian (1062)
\$LwpLanguagesLithuanian (1063)
\$LwpLanguagesMedical (448)
\$LwpLanguagesNorwegian (464)

\$LwpLanguagesNynorsk (465)
\$LwpLanguagesPolish (466)
\$LwpLanguagesPortuguese (467)
\$LwpLanguagesRhaetoRoman (487)
\$LwpLanguagesRomanian (488)
\$LwpLanguagesRussian (469)
\$LwpLanguagesRussianio (470)
\$LwpLanguagesSlovak (492)
\$LwpLanguagesSlovene (493)
\$LwpLanguagesSorbian (1070)
\$LwpLanguagesSpanish (471)
\$LwpLanguagesSpanishArgentina (11274)
\$LwpLanguagesSpanishBolivia (16394)
\$LwpLanguagesSpanishChile (13222)
\$LwpLanguagesSpanishColombia (9226)
\$LwpLanguagesSpanishCostaRica (5130)
\$LwpLanguagesSpanishDominican (7178)
\$LwpLanguagesSpanishEcuador (12298)
\$LwpLanguagesSpanishGuatemala (4106)
\$LwpLanguagesSpanishMexican (2058)
\$LwpLanguagesSpanishPanama (6154)
\$LwpLanguagesSpanishParaguay (15370)
\$LwpLanguagesSpanishPeru (10250)
\$LwpLanguagesSpanishUruguay (14346)
\$LwpLanguagesSpanishVenezuela (8202)
\$LwpLanguagesSwedish (473)
\$LwpLanguagesSystem (434)
\$LwpLanguagesThai (494)
\$LwpLanguagesTurkish (495)
\$LwpLanguagesUkrainian (496)
\$LwpLanguagesUniversal (435)
\$LwpLanguagesUrdu (497)
\$LwpLanguagesVoorkeur (441)

Usage

[Language]

The Language property of the Language class determines the specific language of an object.

[CharacterStyle] [ClickHere] [DivisionOptions] [FormatPreferences] [ParagraphStyle] [Text] [TextMarker]

Contains an instance of the Language class. The Language class in turn has only one property, which is also called Language. For example, to determine the language type of the current text, you could use the following code:

```
Msgbox .Text.Language.Language
```

{button ,AL('H_CURRENTLANGUAGE_PROPERTY_MEMDEF',0)} [See related topics](#)

Word Pro: LastChild property

{button ,AL(^H_DIVISION_CLASS;H_TEXTDOCUMENT_CLASS',0)} [See list of classes](#)

(Read-only)

Data Type

String

Syntax

lastchildvalue = [objectreference].LastChild

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Word Pro: LastCursorableDivision property

{button ,AL(^H_DIVISION_CLASS;H_TEXTDOCUMENT_CLASS',0)} [See list of classes](#)

(Read-only)

This language element has not yet been defined. You can check for a newer Help file on the Lotus web site.

Data Type

String

Syntax

lastcursorabledivisionvalue = [objectreference].LastCursorableDivision

Legal values

This property is read-only. The value of this property cannot be set by a script.

Usage

Word Pro: LastDivision property

{button ,AL('H_DIVISION_CLASS;H_TEXTDOCUMENT_CLASS',0)} [See list of classes](#)

{button ,AL('H_LASTDIVISION_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-only)

Data Type

String.

Syntax

lastdivisionvalue = [objectreference].LastDivision

Legal values**Usage**

Word Pro: LastDocPath property

{button ,AL(^H_USERINTERFACEPREFS_CLASS,0)} [See list of classes](#)

(Read-write) The path (drive and directory) Word Pro uses if the "Use working directory" option is enabled in Word Pro Preferences.

Data Type

String

Syntax

lastdocpathvalue = [objectreference].LastDocPath

[objectreference].LastDocPath = lastdocpathvalue

Legal values

Usage

Contains the last path that was used for document creation or document save. Only used if the IsWorkingDir property is set up as True.

Word Pro: LastEditorName property

{button ,AL('H_CLICKHERE_CLASS;H_TEXTMARKER_CLASS;H_TEXT_CLASS',0)} [See list of classes](#)

{button ,AL('H_LASTEDITORNAME_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-only)

Data Type

[String](#)

Syntax

lasteditornamevalue = [objectreference].LastEditorName

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Word Pro: LastGraphicPath property

{button ,AL(^H_USERINTERFACEPREFS_CLASS',0)} [See list of classes](#)

(Read-write) The path (drive and directory) that was last chosen when importing a picture.

Data Type

String

Syntax

lastgraphicpathvalue = [objectreference].LastGraphicPath

[objectreference].LastGraphicPath = lastgraphicpathvalue

Legal values

Usage

Equivalent to the path in the Import Picture dialog box. Word Pro always displays the last path chosen to import a picture when it opens the Import Picture dialog box.

Word Pro: LastGraphicType property

{button ,AL('H_USERINTERFACEPREFS_CLASS',0)} [See list of classes](#)

(Read-write) The last type of graphic that Word Pro imported

Data Type

String

Syntax

lastgraphictypevalue = [objectreference].LastGraphicType

[objectreference].LastGraphicType = lastgraphictypevalue

Legal values

Usage

Equivalent to the "Files of type" option in the Import Picture dialog box. Although both the file type and the file extension are listed in this box, this property contains only the file type, not the file extension (for example, GIF or Windows Bitmap, not *.GIF or *.BMP).

Word Pro: LastInsertSymbolFont property

{button ,AL('H_USERINTERFACEPREFS_CLASS',0)} [See list of classes](#)

{button ,AL('H_LASTINSERTSYMBOLFONT_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write)

This language element has not yet been defined. You can check for a newer Help file on the Lotus web site.

Data Type

String

Syntax

lastinsertsymbolfontvalue = [objectreference].LastInsertSymbolFont

[objectreference].LastInsertSymbolFont = lastinsertsymbolfontvalue

Legal values

Any value of type String.

Usage

Word Pro: LastMacroPath property

{button ,AL(^H_USERINTERFACEPREFS_CLASS',0)} [See list of classes](#)

(Read-write) The path (drive and directory) that was last chosen when running a script or setting a startup script.

Data Type

String

Syntax

lastmacropathvalue = [objectreference].LastMacroPath

[objectreference].LastMacroPath = lastmacropathvalue

Legal values

A valid path including drive and directory.

Usage

Equivalent to the path that you use when you choose Edit - Script & Macros, and then either Run - Browse, or Set Startup Scripts - Browse. It is the path that is available in the Lotus Word Pro - Choose Script dialog box.

Word Pro: LastMessageID property

{button ,AL('H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL('H_LASTMESSAGEID_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-only)

This language element has not yet been defined. You can check for a newer Help file on the Lotus web site.

Data Type

Long

Syntax

lastmessageidvalue = [objectreference].LastMessageID

Legal values

This property is read-only. The value of this property cannot be set by a script.

Usage

Word Pro: LastMessageStr property

{button ,AL(^H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL(^H_LASTMESSAGESTR_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-only)

This language element has not yet been defined. You can check for a newer Help file on the Lotus web site.

Data Type

String

Syntax

lastmessagestrvalue = [objectreference].LastMessageStr

Legal values

This property is read-only. The value of this property cannot be set by a script.

Usage

Word Pro: LastMessageType property

{button ,AL(^H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL(^H_LASTMESSAGETYPE_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-only)

This language element has not yet been defined. You can check for a newer Help file on the Lotus web site.

Data Type

Integer

Syntax

lastmessagetypevalue = [objectreference].LastMessageType

Legal values

This property is read-only. The value of this property cannot be set by a script.

Usage

Word Pro: LastName property

{button ,AL('H_DIVISION_CLASS;H_TEXTDOCUMENT_CLASS',0)} [See list of classes](#)

{button ,AL('H_LASTNAME_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-only)

Data Type

[String](#)

Syntax

lastnamevalue = [objectreference].LastName

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Word Pro: LastPage property

{button ,AL('H_DIVISION_CLASS;H_TEXTDOCUMENT_CLASS',0)} [See list of classes](#)

{button ,AL('H_LASTPAGE_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-only)

Data Type

[Integer](#)

Syntax

lastpagevalue = [objectreference].LastPage

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts and Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Word Pro: LastUsedDateFormula property

{button ,AL('H_USERINTERFACEPREFS_CLASS',0)} [See list of classes](#)

(Read-write) The last format of a date that was inserted into a document.

Data Type

String

Syntax

lastuseddateformulavalue = [objectreference].LastUsedDateFormula

[objectreference].LastUsedDateFormula = lastuseddateformulavalue

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Equivalent to the "Date format" option in the Insert Date/Time dialog box.

Word Pro: Last property

{button ,AL(^H_CLICKHERE_CLASS',0)} [See list of classes](#)

(Read-only) The name of the ClickHere block which is last in the division (uses Tab order.)

Data Type

String

Syntax

lastvalue = [objectreference].Last

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Word Pro: LayerName property

```
{button ,AL('H_LAYOUT_CLASS;H_CELLGROUPLAYOUT_CLASS;H_CELLLAYOUT_CLASS;H_CONNECTEDLAY  
OUT_CLASS;H_COLUMNGROUPLAYOUT_CLASS;H_FOOTERLAYOUT_CLASS;H_FRAMELAYOUT_CLASS;H  
_FRAMEGROUPLAYOUT_CLASS;H_DROP_CAPLAYOUT_CLASS;H_GROUPLAYOUT_CLASS;H_HEADERLAY  
OUT_CLASS;H_NOTELAYOUT_CLASS;H_PAGELAYOUT_CLASS;H_ROWGROUPLAYOUT_CLASS;H_ROWLA  
YOUT_CLASS;H_RUBYLAYOUT_CLASS;H_SUPERTABLEGROUPLAYOUT_CLASS;H_SUPERTABLELAYOUT  
_CLASS;H_TABLELAYOUT_CLASS;H_ENDNOTELAYOUT_CLASS;H_FOOTNOTELAYOUT_CLASS;H_TABLE  
HEADINGLAYOUT_CLASS;H_TOCSUPERTABLELAYOUT_CLASS',0)} See list of classes
```

```
{button ,AL('H_LAYERNAME_PROPERTY_EXSCRIPT',1)} See example
```

(Read-only) Indicates the name of the Layer object for a specified layout object.

Data Type

String

Syntax

layernamevalue = [objectreference].LayerName

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

The LayerName property provides you with the name of the Layer, or Watermark, object for a specified layout object. If a layer object does not exist in the specified layout object, its LayerName property contains an empty string.

Word Pro: Layer property

```
{button ,AL('H_LAYOUT_CLASS;H_CELLGROUPLAYOUT_CLASS;H_CELLAYOUT_CLASS;H_CONNECTEDLAY  
OUT_CLASS;H_COLUMNGROUPLAYOUT_CLASS;H_FOOTERLAYOUT_CLASS;H_FRAMELAYOUT_CLASS;H  
_FRAMEGROUPLAYOUT_CLASS;H_DROPDOWNLAYOUT_CLASS;H_GROUPLAYOUT_CLASS;H_HEADERLAY  
OUT_CLASS;H_NOTELAYOUT_CLASS;H_PAGELAYOUT_CLASS;H_ROWGROUPLAYOUT_CLASS;H_ROWLA  
YOUT_CLASS;H_RUBYLAYOUT_CLASS;H_SUPERTABLEGROUPLAYOUT_CLASS;H_SUPERTABLELAYOUT  
_CLASS;H_TABLELAYOUT_CLASS;H_ENDNOTELAYOUT_CLASS;H_FOOTNOTELAYOUT_CLASS;H_TABLE  
HEADINGLAYOUT_CLASS;H_TOCSUPERTABLELAYOUT_CLASS',0)} See list of classes
```

```
{button ,AL('H_LAYER_PROPERTY_EXSCRIPT',1)} See example
```

(Read-only) Provides access to the Layer object for a specific layout object.

Data Type

[FrameLayout](#)

Syntax

layervalue = [objectreference].Layer

Legal values

Always contains an instance of the FrameLayout class.

Usage

The Layer property provides access to a layout object's Layer, or Watermark, object.

Word Pro: LayoutName property

{button ,AL(^H_DIVISIONINFO_CLASS;H_GRAPHICOLEBJECT_CLASS;H_GRAPHIC_CLASS;H_OLEOBJECT_CLASS;H_CLICKHERE_CLASS;H_TEXTMARKER_CLASS;H_SECTION_CLASS;H_INDEXSECTION_CLASS;H_TEXT_CLASS',0)} [See list of classes](#)

(Read-only) The content class of the layout that is associated with the text object.

Data Type

String

Syntax

layoutnamevalue = [objectreference].LayoutName

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Word Pro: LayoutOverride property

{button ,AL(^H_DIVISIONINFO_CLASS',0)} [See list of classes](#)

(Read-only)

Data Type

[LayoutOverride](#)

Syntax

layoutoverridevalue = [objectreference].LayoutOverride

Legal values

Always contains an instance of the LayoutOverride class.

Usage

Word Pro: Layouts property

{button ,AL('H_FOUNDRY_CLASS',0)} [See list of classes](#)

(Read-only) An object created from the LayoutCollection class. This object provides access to all types of Layout objects. Layout objects are those objects created from one of the derived Layout classes listed below:

- CellLayout
- ConnectedLayout
- EndnoteLayout
- FooterLayout
- FootnoteLayout
- FrameLayout
- GroupLayout
- HeaderLayout
- NoteLayout
- PageLayout
- RowLayout
- RubyLayout
- SuperTableLayout
- TableHeadingLayout
- TableLayout

All of these classes are derived from the same Layout class and share the common set of Layout class members. The objects created from these classes are all related through their common parent class, Layout. That is why we say they are Layout objects. One of the benefits of related classes of objects is the ability to store related objects in a variable with the parent class data type.

For example, a variable of type CellLayout could only hold an object created from the CellLayout class. But a variable of type Layout could hold any object created from one of the Layout-derived classes listed above.

When you use the LayoutCollection object in this property, you have access to all types of Layout objects.

Data Type

[LayoutCollection](#)

Syntax

layoutsvalue = [objectreference].Layouts

Legal values

Always contains an instance of the LayoutCollection class.

Usage

When accessed through the Foundry property on a Division object, the collection object in this property provides access to all the Layout objects contained in that Division object.

When accessed through the AppFoundry property on the WPAApplication object, this collection object provides access to all the Layout objects contained in the Word Pro Clipboard.

When accessed through the TempFoundry property on the WPAApplication object, this collection object provides access to all the Layout objects placed in TempFoundry by WordPro or a script.

When accessed through the Foundry property on the WPAApplication object, this collection object provides access to all the Layout objects contained in the currently active Division object.

This property is not used in the Foundry property on the TextDocument object.

For more information about collection classes, see [Overview: Word Pro LotusScript Collection Classes](#).

Word Pro: Layout property

```
{button ,AL(^H_WPAPPLICATION_CLASS;H_BASECONTAINER_CLASS;H_CELLCONTAINER_CLASS;H_FRAME  
CONTAINER_CLASS;H_NOTECONTAINER_CLASS;H_RUBYCONTAINER_CLASS;H_DROPDOWNCONTAINER_  
CLASS;H_PAGECONTAINER_CLASS;H_SUBPAGECONTAINER_CLASS;H_SUPERPAGECONTAINER_CLASS  
;H_ROWCONTAINER_CLASS;H_SUPERTABLECONTAINER_CLASS;H_TABLECONTAINER_CLASS;H_PARAL  
LELCOLSCONTAINER_CLASS;H_TABLEONLYCONT_CLASS;H_MARKER_CLASS;H_POWERFIELD_CLASS;  
H_CLICKHERE_CLASS;H_TEXTMARKER_CLASS;H_RUBYMARKER_CLASS;H_TABLEMARKER_CLASS;H_B  
ASETABLE_CLASS;H_FOOTNOTETABLE_CLASS;H_PARALLELCOLUMNS_CLASS;H_GLOSSARY_CLASS;H  
_TABLEHEADING_CLASS;H_TABLE_CLASS',0)} See list of classes
```

(Read-only) Returns the layout of any object.

Data Type

Layout

Syntax

layoutvalue = [objectreference].Layout

Legal values

Always contains an instance of the Layout class or one of its derived classes.

Usage

When called from WPAApplication, this property contains the Layout object for the container object that is uppermost in the focus. When called from a container class, this property contains a layout object that corresponds to the container object's type. For example, the layout property of a frame container object contains a frame layout object; the layout property of a cell container object contains a cell layout object.

Word Pro: LeaderDotDashChar property

{button ,AL(^H_CHARACTERSET_CLASS',0)} [See list of classes](#)

(Read-only)

Data Type

Integer

Syntax

leaderdotdashcharvalue = [objectreference].LeaderDotDashChar

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Word Pro: LeaderDotDotChar property

{button ,AL(^H_CHARACTERSET_CLASS',0)} [See list of classes](#)

(Read-only)

Data Type

[Integer](#)

Syntax

leaderdotdotcharvalue = [objectreference].LeaderDotDotChar

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Word Pro: LeaderDotType property

```
{button ,AL(^H_LAYOUT_CLASS;H_CELLGROUPLAYOUT_CLASS;H_CELLLAYOUT_CLASS;H_CONNECTEDLAY  
OUT_CLASS;H_COLUMNGROUPLAYOUT_CLASS;H_FOOTERLAYOUT_CLASS;H_FRAMELAYOUT_CLASS;H  
_FRAMEGROUPLAYOUT_CLASS;H_DROP_CAPLAYOUT_CLASS;H_GROUPLAYOUT_CLASS;H_HEADERLAY  
OUT_CLASS;H_NOTELAYOUT_CLASS;H_PAGELAYOUT_CLASS;H_ROWGROUPLAYOUT_CLASS;H_ROWLA  
YOUT_CLASS;H_RUBYLAYOUT_CLASS;H_SUPERTABLEGROUPLAYOUT_CLASS;H_SUPERTABLELAYOUT  
_CLASS;H_TABLELAYOUT_CLASS;H_ENDNOTELAYOUT_CLASS;H_FOOTNOTELAYOUT_CLASS;H_TABLE  
HEADINGLAYOUT_CLASS;H_TOCSUPERTABLELAYOUT_CLASS;H_USERINTERFACEPREFS_CLASS',0)}
```

[See list of classes](#)

[Layout]

(Read-write) The type of dot used as a leader in a layout object.

[UserInterfacePrefs]

(Read-write) The type of leader dot to be used when a tab is inserted.

Data Type

Data type is [Variant](#), which allows the value of this property to be one of the constants listed below or its numeric equivalent (in parentheses).

Syntax

leaderdotypevalue = [objectreference].LeaderDotType

[objectreference].LeaderDotType = leaderdotypevalue

Legal values

<u>Value</u>	<u>Effect</u>
\$LwpLeaderDotTypeNone (526)	Prevents leader dots from being inserted in a layout object.
\$LwpLeaderDotTypeDots (527)	Inserts dots as the type of leader dot used in a layout object.
\$LwpLeaderDotTypeDashes (528)	Inserts dashes as the type of leader dot used in a layout object.
\$LwpLeaderDotTypeUnderscores (529)	Inserts underscores as the type of leader dot used in a layout object.

Usage

You can set the value of this property to affect individual layout objects, or you can set it to act as a default for all layout objects.

To set the type of leader dot in a particular layout object, set the value of this property for that layout object. To set default preferences for a type of leader dot used throughout Word Pro, set the value of this property on the UserInterfacePrefs object. Equivalent to choosing the "Leader" box in the Set Tabs dialog box.

Word Pro: LeaderDotUnderscoreChar property

{button ,AL(^H_CHARACTERSET_CLASS',0)} [See list of classes](#)

(Read-only)

Data Type

Integer

Syntax

leaderdotunderscorecharvalue = [objectreference].LeaderDotUnderscoreChar

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Word Pro: LeadingText property

{button ,AL(`H_FOOTNOTENUMOPT_CLASS;H_ENDNOTEDIVISIONGROUPNUM_CLASS;H_ENDNOTEDIVISIONNUM_CLASS;H_ENDNOTEDOCNUM_CLASS;H_FOOTNOTENUMBERING_CLASS',0)} [See list of classes](#)

{button ,AL(`H_LEADINGTEXT_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Adds a string of characters before a footnote or endnote reference.

Data Type

[String](#)

Syntax

leadingtextvalue = [objectreference].LeadingText

[objectreference].LeadingText = leadingtextvalue

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Equivalent to choosing Create - Footnote/Endnote, clicking Options, selecting "Enclosed by text before," and inserting text in the "Text before" box located on the Numbering panel.

{button ,AL(`H_TRAILINGTEXT_PROPERTY_MEMDEF',0)} [See related topics](#)

Word Pro: LeftBorder property

{button ,AL(^H_BORDERLINES_CLASS;H_GUTTER_CLASS;H_BETWEENLINES_CLASS',0)} [See list of classes](#)

(Read-only) Allows you to access an object's left border object.

Data Type

[Border](#)

Syntax

leftbordervalue = [objectreference].LeftBorder

Legal values

Always contains an instance of the Border class.

Usage

You can also use the AllBorders property to simultaneously access an object's BottomBorder, LeftBorder, RightBorder, and TopBorder objects.

Word Pro: LeftExternalMargin property

```
{button ,AL('H_LAYOUT_CLASS;H_CELLGROUPLAYOUT_CLASS;H_CELLLAYOUT_CLASS;H_CONNECTEDLAY  
OUT_CLASS;H_COLUMNGROUPLAYOUT_CLASS;H_FOOTERLAYOUT_CLASS;H_FRAMELAYOUT_CLASS;H  
_FRAMEGROUPLAYOUT_CLASS;H_DROP_CAPLAYOUT_CLASS;H_GROUPLAYOUT_CLASS;H_HEADERLAY  
OUT_CLASS;H_NOTELAYOUT_CLASS;H_PAGELAYOUT_CLASS;H_ROWGROUPLAYOUT_CLASS;H_ROWLA  
YOUT_CLASS;H_RUBYLAYOUT_CLASS;H_SUPERTABLEGROUPLAYOUT_CLASS;H_SUPERTABLELAYOUT  
_CLASS;H_TABLELAYOUT_CLASS;H_ENDNOTELAYOUT_CLASS;H_FOOTNOTELAYOUT_CLASS;H_TABLE  
HEADINGLAYOUT_CLASS;H_TOCSUPERTABLELAYOUT_CLASS',0)} See list of classes
```

```
{button ,AL('H_LEFTEXTERNALMARGIN_PROPERTY_EXSCRIPT',1)} See example
```

(Read-write) Allows you to set the width of the left external margin of a layout object.

Data Type

Long

Syntax

leftexternalmarginvalue = [objectreference].LeftExternalMargin

[objectreference].LeftExternalMargin = leftexternalmarginvalue

Legal values

Data type is Long but the unit of measurement used for this property is Twips. There are 1440 Twips per inch.

Usage

This property cannot be set individually for FrameLayout objects within Word Pro. It is combined with all external margin values in the "Padding around border" setting on the Size & Margins panel of the InfoBox.

Word Pro: LeftPage property

```
{button ,AL('H_LAYOUT_CLASS;H_CELLGROUPLAYOUT_CLASS;H_CELLLAYOUT_CLASS;H_CONNECTEDLAY  
OUT_CLASS;H_COLUMNGROUPLAYOUT_CLASS;H_FOOTERLAYOUT_CLASS;H_FRAMELAYOUT_CLASS;H  
_FRAMEGROUPLAYOUT_CLASS;H_DROP_CAPLAYOUT_CLASS;H_GROUPLAYOUT_CLASS;H_HEADERLAY  
OUT_CLASS;H_NOTELAYOUT_CLASS;H_PAGELAYOUT_CLASS;H_ROWGROUPLAYOUT_CLASS;H_ROWLA  
YOUT_CLASS;H_RUBYLAYOUT_CLASS;H_SUPERTABLEGROUPLAYOUT_CLASS;H_SUPERTABLELAYOUT  
_CLASS;H_TABLELAYOUT_CLASS;H_ENDNOTELAYOUT_CLASS;H_FOOTNOTELAYOUT_CLASS;H_TABLE  
HEADINGLAYOUT_CLASS;H_TOCSUPERTABLELAYOUT_CLASS',0)} See list of classes
```

(Read-only) The left page of a complex layout. A complex layout contains separate layouts for left and right pages.

Data Type

[PageLayout](#)

Syntax

leftpagevalue = [objectreference].LeftPage

Legal values

Always contains an instance of the PageLayout class.

Usage

Word Pro: LeftTopCellId property

{button ,AL(^H_LAYOUT_CLASS;H_CELLGROUPLAYOUT_CLASS;H_CELLLAYOUT_CLASS;H_CONNECTEDLAY
OUT_CLASS;H_COLUMNGROUPLAYOUT_CLASS;H_FOOTERLAYOUT_CLASS;H_FRAMELAYOUT_CLASS;H
_FRAMEGROUPLAYOUT_CLASS;H_DROP_CAPLAYOUT_CLASS;H_GROUPLAYOUT_CLASS;H_HEADERLAY
OUT_CLASS;H_NOTELAYOUT_CLASS;H_PAGELAYOUT_CLASS;H_ROWGROUPLAYOUT_CLASS;H_ROWLA
YOUT_CLASS;H_RUBYLAYOUT_CLASS;H_SUPERTABLEGROUPLAYOUT_CLASS;H_SUPERTABLELAYOUT
_CLASS;H_TABLELAYOUT_CLASS;H_ENDNOTELAYOUT_CLASS;H_FOOTNOTELAYOUT_CLASS;H_TABLE
HEADINGLAYOUT_CLASS;H_TOCSUPERTABLELAYOUT_CLASS',0)} [See list of classes](#)

(Read-only) Indicates the column ID of the top left cell when cells are connected in a layout object. In an unconnected cell, this property returns the current cell's column ID.

Data Type

[Integer](#)

Syntax

lefttopcellidvalue = [objectreference].LeftTopCellId

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

The column ID that is returned is zero based, which means that the first column in a table has a column ID value of zero.

Word Pro: Left property

{button ,AL(^H_WINDOW_CLASS;H_APPLICATIONWINDOW_CLASS;H_STATUSBAR_CLASS;H_DOCWINDOW_CLASS',0)} [See list of classes](#)

(Read-write)

[StatusBar]

The left position of the status bar on the workspace.

[ApplicationWindow]

The position of the window (the origin of the window) relative to the left of the screen.

[Window]

[DocWindow]

Data Type

Long

Syntax

leftvalue = [objectreference].Left

[objectreference].Left = leftvalue

Legal values

Data type is Long but the unit of measurement used for this property is Twips. There are 1440 Twips per inch.

Usage

[StatusBar]

Use this property to set the left position of the status bar on the workspace.

[ApplicationWindow]

Use this property to set the left position of the application window on the workspace.

[Window]

[DocWindow]

Word Pro: Length property

{button ,AL(^H_BAG_CLASS',0)} [See list of classes](#)

(Read-only) The number of bytes of data stored in the bag.

Data Type

Long

Syntax

lengthvalue = [objectreference].Length

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Word Pro: Level property

{button ,AL(^H_KINSOKU_CLASS;H_NUMBERING_CLASS',0)} [See list of classes](#)

(Read-write)

Data Type

Integer

Syntax

levelvalue = [objectreference].Level

[objectreference].Level = [objectreference].Level

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Word Pro: LineLocation property

```
{button ,AL('H_LAYOUT_CLASS;H_CELLGROUPLAYOUT_CLASS;H_CELLLAYOUT_CLASS;H_CONNECTEDLAYOUT_CLASS;H_COLUMNGROUPLAYOUT_CLASS;H_FOOTERLAYOUT_CLASS;H_FRAMELAYOUT_CLASS;H_FRAMEGROUPLAYOUT_CLASS;H_DROP_CAP_LAYOUT_CLASS;H_GROUPLAYOUT_CLASS;H_HEADERLAYOUT_CLASS;H_NOTELAYOUT_CLASS;H_PAGELAYOUT_CLASS;H_ROWGROUPLAYOUT_CLASS;H_ROW_LAYOUT_CLASS;H_RUBY_LAYOUT_CLASS;H_SUPERTABLEGROUPLAYOUT_CLASS;H_SUPERTABLELAYOUT_CLASS;H_TABLELAYOUT_CLASS;H_ENDNOTELAYOUT_CLASS;H_FOOTNOTELAYOUT_CLASS;H_TABLEHEADINGLAYOUT_CLASS;H_TOCSUPERTABLELAYOUT_CLASS',0)} See list of classes
```

```
{button ,AL('H_LINELOCATION_PROPERTY_EXSCRIPT',1)} See example
```

(Read-write) Allows you to determine the placement of a line around or within a layout object.

Data Type

[Integer](#)

Syntax

linelocationvalue = [objectreference].LineLocation

[objectreference].LineLocation = linelocationvalue

Legal values

<u>Value</u>	<u>Effect</u>
0	Places a line around the margins of a layout object.
1	Places a line in the middle of a layout object.
2	Places a line around the edge of a layout object.
3	Specifies the distance of a line from the edge of a layout object.

Usage

Equivalent to the "Line placement" option on the Lines & Colors panel of the InfoBox for certain layout objects.

```
{button ,AL('H_BORDEROFFSET_PROPERTY_MEMDEF',0)} See related topics
```

Word Pro: LineMix property

{button ,AL(^H_TABLELINE_CLASS,0)} [See list of classes](#)

(Read-Write) Allows you to assign mixed table line settings to a table cell.

Data Type

Data type is [Variant](#), which allows the value of this property to be one of the constants listed below or its numeric equivalent (in parentheses).

Syntax

linemix value = [objectreference].LineMix

[objectreference].LineMix = Linemix value

Legal values

<u>Value</u>	<u>Effect</u>
\$LwpTableMixAllmixed (1893)	Assigns a different line style to each line of a table cell.
\$LwpTableMixBottommixed (1890)	Assigns a line style to the bottom line of a table cell that is different from the line style of the top, right, and left line of a table cell.
\$LwpTableMixLeftmixed (1891)	Assigns a line style to the left line of a table cell that is different from the line style of the top, right, and bottom line of a table cell.
\$LwpTableMixRightmixed (1892)	Assigns a line style to the right line of a table cell that is different from the line style of the top, bottom, and left line of a table cell.
\$LwpTableMixTopmixed (1889)	Assigns a line style to the top line of a table cell that is different from the line style of the bottom, right, and left line of a table cell.

Usage

Sets more than one type of Word Pro defined line styles in a table cell and returns different types of line styles assigned to a table cell.

For example, you can set the top and bottom lines of a table cell with a double wavy pattern, and the right and left lines of the cell with a long dash pattern. You can also assign a different line style to each line in a table cell, such as the top line with a wavy pattern, the bottom line with a star pattern, the right line with a long dash pattern, and the left line with a dotted pattern.

Word Pro: LineNumberOptions property

{button ,AL('H_DIVISION_CLASS;H_TEXTDOCUMENT_CLASS',0)} [See list of classes](#)

{button ,AL('H_LINENUMBEROPTIONS_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-only)

Data Type

[LineNumberOptions](#)

Syntax

linenumbersvalue = [objectreference].LineNumberOptions

Legal values

Always contains an instance of the LineNumberOptions class.

Usage

Word Pro: LinePlacement property

{button ,AL(^H_BORDERLINES_CLASS;H_GUTTER_CLASS;H_BETWEENLINES_CLASS',0)} [See list of classes](#)

{button ,AL(^H_LINEPLACEMENT_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Sets or returns the position at which lines appear around an object.

Data Type

Data type is [Variant](#), which allows the value of this property to be one of the constants listed below or its numeric equivalent (in parentheses).

Syntax

lineplacementvalue = [objectreference].LinePlacement

[objectreference].LinePlacement = lineplacementvalue

Legal values

<u>Value</u>	<u>Effect</u>
LwpLinePlacementAllsides (&HF)	Lines should appear on all sides of the object.
LwpLinePlacementBottom (&H8)	Lines should appear at the bottom of the object.
LwpLinePlacementLeft (&H1)	Lines should appear on the left side of the object.
LwpLinePlacementRight (&H2)	Lines should appear on the right side of the object.
LwpLinePlacementTop (&H4)	Lines should appear at the top of the object.

Usage

The values for LinePlacement can be used individually or combined together. For example, you can combine LwpLinePlacementLeft and LwpLinePlacementRight so that lines appear on the left side and the right side of an object. For more information on using these values, called bitmasks, see the Bitmasks section of the topic [Overview: Word Pro LotusScript Enumerated Values](#).

Word Pro: LinesSpacedEveryNthUnit property

{button ,AL('H_LINENUMBEROPTIONS_CLASS',0)} [See list of classes](#)

{button ,AL('H_LINESPACEDEVERYNTHUNIT_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Determines the amount of space between line numbers.

Data Type

Long

Syntax

linesspacedeverynthunitvalue = [objectreference].LinesSpacedEveryNthUnit

[objectreference].LinesSpacedEveryNthUnit = linesspacedeverynthunitvalue

Legal values

Data type is Long but the unit of measurement used for this property is Twips. There are 1440 Twips per inch.

Usage

When the NumberWhichLines property is set to \$LwpLineNumberOptsSpecifiedLines, you can use this property to determine the amount of space between line numbers.

{button ,AL('H_NUMBЕРЕVERYNTHLINE_PROPERTY_MEMDEF',0)} [See related topics](#)

Word Pro: Lines property

{button ,AL('H_FRAMECAPTIONOPTIONS_CLASS',0)} [See list of classes](#)

{button ,AL('H_LINES_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Represents one of the seven options for lines around a frame caption.

Data Type

Integer.

Syntax

linesvalue = [objectreference].Lines

[objectreference].Lines = linesvalue

Legal values

Integer. The legal values are 0 -6, which represents the seven "Line around caption" buttons in the Create Frame Caption dialog box.

Usage

Equivalent to selecting a frame in the document, choosing Frame - New Caption, clicking the Layout tab, and clicking one of the "Line around caption" buttons.

Word Pro: LineValid property

{button ,AL(^H_BORDERLINES_CLASS;H_GUTTER_CLASS;H_BETWEENLINES_CLASS',0)} [See list of classes](#)

(Read-write) Indicates whether or not a line is valid for a specific object.

Data Type

Data type is [Variant](#), which allows the value of this property to be one of the constants listed below or its numeric equivalent (in parentheses).

Syntax

linevalidvalue = [objectreference].LineValid

[objectreference].LineValid = linevalidvalue

Legal values

<u>Value</u>	<u>Effect</u>
LwpLinePlacementAllsides (&HF)	All lines around the object are valid.
LwpLinePlacementBottom (&H8)	The line at the bottom of the object is valid.
LwpLinePlacementLeft (&H1)	The line on the left side of the object is valid.
LwpLinePlacementRight (&H2)	The line on the right side of the object is valid.
LwpLinePlacementTop (&H4)	The line at the top of the object is valid.

Usage

Word Pro: LinkAvailable property

{button ,AL(^H_GRAPHIC_CLASS;H_OLEOBJECT_CLASS',0)} [See list of classes](#)

(Read-only)

Note This property is not supported for the class OLEObject within OS/2.

Data Type

Integer

Syntax

linkavailablevalue = [objectreference].LinkAvailable

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Word Pro: LinkDisplayNameFileLength property

{button ,AL(^H_OLEOBJECT_CLASS',0)} [See list of classes](#)

(Read-only)

Note This property is not supported for the class OLEObject within OS/2.

Data Type

Long

Syntax

linkdisplaynamefilelengthvalue = [objectreference].LinkDisplayNameFileLength

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Word Pro: LinkDisplayName property

{button ,AL('H_GRAPHICOLEBJECT_CLASS;H_GRAPHIC_CLASS;H_OLEOBJECT_CLASS',0)} [See list of classes](#)

(Read-only)

Note This property is not supported for the class OLEObject within OS/2.

Data Type

[String](#)

Syntax

linkdisplaynamevalue = [objectreference].LinkDisplayName

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Word Pro: LinkedFileName property

{button ,AL(^H_GRAPHICOLEBJECT_CLASS;H_GRAPHIC_CLASS;H_OLEOBJECT_CLASS',0)} [See list of classes](#)

(Read-only)

Data Type

[String](#)

Syntax

linkedfilenamevalue = [objectreference].LinkedFileName

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Word Pro: Linked property

{button ,AL(^H_OLEOBJECT_CLASS',0)} [See list of classes](#)

(Read-only)

Data Type

Integer

Syntax

linkedvalue = [objectreference].Linked

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Word Pro: LinkFrame property

```
{button ,AL(^H_LAYOUT_CLASS;H_CELLGROUPLAYOUT_CLASS;H_CELLLAYOUT_CLASS;H_CONNECTEDLAY  
OUT_CLASS;H_COLUMNGROUPLAYOUT_CLASS;H_FOOTERLAYOUT_CLASS;H_FRAMELAYOUT_CLASS;H  
_FRAMEGROUPLAYOUT_CLASS;H_DROP_CAPLAYOUT_CLASS;H_GROUPLAYOUT_CLASS;H_HEADERLAY  
OUT_CLASS;H_NOTELAYOUT_CLASS;H_PAGELAYOUT_CLASS;H_ROWGROUPLAYOUT_CLASS;H_ROWLA  
YOUT_CLASS;H_RUBYLAYOUT_CLASS;H_SUPERTABLEGROUPLAYOUT_CLASS;H_SUPERTABLELAYOUT  
_CLASS;H_TABLELAYOUT_CLASS;H_ENDNOTELAYOUT_CLASS;H_FOOTNOTELAYOUT_CLASS;H_TABLE  
HEADINGLAYOUT_CLASS;H_TOCSUPERTABLELAYOUT_CLASS',0)} See list of classes
```

(Read-write) Determines whether or not a frame layout object will link its contents with another frame layout object.

Data Type

String

Syntax

linkframevalue = [objectreference].LinkFrame

[objectreference].LinkFrame = linkframevalue

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Equivalent to the "Link frame contents to" setting on the Misc panel of the InfoBox for frame layout objects. The LinkFrame property is used only with frame layout objects.

In order to remove the link from a frame layout object, set the LinkFrame property to an empty string value. Setting a layout object's LinkFrame property to a value that does not correspond with an appropriate frame layout object also results in an empty string value being assigned to the LinkFrame property.

Word Pro: LinkGraphic property

{button ,AL(^H_PREFERENCES_CLASS',0)} [See list of classes](#)

(Read-write)

Data Type

Integer

Syntax

linkgraphicvalue = [objectreference].LinkGraphic

[objectreference].LinkGraphic = linkgraphicvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Word Pro: ListCount property

{button ,AL(^H_SMARTFILL_CLASS',0)} [See list of classes](#)

(Read-only)

Data Type

Integer

Syntax

listcountvalue = [objectreference].ListCount

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Word Pro: LoadFilesMaximized property

{button ,AL('H_USERINTERFACEPREFS_CLASS',0)} [See list of classes](#)

{button ,AL('H_LOADFILESMAXIMIZED_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Indicates whether documents cover the entire application workspace when they are created or opened.

Data Type

[Integer \(Bool\)](#)

Syntax

loadfilesmaximizedvalue = [objectreference].LoadFilesMaximized

[objectreference].LoadFilesMaximized = loadfilesmaximizedvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values. Default is True (-1).

Usage

Equivalent to the "Load files maximized" option on the General panel of the Word Pro Preferences dialog box. If this property is set to True (-1), Word Pro loads files maximized. If set to False (0), Word Pro does not load files maximized.

Location property

{button ,AL(^H_APPLICATION_CLASS;H_WPAPPLICATION_CLASS;H_DOCUMENT_CLASS;H_TEXTDOCUMENT_CLASS',0)} [See list of classes](#)

[Application]

(Read-only) Returns the path to the Word Pro executable.

[Document]

(Read-only) Returns the path to the directory in which the document is stored.

Data type

String

Syntax

[objectreference].Location = locationvalue

locationvalue = [objectreference].Location

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Word Pro: Locked property

{button ,AL('H_DIVISION_CLASS;H_TEXTDOCUMENT_CLASS',0)} [See list of classes](#)

(Read-write)

Data Type

Integer

Syntax

lockedvalue = [objectreference].Locked

[objectreference].Locked = lockedvalue

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Word Pro: LockLevel property

{button ,AL(^H_DIVISION_CLASS;H_TEXTDOCUMENT_CLASS',0)} [See list of classes](#)

(Read-only)

Annotations

Data Type

Integer

Syntax

locklevelvalue = [objectreference].LockLevel

Legal values

This property is read-only. The value of this property cannot be set by a script.

Usage

Normal (0)

Locked for annotation (1)

ReadOnly (2)

Word Pro: LockResult property

{button ,AL(^H_POWERFIELD_CLASS',0)} [See list of classes](#)

(Read-write)

Data Type

Integer.

Syntax

[objectreference].LockResult = lockresultvalue

lockresultvalue = [objectreference].LockResult

Legal values**Usage**

Word Pro: Locks property

{button ,AL('H_EDITOR_CLASS',0)} [See list of classes](#)

(Read-only) Allows or restricts the tasks an editor can perform in a document.

Data Type

Data type is [Variant](#), which allows the value of this property to be one of the constants listed below or its hexadecimal equivalent (in parentheses). You can combine these constants when you want Word Pro to combine the features listed below. Use the OR operator to combine constants.

Syntax

locksvalue = [objectreference].Locks

Legal values

<u>Value</u>	<u>Effect</u>
LwpEditLocksNoCopyAndNoSaveas (&H4)	Prevents an editor from copying any part of the document to the Clipboard, using drag and drop on any part of the document, or using File - Save As to save the document with a new name.
LwpEditLocksNoEditNamedStyles (&H2)	Prevents an editor from modifying any named styles in the document.
LwpEditLocksNoLocks (&H0)	Allows an editor to make any changes to a document.
LwpEditLocksNoPrinting (&H8)	Prevents an editor from printing the document.
LwpEditLocksNoVersionOrReview (&H1)	Prevents an editor from creating new versions or seeing any other versions.
LwpEditLocksRevmarkOnly (&H10)	Allows an editor to edit the document. However, all edits appear as markups. The editor cannot accept or reject edits using the Review bar. For example, when the editor deletes text, it does not disappear from the document, but appears with the editor's deleted text attributes.

Usage

You can use one or a combination of the values above to determine which tasks an editor can perform. For more information on combining these values, called bitmasks, see the topic [Overview: Word Pro LotusScript Enumerated Values](#).

Word Pro: LotusMiscStatus property

{button ,AL('H_OLEOBJECT_CLASS',0)} [See list of classes](#)

(Read-only)

This language element has not yet been defined. Lotus provides updated Help files on its web site.

Data Type

The data type for this property is Variant.

Syntax

lotusmiscstatusvalue = [objectreference].LotusMiscStatus

Legal values

This property is read-only. The value of this property cannot be set by a script. The possible values for this property are listed below. Word Pro can combine two or more values in this property.

- LwpLmsDrawsupportsrotation (&H80)
- LwpLmsInplaceactivateinpreview (&H2)
- LwpLmsJustcontextmenus (&H1)
- LwpLmsLoadfromappdirectory (&H10)
- LwpLmsLoadfromshreddirectory (&H8)
- LwpLmsOptimizedcreatefromfile (&H20)
- LwpLmsRoutekeytocontainerDel (&H100)
- LwpLmsRoutekeytocontainerEsc (&H400)
- LwpLmsRoutekeytocontainerF1 (&H200)
- LwpLmsRoutemsgtocontainerRightmouse (&H800)
- LwpLmsSnapshotactiveview (&H40)
- LwpLmsSupportsimageprocessing (&H2000)
- LwpLmsSupportssrcandtransparency (&H1000)
- LwpLmsUiactivateinpreview (&H4)

Usage

Word Pro: LowerCaseAscent property

{button ,AL(^H_FONT_CLASS',0)} [See list of classes](#)

(Read-only) Indicates the ascent, or height, of lower case letters in the current font object.

Data Type

Single

Syntax

lowercaseascentvalue = [objectreference].LowerCaseAscent

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Word Pro: LowerCase property

{button ,AL('H_FONT_CLASS',0)} [See list of classes](#)

(Read-write)

Data Type

Integer

Syntax

lowercasevalue = [objectreference].LowerCase

[objectreference].LowerCase = lowercasevalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Word Pro: LwpMenuBar property

{button ,AL('H_APPLICATIONWINDOW_CLASS',0)} [See list of classes](#)

(Read-only) The main Word Pro application menu bar object, created from the MenuItem class.

Data Type

[MenuItem](#)

Syntax

lwpmenubarvalue = [objectreference].LwpMenuBar

Legal values

Always contains an instance of the MenuItem class.

Usage

The MenuItem class is used to set and get your own menu items. You can get the current values of LWP menu items, but you cannot change them. To change LWPMenultems, you must create your own menu item and replace the LWPMenultem with the new menu items.

Word Pro: MacroName property

{button ,AL('H_MACRO_CLASS',0)} [See list of classes](#)

(Read-write)

Data Type

String

Syntax

macronamevalue = [objectreference].MacroName

[objectreference].MacroName = macronamevalue

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Word Pro: MacroPaths property

{button ,AL('H_USERINTERFACEPREFS_CLASS',0)} [See list of classes](#)

(Read-only) Stores multiple paths (drives and directories) for Word Pro scripts.

Data Type

[StringCollection](#)

Syntax

macropathvalue = [objectreference].MacroPaths

Legal values

Always contains an instance of the StringCollection class.

Usage

Equivalent to the "Scripts" field on the Locations panel of the Word Pro Preferences dialog box. This field can contain multiple paths. This property returns a collection of String objects which contain the names of all default Word Pro script paths.

Word Pro: MacroPath property

{button ,AL(^H_USERINTERFACEPREFS_CLASS',0)} [See list of classes](#)

(Read-write) Stores the default path (drive and directory) for Word Pro scripts.

Data Type

String

Syntax

macropathvalue = [objectreference].MacroPath

[objectreference].MacroPath = macropathvalue

Legal values

A valid path including drive and directory.

Usage

Equivalent to the "Scripts" field on the Locations panel of the Word Pro Preferences dialog box. In Word Pro, this field can contain multiple paths. This property contains the first path listed in the "Scripts" field.

Word Pro: MacroStatus property

{button ,AL('H_MACRO_CLASS',0)} [See list of classes](#)

(Read-only)

Data Type

Integer (Enumerated Bitmask)

MacroStatus

Syntax

macrostatusvalue = [objectreference].MacroStatus

Legal values

LwpMacroStatusIsPaused (&H1)

LwpMacroStatusIsQuickRecord (&H4)

LwpMacroStatusIsRecording (&H2)

LwpMacroStatusIsRunning (&H1)

Usage

Word Pro: Macro property

{button ,AL('H_APPLICATIONWINDOW_CLASS',0)} [See list of classes](#)

(Read-only) An instance of the macro class for the current application window.

Data Type

Macro

Syntax

macrovalue = [objectreference].Macro

Legal values

Always contains an instance of the Macro class.

Usage

Runs scripts and/or macros saved in another file, or runs scripts and/or macros when you do not have any open documents.

Word Pro: MailRoutingPtr property

{button ,AL(^H_MAILROUTING_CLASS',0)} [See list of classes](#)

(Read-only) An internal processor that points to the document's internal data structure.

Data Type

Long

Syntax

mailroutingptrvalue = [objectreference].MailRoutingPtr

Legal values**Usage**

This property identifies the location in memory.

Word Pro: MailRouting property

{button ,AL('H_TEXTDOCUMENT_CLASS',0)} [See list of classes](#)

(Read-only)

Data Type

[MailRouting](#)

Syntax

mailroutingvalue = [objectreference].MailRouting

Legal values

Always contains an instance of the MailRouting class.

Usage

Word Pro: MaintainAspectRatio property

```
{button ,AL(^H_LAYOUT_CLASS;H_CELLGROUPLAYOUT_CLASS;H_CELLLAYOUT_CLASS;H_CONNECTEDLAY  
OUT_CLASS;H_COLUMNGROUPLAYOUT_CLASS;H_FOOTERLAYOUT_CLASS;H_FRAMELAYOUT_CLASS;H  
_FRAMEGROUPLAYOUT_CLASS;H_DROPDOWNLAYOUT_CLASS;H_GROUPLAYOUT_CLASS;H_HEADERLAY  
OUT_CLASS;H_NOTELAYOUT_CLASS;H_PAGELAYOUT_CLASS;H_ROWGROUPLAYOUT_CLASS;H_ROWLA  
YOUT_CLASS;H_RUBYLAYOUT_CLASS;H_SUPERTABLEGROUPLAYOUT_CLASS;H_SUPERTABLELAYOUT  
_CLASS;H_TABLELAYOUT_CLASS;H_ENDNOTELAYOUT_CLASS;H_FOOTNOTELAYOUT_CLASS;H_TABLE  
HEADINGLAYOUT_CLASS;H_TOCSUPERTABLELAYOUT_CLASS',0)} See list of classes
```

(Read-write) Allows you to specify whether an image scales proportionately within a layout object.

Data Type

Integer

Syntax

maintainaspectratiovalue = [objectreference].MaintainAspectRatio

[objectreference].MaintainAspectRatio = maintainaspectratiovalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Equivalent to the "Scale proportionately" setting on the Misc panel of the InfoBox for certain layout objects.

Word Pro: MaintainEditor property

{button ,AL(^H_CLICKHERE_CLASS;H_TEXTMARKER_CLASS;H_TEXT_CLASS',0)} [See list of classes](#)

(Write-only)

Data Type

Integer

Syntax

[objectreference].MaintainEditor = maintaineditorvalue

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Word Pro: MarginBetween property

{button ,AL('H_PARAGRAPHBORDER_CLASS','0')} [See list of classes](#)

{button ,AL('H_MARGINBETWEEN_PROPERTY_EXSCRIPT',1')} [See example](#)

(Read-write)

This language element has not yet been defined. You can check for a newer Help file on the Lotus web site.

Data Type

Long

Syntax

marginbetweenvalue = [objectreference].MarginBetween

[objectreference].MarginBetween = marginbetweenvalue

Legal values

Any numeric value of data type Long. The value represents the number of twips. There are 1440 Twips per inch.

Usage

Word Pro: MarginBottom property

```
{button ,AL('H_CHARACTERBORDER_CLASS;H_LAYOUT_CLASS;H_CELLGROUPLAYOUT_CLASS;H_CELLLAY  
OUT_CLASS;H_CONNECTEDLAYOUT_CLASS;H_COLUMNGROUPLAYOUT_CLASS;H_FOOTERLAYOUT_CL  
ASS;H_FRAMELAYOUT_CLASS;H_FRAMEGROUPLAYOUT_CLASS;H_DROPCAPLAYOÜT_CLASS;H_GROÜ  
PLAYOÜT_CLASS;H_HEADERLAYOUT_CLASS;H_NOTELAYOUT_CLASS;H_PAGELAYOUT_CLASS;H_ROW  
GROUPLAYOÜT_CLASS;H_ROWLAYOÜT_CLASS;H_RUBYLAYOÜT_CLASS;H_SUPERTABLEGROUPLAYOÜ  
T_CLASS;H_SUPERTABLELAYOÜT_CLASS;H_TABLELAYOÜT_CLASS;H_ENDNOTELAYOUT_CLASS;H_FOO  
TNOTELAYOUT_CLASS;H_TABLEHEADINGLAYOÜT_CLASS;H_TOCSUPERTABLELAYOÜT_CLASS;H_PARA  
GRAPHBORDER_CLASS',0)} See list of classes
```

```
{button ,AL('H_MARGINBOTTOM_PROPERTY_EXSCRIPT',1)} See example
```

(Read-write) Determines the distance from the border to the bottom edge of the contents in a layout object.

Data Type

Long

Syntax

marginbottomvalue = [objectreference].MarginBottom

[objectreference].MarginBottom = marginbottomvalue

Legal values

Data type is Long but the unit of measurement used for this property is Twips.

Usage

Equivalent to the "Bottom margin" option on the Size & Margins panel of the InfoBox for certain layout objects.

Word Pro: MarginColor property

{button ,AL(^H_APPVIEWPREFS_CLASS',0)} [See list of classes](#)

{button ,AL(^H_MARGINCOLOR_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-only) Stores the color of a document's margin as seen in Layout view when "Show margin in color" is enabled.

Data Type

[Color](#)

Syntax

margincolorvalue = [objectreference].MarginColor

Legal values

Always contains an instance of the Color class.

Usage

Word Pro: MarginLeft property

```
{button ,AL('H_CHARACTERBORDER_CLASS;H_LAYOUT_CLASS;H_CELLGROUPLAYOUT_CLASS;H_CELLLAY  
OUT_CLASS;H_CONNECTEDLAYOUT_CLASS;H_COLUMNGROUPLAYOUT_CLASS;H_FOOTERLAYOUT_CL  
ASS;H_FRAMELAYOUT_CLASS;H_FRAMEGROUPLAYOUT_CLASS;H_DROPCAPLAYOÛT_CLASS;H_GROÛ  
PLAYOÛT_CLASS;H_HEADERLAYOUT_CLASS;H_NOTELAYOUT_CLASS;H_PAGELAYOUT_CLASS;H_ROW  
GROUPLAYOÛT_CLASS;H_ROWLAYOÛT_CLASS;H_RUBYLAYOÛT_CLASS;H_SUPERTABLEGROUPLAYOÛ  
T_CLASS;H_SUPERTABLELAYOÛT_CLASS;H_TABLELAYOÛT_CLASS;H_ENDNOTELAYOUT_CLASS;H_FOO  
TNOTELAYOUT_CLASS;H_TABLEHEADINGLAYOUT_CLASS;H_TOCSUPERTABLELAYOÛT_CLASS;H_PARA  
GRAPHBORDER_CLASS',0)} See list of classes
```

```
{button ,AL('H_MARGINLEFT_PROPERTY_EXSCRIPT',1)} See example
```

(Read-write) Determines the distance from the border to the left edge of the contents in a layout object.

Data Type

Long

Syntax

marginleftvalue = [objectreference].MarginLeft

[objectreference].MarginLeft = marginleftvalue

Legal values

Data type is Long but the unit of measurement used for this property is Twips. There are 1440 Twips per inch.

Usage

Equivalent to the "Left margin" option on the Size & Margins panel of the InfoBox for certain layout objects.

Word Pro: MarginRight property

```
{button ,AL('H_CHARACTERBORDER_CLASS;H_LAYOUT_CLASS;H_CELLGROUPLAYOUT_CLASS;H_CELLLAY  
OUT_CLASS;H_CONNECTEDLAYOUT_CLASS;H_COLUMNGROUPLAYOUT_CLASS;H_FOOTERLAYOUT_CL  
ASS;H_FRAMELAYOUT_CLASS;H_FRAMEGROUPLAYOUT_CLASS;H_DROPCAPLAYOÜT_CLASS;H_GROÜ  
PLAYOÜT_CLASS;H_HEADERLAYOUT_CLASS;H_NOTELAYOUT_CLASS;H_PAGELAYOUT_CLASS;H_ROW  
GROUPLAYOÜT_CLASS;H_ROWLAYOÜT_CLASS;H_RUBYLAYOÜT_CLASS;H_SUPERTABLEGROUPLAYOÜ  
T_CLASS;H_SUPERTABLELAYOÜT_CLASS;H_TABLELAYOÜT_CLASS;H_ENDNOTELAYOUT_CLASS;H_FOO  
TNOTELAYOUT_CLASS;H_TABLEHEADINGLAYOÜT_CLASS;H_TOCSUPERTABLELAYOÜT_CLASS;H_PARA  
GRAPHBORDER_CLASS',0)} See list of classes
```

```
{button ,AL('H_MARGINRIGHT_PROPERTY_EXSCRIPT',1)} See example
```

(Read-write) Determines the distance from the border to the right edge of the contents in a layout object.

Data Type

[Long](#)

Syntax

marginrightvalue = [objectreference].MarginRight

[objectreference].MarginRight = marginrightvalue

Legal values

Data type is Long but the unit of measurement used for this property is [Twips](#).

Usage

Equivalent to the "Right margin" option on the Size & Margins panel of the InfoBox for certain layout objects.

Word Pro: MarginTop property

```
{button ,AL('H_CHARACTERBORDER_CLASS;H_LAYOUT_CLASS;H_CELLGROUPLAYOUT_CLASS;H_CELLLAY  
OUT_CLASS;H_CONNECTEDLAYOUT_CLASS;H_COLUMNGROUPLAYOUT_CLASS;H_FOOTERLAYOUT_CL  
ASS;H_FRAMELAYOUT_CLASS;H_FRAMEGROUPLAYOUT_CLASS;H_DROPCAPLAYOOUT_CLASS;H_GROU  
PLAYOOUT_CLASS;H_HEADERLAYOUT_CLASS;H_NOTELAYOUT_CLASS;H_PAGELAYOUT_CLASS;H_ROW  
GROUPLAYOOUT_CLASS;H_ROWLAYOOUT_CLASS;H_RUBYLAYOOUT_CLASS;H_SUPERTABLEGROUPLAYO  
T_CLASS;H_SUPERTABLELAYOOUT_CLASS;H_TABLELAYOOUT_CLASS;H_ENDNOTELAYOUT_CLASS;H_FOO  
TNOTELAYOUT_CLASS;H_TABLEHEADINGLAYOUT_CLASS;H_TOCSUPERTABLELAYOOUT_CLASS;H_PARA  
GRAPHBORDER_CLASS',0)} See list of classes
```

```
{button ,AL('H_MARGINTOP_PROPERTY_EXSCRIPT',1)} See example
```

(Read-write) Determines the distance from the border to the top edge of the contents in a layout object.

Data Type

Long

Syntax

margintopvalue = [objectreference].MarginTop

[objectreference].MarginTop = margintopvalue

Legal values

Data type is Long but the unit of measurement used for this property is Twips.

Usage

Equivalent to the "Top margin" option on the Size & Margins panel of the InfoBox for certain layout objects.

Word Pro: MarkCharacter property

{button ,AL('H_OPTIONS_CLASS',0)} [See list of classes](#)

{button ,AL('H_MARKCHARACTER_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Allows you to indicate a character for marked edits in the document.

Data Type

[Integer](#)

Syntax

markcharactervalue = [objectreference].MarkCharacter

[objectreference].MarkCharacter = markcharactervalue

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Equivalent to choosing File - Document Properties - Document, clicking the Options tab and, in the "Review and comment options" section, selecting "Character" in "Indicate marked edits in column with." You must set the MarkType property to "Character," in order to use the character you choose here. The character is its ASCII number representation. Use ASC('char') to get the value (or if reading the property, use CHR(property)).

{button ,AL('H_MARKTYPE_PROPERTY_MEMDEF',0)} [See related topics](#)

Word Pro: MarkerClass property

{button ,AL('H_MARKER_CLASS;H_POWERFIELD_CLASS;H_CLICKHERE_CLASS;H_TEXTMARKER_CLASS;H_RUBYMARKER_CLASS;H_TABLEMARKER_CLASS',0)} [See list of classes](#)

(Read-write)

Data Type

Variant (Enumerated)

MarkerType

Syntax

markerclassvalue = [objectreference].MarkerClass

[objectreference].MarkerClass = markerclassvalue

Legal values

\$LwpMarkerTypeBookmark (590)

\$LwpMarkerTypeClickhere (593)

\$LwpMarkerTypeDde (591)

\$LwpMarkerTypeDefault (589)

\$LwpMarkerTypeField (592)

\$LwpMarkerTypeRuby (594)

Usage

Word Pro: MarkerName property

{button ,AL(^H_BOOKMARK_CLASS',0)} [See list of classes](#)

(Read-only) Indicates the internal (hexadecimal) representation of a bookmark name.

Data Type

String

Syntax

markernamevalue = [objectreference].MarkerName

Legal values**Usage**

If you don't name a bookmark, Word Pro always assigns an internal (hexadecimal) name to the bookmark. This name cannot be changed.

Word Pro: Markers property

{button ,AL(^H_FOUNDRY_CLASS',0)} [See list of classes](#)

(Read-only) An object created from the MarkerCollection class. This object provides access to Marker objects.

Data Type

[MarkerCollection](#)

Syntax

markersvalue = [objectreference].Markers

Legal values

Always contains an instance of the MarkerCollection class.

Usage

When accessed through the Foundry property on a Division object, the collection object in this property provides access to all the Marker objects contained in that Division object.

When accessed through the AppFoundry property on the WPAApplication object, this collection object provides access to all the Marker objects contained in the Word Pro Clipboard.

When accessed through the TempFoundry property on the WPAApplication object, this collection object provides access to all the Marker objects placed in TempFoundry by WordPro or a script.

When accessed through the Foundry property on the WPAApplication object, this collection object provides access to all the Marker objects contained in the currently active Division object.

This property is not used in the Foundry property on the TextDocument object.

For more information about collection classes, see [Overview: Word Pro LotusScript Collection Classes](#).

Word Pro: MarkPosition property

{button ,AL('H_OPTIONS_CLASS',0)} [See list of classes](#)

{button ,AL('H_MARKPOSITION_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Allows you to position where marked edits appear in the document.

Data Type

Variant (Enumerated)

MarkPosition

Syntax

markpositionvalue = [objectreference].MarkPosition

[objectreference].MarkPosition = markpositionvalue

Legal values

<u>Value</u>	<u>Effect</u>
\$LwpMarkPositionBothSides (585)	
\$LwpMarkPositionLeft (583)	
\$LwpMarkPositionRight (584)	

Usage

Equivalent to choosing File - Document Properties - Document, clicking the Options tab, and, in the "Review and comment options" section, selecting an option in "Position in margin."

Word Pro: MarkType property

{button ,AL(`H_OPTIONS_CLASS',0)} [See list of classes](#)

{button ,AL(`H_MARKTYPE_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Allows you to indicate the type of marked edits in the document.

Data Type

Variant (Enumerated)

MarkType

Syntax

marktypevalue = [objectreference].MarkType

[objectreference].MarkType = marktypevalue

Legal values

<u>Value</u>	<u>Effect</u>
\$LwpMarkTypeBars (587)	
\$LwpMarkTypeChar (588)	
\$LwpMarkTypeNone (586)	

Usage

Equivalent to choosing File - Document Properties - Document, clicking the Options tab, and selecting an option for "Indicate marked edits in column with" in the "Review and comment options." If you set this option to "Character," use the MarkCharacter property to set the desired character.

{button ,AL(`H_MARKCHARACTER_PROPERTY_MEMDEF',0)} [See related topics](#)

Word Pro: MasterName property

```
{button ,AL(^H_LAYOUT_CLASS;H_CELLGROUPLAYOUT_CLASS;H_CELLLAYOUT_CLASS;H_CONNECTEDLAY  
OUT_CLASS;H_COLUMNGROUPLAYOUT_CLASS;H_FOOTERLAYOUT_CLASS;H_FRAMELAYOUT_CLASS;H  
_FRAMEGROUPLAYOUT_CLASS;H_DROP_CAPLAYOUT_CLASS;H_GROUPLAYOUT_CLASS;H_HEADERLAY  
OUT_CLASS;H_NOTELAYOUT_CLASS;H_PAGELAYOUT_CLASS;H_ROWGROUPLAYOUT_CLASS;H_ROWLA  
YOUT_CLASS;H_RUBYLAYOUT_CLASS;H_SUPERTABLEGROUPLAYOUT_CLASS;H_SUPERTABLELAYOUT  
_CLASS;H_TABLELAYOUT_CLASS;H_ENDNOTELAYOUT_CLASS;H_FOOTNOTELAYOUT_CLASS;H_TABLE  
HEADINGLAYOUT_CLASS;H_TOCSUPERTABLELAYOUT_CLASS',0)} See list of classes
```

(Read-only) Returns the name of the layout unless the layout is part of a complex layout, in which case, it returns the name of the complex layout object.

Data Type

String

Syntax

masternamevalue = [objectreference].MasterName

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

A complex layout is one which uses separate layout objects for left and right pages.

Word Pro: Master property

{button ,AL(^H_DIVISION_CLASS;H_TEXTDOCUMENT_CLASS',0)} [See list of classes](#)

(Read-only)

Data Type

[TextDocument](#)

Syntax

mastervalue = [objectreference].Master

Legal values

Always contains an instance of the TextDocument class.

Usage

Word Pro: MatchType property

{button ,AL(^H_FINDANDREPLACE_CLASS',0)} [See list of classes](#)

(Read-write) Enables you to narrow a search in Find & Replace.

Data Type

Variant (Enumerated)

FindMatch

Syntax

matchtypevalue = [objectreference].MatchType

[objectreference].MatchType = matchtypevalue

Legal values

<u>Value</u>	<u>Effect</u>
\$LwpFindMatchOnBeginningOfWord (283)	Sets the option to match words beginning with the find text.
\$LwpFindMatchOnEndingOfWord (284)	Sets the option to match words ending with the find text.
\$LwpFindMatchOnWholeWord (282)	Sets the option to match the find text to whole words only.
\$LwpFindMatchWithinAWord (285)	Sets the option to match words containing the find text.

Usage

Equivalent to choosing Edit - Find & Replace Text, and choosing an option in the "Find & Replace" box.

Word Pro: MaxBottomBorder property

{button ,AL(^H_BASETABLE_CLASS;H_FOOTNOTETABLE_CLASS;H_PARALLELCOLUMNS_CLASS;H_GLOSSARY_CLASS;H_TABLEHEADING_CLASS;H_TABLE_CLASS',0)} [See list of classes](#)

(Read-only) Returns the specific maximum width that can be set for the bottom border of a row in a table object.

Data Type

Long

Syntax

maxbottombordervalue = [objectreference].MaxBottomBorder

Legal values

Data type is Long but the unit of measurement used for this property is Twips. There are 1440 Twips per inch.

Usage

Word Pro: MaxBottomGutter property

{button ,AL(^H_BASETABLE_CLASS;H_FOOTNOTETABLE_CLASS;H_PARALLELCOLUMNS_CLASS;H_GLOSSARY_CLASS;H_TABLEHEADING_CLASS;H_TABLE_CLASS',0)} [See list of classes](#)

(Read-only) Returns the specific maximum width that can be set for the bottom gutter of a row in a table.

Data Type

Long

Syntax

maxbottomguttervalue = [objectreference].MaxBottomGutter

Legal values

Data type is Long but the unit of measurement used for this property is Twips. There are 1440 Twips per inch.

Usage

Word Pro: MaxContentHeight property

```
{button ,AL(^H_BASECONTAINER_CLASS;H_CELLCONTAINER_CLASS;H_FRAMECONTAINER_CLASS;H_NOTE  
CONTAINER_CLASS;H_RUBYCONTAINER_CLASS;H_DROPDOWNCONTAINER_CLASS;H_PAGECONTAINER_  
CLASS;H_SUBPAGECONTAINER_CLASS;H_SUPERPAGECONTAINER_CLASS;H_ROWCONTAINER_CLASS;  
H_SUPERTABLECONTAINER_CLASS;H_TABLECONTAINER_CLASS;H_PARALLELCOLSCONTAINER_CLASS  
;H_TABLEONLYCONT_CLASS',0)} See list of classes
```

(Read-only) Returns the maximum height to which the content of a container can expand and rotate.

Data Type

Long

Syntax

maxcontentheightvalue = [objectreference].MaxContentHeight

Legal values

Data type is Long but the unit of measurement used for this property is Twips. There are 1440 Twips per inch.

Usage

'Example: Shadow property

'This example creates a frame, and places a shadow on the bottom
'and right sides of the frame.

.CreateFrame

.Frame.Layout.**Shadow**.XPosition = 101

.Frame.Layout.**Shadow**.YPosition = 101

```
'Example: SendFrameToBackOne method
' This example creates two frames and changes the order of the layering
' for the two frames.
' RUNTIME DEPENDENCIES: You must have a document open for this script to work.

.NewFrame 3285, 1200, 1575, 1830
.Frame.Layout.Background.Color.Red = 82
.Frame.Layout.Background.Color.Blue = 239
.Frame.Layout.Background.Color.Green = 145
.Frame.Layout.Background.Color.Override = $LwpColorOverrideRgb
.Frame.Anchor $LwpAnchorWhereLayout, $LwpConditionTypeOnlyspecificpage,
$LwpRelativeTypeLytParent

.Deselect

.NewFrame 5285, 2200, 1575, 1830
.Frame.Anchor $LwpAnchorWhereLayout, $LwpConditionTypeOnlyspecificpage,
$LwpRelativeTypeLytParent
.Frame.Layout.Background.Color.Red = 182
.Frame.Layout.Background.Color.Blue = 139
.Frame.Layout.Background.Color.Green = 45
.Frame.Layout.Background.Color.Override = $LwpColorOverrideRgb

MessageBox "Click OK to send frame to back. ",MB_OK,"Example Script"
.SendFrameToBackOne
```

```
'Example: SendFrameToBack method
' This example creates two frames and changes the order of the layering
' for the two frames.
' RUNTIME DEPENDENCIES: You must have a document open for this script to work.

.NewFrame 3285, 1200, 1575, 1830
.Frame.Layout.Background.Color.Red = 82
.Frame.Layout.Background.Color.Blue = 239
.Frame.Layout.Background.Color.Green = 145
.Frame.Layout.Background.Color.Override = $LwpColorOverrideRgb
.Frame.Anchor $LwpAnchorWhereLayout, $LwpConditionTypeOnlyspecificpage,
$LwpRelativeTypeLytParent

.Deselect

.NewFrame 5285, 2200, 1575, 1830
.Frame.Anchor $LwpAnchorWhereLayout, $LwpConditionTypeOnlyspecificpage,
$LwpRelativeTypeLytParent
.Frame.Layout.Background.Color.Red = 182
.Frame.Layout.Background.Color.Blue = 139
.Frame.Layout.Background.Color.Green = 45
.Frame.Layout.Background.Color.Override = $LwpColorOverrideRgb

MessageBox "Click OK to send frame to back. ",MB_OK,"Example Script"
.SendFrameToBack

MessageBox "Click OK to bring frame to front. ",MB_OK,"Example Script"
.BringFrameToFront
```

```
'Example: SendMailSelectedText method
' This example inserts some sample text into the current document, selects the
' text, and then mails the selected text.
' RUNTIME DEPENDENCIES: You must have a document open for this script to work and a
VIM compliant mail application must be available.

.Type "This is some sample text to be mailed."
.SelectParagraph
.SendMailSelectedText
```

```
'Example: SequenceName property
'This example asks the user for the name of the numbering sequence
'to use for frame captions.

' First get what we're using now.
CurName = .ActiveDocument.FrameCaptionOptions.SequenceName
' Ask user for the new name. (Defaults to old sequence name)
NewName = Inputbox ("Type the name of the sequence you want to use for frame
captions:", "Example Script", CurName)
If NewName = "" Then Exit Sub 'user cancelled
'Apply the new sequence name to the property.
.ActiveDocument.FrameCaptionOptions.SequenceName = NewName
```

```
'Example: SequenceNumber property
'This example asks the user for the value of the numbering sequence
'to use for frame captions.

' First get what we're using now.
CurNumber = .ActiveDocument.FrameCaptionOptions.SequenceNumber
' Ask user for the new sequence number. (Defaults to old sequence number)
NewNumber = Inputbox ("What is the next number in the sequence you want to use for
frame captions:", "Example Script", CurNumber)
If IsNumeric(NewNumber) = False Then Exit Sub 'user cancelled
'Apply the new sequence number to the property.
NewNumber = NewNumber - 1 ' subtract 1 to get current number
.ActiveDocument.FrameCaptionOptions.SequenceNumber = NewNumber
```

'Example: SetData method

' This example creates a dataset named 'ExampleDataSet' off of the active
' document. The 'FirstName' and 'LastName' items are created and filled with
' data. Finally the values for the dataset items are printed to the Script
' Editor Output panel. Since no dataset item named 'Address' was defined, the
' default dataset value will be printed in the last statement.
' RUNTIME DEPENDENCIES: You must have a document open for this script to work.

```
Dim DataSetName as String
```

```
Dim Defaultvalue as String
```

```
Dim DataSet As WPDataSetCollection
```

```
Set DataSet = .ActiveDocument.WPDataSets
```

```
DataSetName = "ExampleDataSet"
```

```
Defaultvalue = "Default"
```

```
DataSet(DataSetName).SetData "FirstName","John"
```

```
DataSet(DataSetName).SetData "LastName","Doe"
```

```
Print DataSet(DataSetName).GetData("FirstName",Defaultvalue)
```

```
Print DataSet(DataSetName).GetData("LastName",Defaultvalue)
```

```
Print DataSet(DataSetName).GetData("Address",Defaultvalue)
```

```
'Example: SetFormula method
' This example creates a table and enters a formula. The formula is then
' retrieved and printed to the Lotus Script Output panel.
' RUNTIME DEPENDENCIES: You must have a document open for this script to work.

.CreateTable False, "Default Table", 5,5
.Table.CellLayout(4,0).GotoLayout
.Table.CellEngine.SetFormula 4,0,"@SUM(A1:A2)"
Print .Table.CellEngine.GetFormula(4, 0)
```


'Example: SetLastUsedFilter method

' This example inserts the 'Turtle' Word Pro Drawing graphic into the current
' document. The graphic is placed in a frame based upon the 'Default
' Graphic/OLE' frame style.
' RUNTIME DEPENDENCIES: You must have a document open and have installed the
' the Word Pro clipart for this script to 'work.

Dim FileName as String

FileName = .ApplicationWindow.UserInterfacePrefs.GraphicPath & "\Turtle.SDW"

.ApplicationWindow.Filter.SetLastUsedFilter \$LwpFilterTypeGraphic,"Word Pro Draw"

.ApplicationWindow.UserInterfacePrefs.LastGraphicType = "Word Pro Draw"

.ImportGraphic FileName, ".SDW", False, False, "Default Graphic/OLE"

Note OLE is not supported under OS/2.

```
'Example: SetLineOneSide method
Option Public

Sub Main
    .Table.TableLine.BeginCustomLines
    .Table.TableLine.SetLineOneSide &H4, $LwpBorderPatternDb1Thick, 80,
16711680, 0
    .Table.TableLine.SetLineOneSide &H8, $LwpBorderPatternDb1Thick, 80,
        16711680, 0
    .Table.TableLine.EndCustomLines
End Sub
```

```
'Example: SetNamedProperty method  
' This example creates a named property, 'ExampleProp' on the active document  
' and assigns it a value. The value is then printed to the LotusScript Output panel.  
' RUNTIME DEPENDENCIES: You must have a document open for this script to work.
```

```
.ActiveDocument.SetNamedProperty "ExampleProp", "Here is some data."  
Print .ActiveDocument.GetNamedProperty "ExampleProp"
```

```
'Example: SetOverrideText method
' This example creates a new button to the status bar and then adds text to
' the button. The STATUSBARBUTTONOVERRIDE TEXT is then bound to the
' SetTheButtonText subroutine to set the button text during needs repainting.
' RUNTIME DEPENDENCIES: You must have a document open for this script to work.
```

```
Dim ButtonName As String
Dim NewButton As StatusBarButton
With .ApplicationWindow.StatusBar
    ButtonName = .CreateNewbutton (0,0,100,&H1) 'create the new button
    Set NewButton = .StatusBarButtons(ButtonName)
    With .StatusBarButtons(ButtonName)
        .SetOverrideText("New Button...")
        Call .SetButtonText("Button",True)
        .InvalidateButton
        On Event STATUSBARBUTTONOVERRIDE TEXT From NewButton Call
            SetTheButtonText
    End With
    .InvalidateWholeBar ' Force the bar to repaint
End With
End Sub

Sub SetTheButtonText (Source As StatusBarButton, ButtonName As String)
    'Add the the button text each time the status bar needs repainting.
    Source.SetOverrideText("New Button...")
    End 2
End Sub
```

'Example: SetPageBottomMargin method

'This example sets the bottom margin for the current page to 2 inches,

'and the top margin for the current page to 1 1/2 inches.

.SetPageBottomMargin 2880, True

.SetPageTopMargin 2160, True

'Example: SetPageTopMargin method

'This example sets the bottom margin for the current page to 2 inches,
'and the top margin for the current page to 1 1/2 inches.

.SetPageBottomMargin 2880, True

.SetPageTopMargin 2160, True

```
'Example: SetPattern method
' This example create a table with 10 columes and 1 row and changes the color
' pattern for every other row.
' RUNTIME DEPENDENCIES: You must have a document open for this script to work.

.CreateTable False, "Default Table", 1, 10
.SelectEntireTable

BackgroundColor = 92654
PatternColor = 23424

.Table.TableFill.SetPattern $LwpTableFillStyleEveryotherrow,
$LtsFillDottedZigzag ,BackgroundColor, PatternColor
.Deselect
```

```
'Example: SetRGB method
' This example creates a table with 5 rows and 5 columns into the current
' document. The background and pattern colors are then changed for the current
' cell.
' RUNTIME DEPENDENCIES: You must have a document open for this script to work.

.CreateTable False, "Default Table", 5,5
.Table.TableFill.Background.Pattern = $LtsFillNwToSeGrad

.Table.CurrentCell.Background.BackColor.SetRGB 255,255,255

.Table.TableFill.Background.Color.Override = $LwpColorOverrideRgb
.Table.TableFill.Background.BackColor.Red = 82
.Table.TableFill.Background.BackColor.Green = 145
.Table.TableFill.Background.BackColor.Blue = 239

.Table.TableFill.Background.BackColor.SetRGB 82,145,239

.Table.TableFill.Background.BackColor.Override = $LwpColorOverrideRgb
.Table.TableFill.FillStyle = $LwpTableFillStyleAll
```



```
'Example: SetStyle method
' This example inserts some text into the current document, changes some
' text properties and then creates a new paragraph style based on those
' properties.
' RUNTIME DEPENDENCIES: You must have a document open for this script to work.
```

```
With .Text
    .InsertText "This is some sample text."
    .Select $LwpSelectObjectTypeParagraph
    .Font.Size = 15.00
    .Font.Bold = True
    .Alignment.AlignmentType = $LtsAlignmentHorizCenter
    .SetStyle $LwpStyleTypeParagraph, "New Example Style",
End With
```

'Example: Shade method

' This example shades the first 8 words the the example sentence inserted
' into the current document.

' RUNTIME DEPENDENCIES: You must have a document open with the cursor
' positioned to the left of some text for this script to work.

.Text.InsertText "The first eight words of this sentence will be shaded."

.Text.Backward \$LwpNavigateObjectTypeSentence, 1

.Text.Shade \$LwpLocationTypeWord,\$LwpNavigateDirectionRight,8

```
'Example: ShowDivisionTabs property
'This example asks the user whether to display all division divider tabs in
'the document, and then sets the appropriate option.

stat = MessageBox ("Do you want to show all division tabs?", 36, "Example Script")
If stat = 6 Then ' user said yes
    .ActiveDocument.DocControl.ShowDivisionTabs = True
Else
    .ActiveDocument.DocControl.ShowDivisionTabs = False
End If
```

'Example: ShowHeaderFooterBar property

'This example asks the user whether to display the Header/Footer bar when the
'insertion point is in a header or footer, and then sets the appropriate option.

```
stat = MessageBox ("Do you want to show the header/footer bar?", 36, "Example Script")
```

```
If stat = 6 Then ' user said yes
```

```
    .ApplicationWindow.UserInterfacePrefs.ShowHeaderFooterBar = True
```

```
Else
```

```
    .ApplicationWindow.UserInterfacePrefs.ShowHeaderFooterBar = False
```

```
End If
```

```
'Example: ShowHiddenText property
'This example sets the hidden attribute, then types some text.
'The option to show hidden text is then deselected, causing the text to be hidden.
.Text.Attributes.HiddenMode = True ' put on the Hidden attribute.
.Type "Now is the time for all good men to come to the aid of their party."
.Text.Attributes.HiddenMode = False
CurrentDocument.DivisionOptions.ShowHiddenText = False ' don't display hidden text
MessageBox "Click OK to show hidden text", MB_OK, "Example Script"
CurrentDocument.DivisionOptions.ShowHiddenText = True
```

'Example: ShowIconBars method

' This example hides all SmartIcon bars the redisplays them after the message
' box is closed.

.ApplicationWindow.IconBarManager.HideIconBars

MessageBox "Click OK to show SmartIcon bars.",MB_OK,"Example Script"

.ApplicationWindow.IconBarManager.ShowIconBars

```
'Example: ShowInContext property
Dim IcnPallet As String
Dim MsgStr As String
Dim IcnMgr As IconBarManager
IcnPallet = "Comment Tools"
Set IcnMgr = .ApplicationWindow.IconBarManager

' Set icon palette to show in its context
IcnMgr.IconBars(IcnPallet).ShowInContext = True
' This will force a redraw of IconBars
IcnMgr.ShowIconBars
MsgStr = "|" & IcnPallet
MsgStr = MsgStr & "|" palette is now displayed, click OK to hide this palette"
MessageBox MsgStr, 48, "Script Example"

' Reset icon palette not to show in its context
IcnMgr.IconBars(IcnPallet).ShowInContext = False
IcnMgr.ShowIconBars
```

```
'Example: ShowMailDisabled property
'This example asks the user whether to disable checking for new mail while
'Word Pro is running, and then sets the appropriate option.

stat = MessageBox ("Do you want to disable mail checking?", 36, "Example Script")
If stat = 6 Then ' user said yes
    .ApplicationWindow.UserInterfacePrefs.ShowMailDisabled = True
Else
    .ApplicationWindow.UserInterfacePrefs.ShowMailDisabled = False
End If
```


'Example: ShowMarks property

'This example inserts a Click Here block at the insertion point, then

'hides the marks around the block.

```
MyClickHere = .InsertClickHere()  
.Division.Foundry.ClickHeres(MyClickHere).Prompt.Clear  
.Division.Foundry.ClickHeres(MyClickHere).Prompt.InsertText "Click here to type Text",  
False, $LwpTextTypeNative  
.Division.Foundry.ClickHeres(MyClickHere).MirrorName = ""  
.Division.Foundry.ClickHeres(MyClickHere).Action = 1  
.Division.Foundry.ClickHeres(MyClickHere).HelpText = "Click here to type Text"  
.Division.Foundry.ClickHeres(MyClickHere).UsesHelp = False  
.Division.Foundry.ClickHeres(MyClickHere).TabOrder = 3  
.Division.Foundry.ClickHeres(MyClickHere).TabExits = True  
.Division.Foundry.ClickHeres(MyClickHere).ReturnExits = False  
.Division.Foundry.ClickHeres(MyClickHere).Name = MyClickHere  
.Division.Foundry.ClickHeres(MyClickHere).AllowListEdit = True  
.Division.Foundry.ClickHeres(MyClickHere).AllowListMultiValues = False  
.Division.Foundry.ClickHeres(MyClickHere).RemoveNamedProperty  
.Division.Foundry.ClickHeres(MyClickHere).SetNamedProperty "Collect", "Off"  
.Division.Foundry.ClickHeres(MyClickHere).SetNamedProperty "Required", "Off"  
.Division.Foundry.ClickHeres(MyClickHere).SetNamedProperty "Notes/FX", "Off"  
.UpdateUI
```

'Now hide the marks around the Click Here block

```
MessageBox "Click OK to hide the marks around the Click Here Block", MB_OK, "Example  
Script"
```

```
.Division.Foundry.ClickHeres(MyClickHere).ShowMarks = False  
.Text.InsertText " Click Here Marks have been turned off. ", True
```

'Example: ShowNoWelcomeBox property

'This example asks the user whether to display the Welcome box when Word Pro
'is started, and then sets the appropriate option.

```
stat = MessageBox ("Do you want to display the Welcome screen?", 36, "Example Script")
```

```
If stat = 6 Then ' user said yes
```

```
    .ApplicationWindow.UserInterfacePrefs.ShowNoWelcomeBox = False
```

```
Else
```

```
    .ApplicationWindow.UserInterfacePrefs.ShowNoWelcomeBox = True
```

```
End If
```

```
'Example: ShowScrollBars method
' This example hides the vertical and horizontal scroll bars if the are
' visible and shows them if they are hidden.
' RUNTIME DEPENDENCIES: You must have a document open for this script to work.
```

```
Dim ScrollVert as Long
Dim ScrollHorz as Long
MessageBox "Click OK to hide scroll bars ",MB_OK,"Example Script"
ScrollVert = False
ScrollHorz = False
.ActiveDocWindow.ShowScrollBars ScrollVert,ScrollHorz
MessageBox "Click OK to show scroll bars ",MB_OK,"Example Script"
ScrollVert = True
ScrollHorz = True
.ActiveDocWindow.ShowScrollBars ScrollVert,ScrollHorz
```

'Example: ShowStatistics property
'This example asks the user whether to check display usage statistics following
'grammar check, and then sets the appropriate option.

```
stat = MessageBox ("Do you want to display grammar statistics?", 36, "Example Script")
If stat = 6 Then    ' user said yes
    .ApplicationWindow.UserInterfacePrefs.GrammarOptions.ShowStatistics = True
Else
    .ApplicationWindow.UserInterfacePrefs.GrammarOptions.ShowStatistics = True
End If
```

```
'Example: ShowStatusBar method
' This example hides the status bar if it is visible, and shows it if it is
' hidden.
If .ApplicationWindow.StatusBar.Visible = True Then
    .ApplicationWindow.StatusBar.HideStatusBar
Else
    .ApplicationWindow.StatusBar.ShowStatusBar
End If
```

'Example: ShowTabs property

'This example asks the user whether to show the divider tab for the current
'division, and then sets the appropriate option.

```
stat = MessageBox ("Do you want to show the divider tab for this division?", 36,  
"Example Script")
```

```
If stat = 6 Then ' user said yes
```

```
    .Division.DivisionInfo.ShowTabs = True
```

```
Else
```

```
    .Division.DivisionInfo.ShowTabs = False
```

```
End If
```

```
'Example: Show method
'[DocWindow.Show]
' This example gets the name of the document to activate and then
' cycles through all the documents to find the one requested
' RUNTIME DEPENDENCIES: You must have a document open for this script to work.
```

```
Dim DocName As String
DocName = InputBox("Enter the name of the document to activate:")
Forall Document In .Documents
    If Document.Name = DocName Then
        Document.Activate
        .ActiveDocWindow.Show
    End If
End Forall
```

```
'[IconBar.Show]
' Display a specific set of SmartIcons
Dim IcnPallet As String
Dim MsgStr As String
Dim IcnMgr As IconBarManager
IcnPallet = "Comment Tools"
Set IcnMgr = .ApplicationWindow.IconBarManager
```

```
' Set the context and show the bar
IcnMgr.IconBars(IcnPallet).ShowInContext = True
IcnMgr.IconBars(IcnPallet).Show
```

```
MsgStr = "|" & IcnPallet
MsgStr = MsgStr & "|" palette is now displayed, click OK to hide this palette"|
MessageBox MsgStr, 48, "Script Example"
```

```
' Reset the context and hide the bar. You can hide the bar without resetting
' the context first. But the bar will reappear once the context in the
' document returns to the setting for this palette
IcnMgr.IconBars(IcnPallet).ShowInContext = False
IcnMgr.IconBars(IcnPallet).HideIconBar
```

```
'Example: SimulateButtonClick method
' This example simulates clicking on the font status bar button.
' RUNTIME DEPENDENCIES: You must have a document open for this script to work.

Dim StatBar As StatusBar
Dim ButtonId as Integer

Set StatBar = .ApplicationWindow.StatusBar
Forall Button In StatBar.StatusBarButtons
    ButtonId = Button.GetButtonId
    If (ButtonId = StatBar.GetStandardButtonId($LwpStandButtFontButton)) Then
        Button.SimulateButtonClick
    End If
End Forall
```



```
'Example: Size property
' Change the font size from 12 points to 10 points for every paragraph style
Sub Main
Forall x In .division.foundry.paragraphstyles
    if x.font.size = 12 then x.font.size = 10
End Forall
End Sub
```

'Example: SizingUnits property

'This example sets the Word Pro unit of measurement to centimeters.

.ApplicationWindow.UserInterfacePrefs.**SizingUnits** = \$ItsScaleModeCentimeter

'Example: Skipped property

'This example types a line of bulleted text, then skips the bullet for a line, then resumes.

```
.InsertBullet "Wingdings", "Ÿ"  
.Type "This is a line of bulleted text[Enter]"  
.Text.Bullet.Skipped = $LwpCommandStateOn  
.Type "We are skipping the bullet on this line[Enter]"  
.Text.Bullet.Skipped = $LwpCommandStateOff  
.Type "This is a line of bulleted text[Enter]"
```

```
'Example: SmallCaps method
' This example displays the popup style menu located on Word Pro's status bar.
' RUNTIME DEPENDENCIES: You must have a document open for this script to work.
.Type "This Is A Test Of Small Caps."
.SelectParagraph
.SmallCaps
MessageBox "Click OK undo the smallcaps change.",MB_OK,"Example Script"
.SmallCaps
```

'Example: SmallCaps property

'This example enables the Small Caps attribute, types some text, then disables the attribute.

.Text.Font.**SmallCaps** = True

.Type "This is some text in small caps."

.Text.Font.**SmallCaps** = False

'Example: SmallFileFormat property

'This example disables compression of Word Pro files when they are saved.

.Application.Preferences.**SmallFileFormat** = False

'Example: SmartCorrects property

'This example adds an entry to the English SmartCorrect list.

```
.Application.SmartCorrects("English (United States)").AddSmartCorrect "lotus", "Lotus"
```

'Example: SmartCorrect property

'This example script sets the SmartCorrect options for correcting initial caps
'and Smart Quotes.

.Application.**SmartCorrect**.DoInitialCaps = True

.Application.**SmartCorrect**.DoSmartQuotes = True

'Example: SmartFill property

'This example creates a ZSmartFill list called "Day Parts". It then adds
'items to the new list.

'Note that 1033 represents the language number for the American language.

```
.Application.SmartFill(1033).CreateEmptyList "Day Parts", True  
.Application.SmartFill(1033).AddStringToList "Day Parts", "Morning"  
.Application.SmartFill(1033).AddStringToList "Day Parts", "Afternoon"  
.Application.SmartFill(1033).AddStringToList "Day Parts", "Evening"  
.Application.SmartFill(1033).AddStringToList "Day Parts", "Night"
```

```
'Example: SmartLabelsEnabled property
'This example script asks the user whether to enable the use of automatic
'totalling in Tables, and then sets the appropriate option.

stat = MessageBox ("Do you want to be able to type 'total' to get totals in tables?",
36, "Example Script")
If stat = 6 Then ' user said yes
    .Application.Preferences.SmartLabelsEnabled = True
Else
    .Application.Preferences.SmartLabelsEnabled = False
End If
```

```
'Example: SmartSumColumn method
' This example creates a table with 5 rows and 5 columns into the current
' document. Headers are created for the first column and row. Numbers are
' inserted elsewhere and the column and rows are totaled.
' RUNTIME DEPENDENCIES: You must have a document open for this script to work.
```

```
Dim MyTable As Table
Dim RowCounter As Integer
Dim ColumnCounter As Integer

.CreateTable False, "Default Table", 5,5
Set MyTable = .Table

For RowCounter = 1 To 3
    MyTable.CellLayout (RowCounter,0) .GotoLayout
    .Text.InsertText "Sales Rep" & Format$(RowCounter)
Next
For ColumnCounter = 1 To 3
    MyTable.CellLayout (0,ColumnCounter) .GotoLayout
    .Text.InsertText "Year" & Format$(ColumnCounter)
Next
For ColumnCounter = 1 To 3
    For RowCounter = 1 To 3
        MyTable.CellLayout (RowCounter,ColumnCounter) .GotoLayout
        .Text.InsertText Format$(RowCounter * ColumnCounter)
    Next
Next
For ColumnCounter = 1 To 3
    MyTable.CellLayout (4,ColumnCounter) .GotoLayout
    .SmartSumColumn
Next
For RowCounter = 1 To 4
    MyTable.CellLayout (RowCounter,4) .GotoLayout
    .SmartSumRow
Next
```

```
'Example: SmartSumRow method
' This example creates a table with 5 rows and 5 columns into the current
' document. Headers are created for the first column and row. Numbers are
' inserted elsewhere and the column and rows are totaled.
' RUNTIME DEPENDENCIES: You must have a document open for this script to work.
```

```
Dim MyTable As Table
Dim RowCounter As Integer
Dim ColumnCounter As Integer

.CreateTable False, "Default Table", 5,5
Set MyTable = .Table

For RowCounter = 1 To 3
    MyTable.CellLayout (RowCounter,0) .GotoLayout
    .Text.InsertText "Sales Rep" & Format$(RowCounter)
Next
For ColumnCounter = 1 To 3
    MyTable.CellLayout (0,ColumnCounter) .GotoLayout
    .Text.InsertText "Year" & Format$(ColumnCounter)
Next
For ColumnCounter = 1 To 3
    For RowCounter = 1 To 3
        MyTable.CellLayout (RowCounter,ColumnCounter) .GotoLayout
        .Text.InsertText Format$(RowCounter * ColumnCounter)
    Next
Next
For ColumnCounter = 1 To 3
    MyTable.CellLayout (4,ColumnCounter) .GotoLayout
    .SmartSumColumn
Next
For RowCounter = 1 To 4
    MyTable.CellLayout (RowCounter,4) .GotoLayout
    .SmartSumRow
Next
```

'Example: SortLevel1 property

'This example creates a text list and sorts it in ascending alphanumeric
'order based on the first field.

```
.NewDocument
.Text.InsertText "Manufacturing, 555-1234", True
.Text.InsertText "Distribution, 555-6845", True
.Text.InsertText "Sales, 555-9425", True
.ActiveDocument.SortOptions.SortLevel1.FieldNumber = 1
.ActiveDocument.SortOptions.SortLevel1.SortType = $LwpSortTypeAlphanumeric
.ActiveDocument.SortOptions.SortLevel1.SortOrder = $LtsSortAscending
.ActiveDocument.SortOptions.NumParagraphs = 1
.ActiveDocument.SortOptions.SortNumbers = $LwpSortNumberOrderLast
.ActiveDocument.SortOptions.FieldDelimiter = $LwpDelimiterTypeTextdelimited
.ActiveDocument.SortOptions.FieldDelimiterText = ","
.SortParagraphs
```

'Example: SortLevel2 property

'This example creates a list and sorts it by the department field, then

'by the last name of the name field.

```
.NewDocument
```

```
.Text.InsertText "Manufacturing, George Roberts, 555-1234", True
```

```
.Text.InsertText "Distribution, Pete Richards, 555-6845", True
```

```
.Text.InsertText "Manufacturing, Albert Dunne, 555-9678", True
```

```
.Text.InsertText "Sales, Mary Hudson, 555-9425", True
```

```
.ActiveDocument.SortOptions.SortLevel2.FieldNumber = 2
```

```
.ActiveDocument.SortOptions.SortLevel2.SortWordOption = $LwpSortWhichWordLastword
```

```
.ActiveDocument.SortOptions.FieldDelimiter = $LwpDelimiterTypeTextdelimited
```

```
.ActiveDocument.SortOptions.FieldDelimiterText = ","
```

```
.SortParagraphs
```

```

'Example: SortParagraphs method
' This example inserts 5 rows and 5 columns of text into the current document
' where each column is tab delimited and then pause to display a message box.
' When you click OK, the message box closes, the sort options are set, and the
' grid is sorted in descending order relative to the first column.
' RUNTIME DEPENDENCIES: You must have a document open for this script to work.

'Insert tab-delimited text grid
Dim RowCount as Integer
Dim ColumnCount as Integer
Dim OutputText as string
For RowCount = 1 To 5
    For ColumnCount = 1 To 5
        OutputText = "r" & RowCount & "c" & ColumnCount
        .Text.InsertText OutputText
        .Text.InsertTab
    Next
    .Text.SplitParagraph
Next
.Text.Shade $LwpLocationTypeParagraph,$LwpNavigateDirectionLeft,5

'Set sort options
MessageBox "Click OK to sort selected text.",MB_OK,"Example Script"
.ActiveDocument.SortOptions.SortLevel1.FieldNumber = 1
.ActiveDocument.SortOptions.SortLevel2.SortWordOption = LwpSortWhichWordAllwords
.ActiveDocument.SortOptions.FieldDelimiter = $LwpDelimiterTypeTabdelimited
.ActiveDocument.SortOptions.SortLevel1.SortOrder = $LtsSortDescending
.ActiveDocument.SortOptions.SortLevel1.SortType = $LwpSortTypeAlphanumeric

'Sort paragraphs
.Text.SortParagraphs

```

'Example: SortWordOption property

'This example creates a list and sorts it by the department field, then

'by the last name of the name field.

```
.NewDocument
```

```
.Text.InsertText "Manufacturing, George Roberts, 555-1234", True
```

```
.Text.InsertText "Distribution, Pete Richards, 555-6845", True
```

```
.Text.InsertText "Manufacturing, Albert Dunne, 555-9678", True
```

```
.Text.InsertText "Sales, Mary Hudson, 555-9425", True
```

```
.ActiveDocument.SortOptions.SortLevel2.FieldNumber = 2
```

```
.ActiveDocument.SortOptions.SortLevel2.SortWordOption = $LwpSortWhichWordLastword
```

```
.ActiveDocument.SortOptions.FieldDelimiter = $LwpDelimiterTypeTextdelimited
```

```
.ActiveDocument.SortOptions.FieldDelimiterText = ","
```

```
.SortParagraphs
```


'Example: SpacesBetweenSentences property

'This example sets grammar check to flag sentences which do not have
'two spaces between them.

.ApplicationWindow.UserInterfacePrefs.GrammarOptions.**SpacesBetweenSentences** = 2

```
'Example: SpellAddToUserDict method
' This example prompts for a word to be added to the user dictionary. The word
' is inserted into the current document, selected and then added to the user
' dictionary.
' RUNTIME DEPENDENCIES: You must have a document open for this script to work.

Dim NewWord as String
NewWord = InputBox ("Enter a word to add to the user dictionary:", "Example Script", "")
If NewWord <> "" Then
    .Type NewWord
    .Text.MoveToStart $LwpLocationTypeWord
    .SelectWord
    .SpellAddToUserDict
    MessageBox NewWord & " was added to the user dictionary.", MB_OK, "Example Script"
End If
```

```
'Example: SpellMarkSkippedWords method  
' This example marks the selected word as skipped.  
' RUNTIME DEPENDENCIES: You must have a document open and a word selected  
' for this script to work.
```

```
.SpellMarkSkippedWords
```

'Example: SpellSkipAll method

' This example inserts a word, and selects it and then

' adds the selected word to Spell Check's skip all list.

' RUNTIME DEPENDENCIES: You must have a document open for this script to work.

.Type "Osborne"

.SelectWord

.SpellSkipAll

'Example: LineLocation property

'This example places lines around the page, then sets the location of the
'lines to the middle of the margins, and the offset to 1/2 inch.

```
.Page.Layout.RightPage.BorderLines.LinePlacement = &Hf  
.Page.Layout.RightPage.BorderLines.AllBorders.Pattern = $LwpBorderPatternWavy  
.Page.Layout.RightPage.BorderLines.AllBorders.WidthInTwips = 80  
.Page.Layout.RightPage.BorderOffset = 720  
.Page.Layout.RightPage.LineLocation = 1
```

'Example: LineNumberOptions property

'This example creates a new document, and sets line numbering to number every other line, with a spacing of 1/4 inch per line.

.newdocument

Set LineNumbers = .Division.**LineNumberOptions**

LineNumbers.NumberWhichLines = \$LwpLineNumberOptsSpecifiedLines

LineNumbers.NumberEveryNthLine = 2

LineNumbers.LinesSpacedEveryNthUnit = 360 ' 1/4 inch

'Example: LinePlacement property

'This example places lines around all sides of the page.

```
%Include "WPBITMSK.LSS" ' lets us use bitmask constants for LinePlacement
.Page.Layout.RightPage.BorderLines.LinePlacement = LwpLinePlacementAllsides
.Page.Layout.RightPage.BorderLines.AllBorders.Pattern = $LwpBorderPatternWavy
.Page.Layout.RightPage.BorderLines.AllBorders.WidthInTwips = 80
```

'Example: LinesSpacedEveryNthUnit property
'This example creates a new document, and sets line numbering to number
'every other line, with a spacing of 1/4 inch per line.

```
.newdocument  
Set LineNumbers = .Division.LineNumberOptions  
LineNumbers.NumberWhichLines = $LwpLineNumberOptsSpecifiedLines  
LineNumbers.NumberEveryNthLine = 2  
LineNumbers.LinesSpacedEveryNthUnit = 360 ' 1/4 inch
```


'Example: Lines property

'This example reads the current setting for lines around frame captions

'and allows the user to change it.

```
Dim CurrentLines As Integer, NewLines As Integer
```

```
CurrentLines = .ActiveDocument.FrameCaptionOptions.Lines
```

```
Do
```

```
    NewLines = Inputbox("Type a new value (0-6) for the default lines around caption  
option", "Example Script", CurrentLines)
```

```
Loop Until NewLines >= 0 And NewLines <= 6
```

```
.ActiveDocument.FrameCaptionOptions.Lines = NewLines
```

```
'Example: Link method
' This example saves the current division as an external file called LINKFILE.LWP.
' The current document becomes a master document.
' RUNTIME DEPENDENCIES: You must have a document open for this script to work.
```

```
Dim DivId as String
Dim LinkFile as String
```

```
LinkFile = .ActiveDocument.Path & "\LINKFILE.LWP"
DivId = .Division.Name
```

```
.ActiveDocument.Divisions(DivId).Link LinkFile, "Lotus Word Pro"
```

'Example: LoadFilesMaximized property

'This example asks the user whether to maximize document windows when they
'are displayed, and then sets the appropriate option.

```
stat = MessageBox ("Do you want to load documents maximized?", 36, "Example Script")
```

```
If stat = 6 Then ' user said yes
```

```
    .ApplicationWindow.UserInterfacePrefs.LoadFilesMaximized = True
```

```
Else
```

```
    .ApplicationWindow.UserInterfacePrefs.LoadFilesMaximized = False
```

```
End If
```

'Example: LowerCase method

' This example toggles the lowercase attribute of the selected text, displays
' a message box, then toggles the lowercase again.

' RUNTIME DEPENDENCIES: You must have a document open and some text selected
' for this script to work.

.LowerCase

MessageBox "Click OK to reverse the lowercase change.",MB_OK,"Example Script"

.LowerCase

```
'Example: MakeTableFromText method
' This example inserts 5 rows and 5 columns of text into the current document
' where each column is seperated by a tab and each row by a paragraph break.
' The text is then selected and converted into a table.
' RUNTIME DEPENDENCIES: You must have a document open for this script to work.
```

```
Dim RowCount as Integer
Dim ColumnCount as Integer
Dim OutputText as string
```

```
'Insert text in 5 tabbed rows
For RowCount = 1 To 5
  For ColumnCount = 1 To 5
    OutputText = "r" & RowCount & "c" & ColumnCount
    .Text.InsertText OutputText
    .Text.InsertTab
  Next
  .Text.SplitParagraph
Next
```

```
'Select the text
.Text.Shade $LwpLocationTypeParagraph,$LwpNavigateDirectionLeft,5
```

```
'Convert to table
.MakeTableFromText
```

'Example: MakeUniqueLinkName method

' This example prints a unique DDE link name to the Lotus Script Output panel.

' RUNTIME DEPENDENCIES: You must have a document open for this script to work.

Print .Division.DdeLinkManager.MakeUniqueLinkName("DDELink")

'Example: ManualFrame method

' This example displays the Create Frame pointer so you can create a frame manually.

' RUNTIME DEPENDENCIES: You must have a document open for this script to work.

.ManualFrame

'Example: ManualTable method

' This example displays the Create Table pointer so you can create a table manually.

' RUNTIME DEPENDENCIES: You must have a document open for this script to work.

.ManualTable


```
'Example: MarginBetween property
'This example inserts some text into the current document, then places
'a red line between the paragraphs. The spacing between the line and the
'text is set to 144 twips, or 1/10th of an inch

' Insert some text, then select it.
.Text.InsertText "This is an example paragraph.", True
.Text.InsertText "This is another example paragraph.", True
.Text.Backward $LwpNavigateObjectTypeParagraph , 2
' Set up the borders
.Text.ParagraphBorder.Betweenlines.LinePlacement = &Hf
.Text.ParagraphBorder.Betweenlines.AllBorders.Pattern = $LtsBorderPatternSolid
.Text.ParagraphBorder.Betweenlines.AllBorders.WidthInTwips = 20 ' 1 point line
' make the line red
.Text.ParagraphBorder.Betweenlines.AllBorders.Color.Red = 255
.Text.ParagraphBorder.Betweenlines.AllBorders.Color.Blue = 0
.Text.ParagraphBorder.Betweenlines.AllBorders.Color.Green = 0
.Text.ParagraphBorder.Betweenlines.AllBorders.Color.Override = $LwpColorOverrideRgb
'Put 1/10 inch between the text and the line
.Text.ParagraphBorder.MarginBetween = 144
'Make the line as long as the paragraph
.Text.ParagraphBorder.TypeBetween = $LwpParaBorderWidthText
```

'Example: MarginBottom property

'This example creates a new document, and then changes the bottom margin.

.NewDocument

.Layout.**MarginBottom** = 2880 ' 2 inches

```
'Example: MarginColor property
'This example script prints the red, green, and blue values of the Word Pro
'Margin color to the LotusScript output panel.
Print .AppViewPrefs.MarginColor.Red
Print .AppViewPrefs.MarginColor.Green
Print .AppViewPrefs.MarginColor.Blue
```

'Example: MarginLeft property

'This example creates a new document, and then changes the left margin.

.NewDocument

.Layout.**MarginLeft** = 2880 ' 2 inches

'Example: MarginRight property

'This example creates a new document, and then changes the header margins.

.NewDocument

.Layout.Header.IsMarginSameAsParent = False

.Layout.Header.MarginLeft = 2880 ' 2 inches

.Layout.Header.**MarginRight** = 720 ' 1/2 inch

'Example: MarginTop property

'This example creates a new document, and then changes the top page margin.

.NewDocument

.Layout.**MarginTop** = 2880 ' 2 inches

'Example: MarkCharacter property

'This example sets the option for indicating revision marked lines to a character to the right of the text, and sets the marking character to the exclamation point.

```
.ActiveDocument.DocOptions.MarkType = $LwpMarkTypeChar
```

'the value for MarkCharacter is the numeric representation of the character

```
.ActiveDocument.DocOptions.MarkCharacter = Asc("!")
```

```
.ActiveDocument.DocOptions.MarkPosition = $LwpMarkPositionRight
```

'Example: MarkPosition property

'This example sets the option for indicating revision marked lines to a
'character to the right of the text, and sets the marking character to
'the exclamation point.

```
.ActiveDocument.DocOptions.MarkType = $LwpMarkTypeChar
```

'the value for MarkCharacter is the numeric representation of the character

```
.ActiveDocument.DocOptions.MarkCharacter = Asc("!")
```

```
.ActiveDocument.DocOptions.MarkPosition = $LwpMarkPositionRight
```


'Example: MarkRevisionInsert method

'This example inserts some text into the current document,

'then marks it as revision inserted.

```
.Text.InsertText "This text will be marked as revision inserted. "
```

```
. Select $LwpSelectObjectTypeParagraph
```

```
.MarkRevisionInsert
```

'Example: MarkType property

'This example sets the option for indicating revision marked lines to a
'character to the right of the text, and sets the marking character to
'the exclamation point.

```
.ActiveDocument.DocOptions.MarkType = $LwpMarkTypeChar
```

'the value for MarkCharacter is the numeric representation of the character

```
.ActiveDocument.DocOptions.MarkCharacter = Asc("!")
```

```
.ActiveDocument.DocOptions.MarkPosition = $LwpMarkPositionRight
```

'Example: Mark method

' This example creates a bookmark, adds some text to it and then expands it to cover the inserted text.

' RUNTIME DEPENDENCIES: You must have a document open for this script to work.

Dim MarkerName as String

MarkerName = .Text.Mark(\$LwpMarkerTypeBookmark)

.Division.BookmarkManager.AddBookmark "NewBookMark" , MarkerName

.Text.InsertText "This is a new bookmark"

.Text.Mark \$LwpMarkerTypeBookmark,MarkerName,\$LwpRangePartEnd

'Example: MaxHyphLines property

'This example script sets the maximum number of consecutive hyphenated
'lines in a row to two.

.Division.DivisionOptions.HyphenationOptions.**MaxHyphLines** = 2

'Example: MaxIdenticalConsecSentOpens property

'This example sets the number of identical sentence openers used in grammar

'checking to three in a row, or five within ten consecutive sentences.

.ApplicationWindow.UserInterfacePrefs.GrammarOptions.**MaxIdenticalConsecSentOpens** = 3

.ApplicationWindow.UserInterfacePrefs.GrammarOptions.MaxIdenticalSentOpensWithin10 = 5

'Example: MaxIdenticalSentOpensWithin10 property

'This example sets the number of identical sentence openers used in grammar

'checking to three in a row, or five within ten consecutive sentences.

.ApplicationWindow.UserInterfacePrefs.GrammarOptions.MaxIdenticalConsecSentOpens = 3

.ApplicationWindow.UserInterfacePrefs.GrammarOptions.**MaxIdenticalSentOpensWithin10** = 5

```
'Example: MaximizeOnStartUp property
'This example asks the user whether to maximize the Word Pro window when
'starting the program, and then sets the appropriate option.

stat = MessageBox ("Do you want to maximize the Word Pro window?", 36, "Example
Script")
If stat = 6 Then ' user said yes
    .ApplicationWindow.UserInterfacePrefs.MaximizeOnStartUp = True
Else
    .ApplicationWindow.UserInterfacePrefs.MaximizeOnStartUp = False
End If
```

'Example: Maximize method

' This example maximizes the main application window.

.ApplicationWindow.Maximize

'Example: MaximumWordsinaSentence property

'This example sets the maximum number of words in a sentence

'before considering the sentence in error to 30 words.

.ApplicationWindow.UserInterfacePrefs.GrammarOptions.**MaximumWordsinaSentence** = 30

```
'Example: MergeSetDataFile method
' This example assigns the merge data file 'DATAFILE.LWP' to the currently
' active document.
' RUNTIME DEPENDENCIES: You must have a document open and a file named
' 'DATAFILE.LWP' located in the Word Pro default documents directory for this
' script to work.

Dim DataFile As String
Dim Status As Integer

DataFile = .ApplicationWindow.UserInterfacePrefs.DocPath
DataFile = DataFile & "\DATAFILE.LWP"

.ApplicationWindow.ActiveDocument.MergeOptions.MergeStepNumber = $LwpMergeStepNumber1

Status = .MergeSetDataFile (DataFile, 0)
If Status = False Then
    MsgBox "Unable to set up the data file!", MB_OK, "Example Script"
End If
```

```
'Example: MergeStart method
' This example merges data for the current merge document.
' RUNTIME DEPENDENCIES: You must have a document open which has been assigned
' to a merge data file has inserted merge fields for this script to work.

Dim stat As Integer

' Set up to merge and view
.ApplicationWindow.ActiveDocument.MergeOptions.Options = &H2
.ApplicationWindow.ActiveDocument.MergeOptions.MergeStepNumber = $LwpMergeStepNumber3

.MergeStart

Do
    stat = .Merge($LwpMergeActionNextRecord)
    .Merge $LwpMergeActionMergeOne
Loop Until stat = False

.Merge $LwpMergeActionClose
.CloseMergeDataFile
```

Word Pro: MaxContentWidth property

```
{button ,AL(^H_BASECONTAINER_CLASS;H_CELLCONTAINER_CLASS;H_FRAMECONTAINER_CLASS;H_NOTECONTAINER_CLASS;H_RUBYCONTAINER_CLASS;H_DROPDOWNCONTAINER_CLASS;H_PAGECONTAINER_CLASS;H_SUBPAGECONTAINER_CLASS;H_SUPERPAGECONTAINER_CLASS;H_ROWCONTAINER_CLASS;H_SUPERTABLECONTAINER_CLASS;H_TABLECONTAINER_CLASS;H_PARALLELCOLSCONTAINER_CLASS;H_TABLEONLYCONT_CLASS',0)} See list of classes
```

(Read-only) Returns the maximum width to which the content of a container can expand and rotate.

Data Type

Long

Syntax

maxcontentwidthvalue = [objectreference].MaxContentWidth

Legal values

Data type is Long but the unit of measurement used for this property is Twips. There are 1440 Twips per inch.

Usage

Word Pro: MaxHorzPaneDistance property

{button ,AL(^H_DOCWINDOW_CLASS',0)} [See list of classes](#)

(Read-write)

Data Type

Long

Syntax

maxhorzpanedistancevalue = [objectreference].MaxHorzPaneDistance

objectreference].MaxHorzPaneDistance = maxhorzpanedistancevalue

Legal values

Data type is Long but the unit of measurement used for this property is Twips. There are 1440 Twips per inch.

Usage

Word Pro: MaxHyphLines property

{button ,AL('H_HYPHENATIONOPTIONS_CLASS',0)} [See list of classes](#)

{button ,AL('H_MAXHYPHLINES_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Represents the maximum number of hyphenated lines in a document.

Data Type

[Integer](#)

Syntax

maxhyphlinesvalue = [objectreference].MaxHyphLines

[objectreference].MaxHyphLines = maxhyphlinesvalue

Legal values

The legal values for this property are:

<u>Value</u>	<u>Effect</u>
0	Represents an unlimited number of hyphenated lines.
1	Represents a one-line maximum.
2	Represents a two-line maximum.

Usage

Equivalent to choosing File - Document Properties - Document, clicking the Options tab, and selecting an option in "Consecutive hyphenated lines."

Word Pro: MaxIdenticalConsecSentOpens property

{button ,AL('H_GRAMMAR_CLASS',0)} [See list of classes](#)

{button ,AL('H_MAXIDENTICALCONSECSSENTOPENS_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Indicates how many identical openers for consecutive sentences are flagged by Grammar Check in a document.

Data Type

[Integer](#)

Syntax

maxidenticalconsecsentopensvalue = [objectreference].MaxIdenticalConsecSentOpens

[objectreference].MaxIdenticalConsecSentOpens = maxidenticalconsecsentopensvalue

Legal values

The legal values for this property range from 0 (never flag consecutive sentence openers) to 9 (flag 9 or more consecutive sentence openers). Default is 3.

Usage

Sets a maximum number of identical sentence openers for consecutive sentences in Grammar Check. Choose from 0 to 9. Equivalent to choosing Edit - Check Grammar, clicking Options, and entering a number in the Maximum number of identical sentence openers section, "For consecutive sentences" field, on the Grammatical Style panel.

Word Pro: MaxIdenticalSentOpensWithin10 property

{button ,AL('H_GRAMMAR_CLASS',0)} [See list of classes](#)

{button ,AL('H_MAXIDENTICALSENTOPENSWITHIN10_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Indicates how many identical sentence openers, within 10 sentences, are flagged by Grammar Check in a document.

Data Type

[Integer](#)

Syntax

maxidenticalsentopenswithin10value = [objectreference].MaxIdenticalSentOpensWithin10

[objectreference].MaxIdenticalSentOpensWithin10 = maxidenticalsentopenswithin10value

Legal values

The legal values for this property range from 0 (never flag consecutive sentence openers) to 9 (flag 9 or more consecutive sentence openers within 10 sentences). Default is 3.

Usage

Sets a maximum number of identical sentence openers, within 10 sentences, in Grammar Check. Choose from 0 to 9. Equivalent to choosing Edit - Check Grammar, clicking Options, and entering a number in the Maximum number of identical sentence openers section, "Within 10 sentences" field, on the Grammatical Style panel.

Word Pro: MaximizeOnStartUp property

{button ,AL('H_USERINTERFACEPREFS_CLASS',0)} [See list of classes](#)

{button ,AL('H_MAXIMIZEONSTARTUP_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Indicates whether or not the Word Pro application window is maximized on startup.

Data Type

[Integer \(Bool\)](#)

Syntax

maximizeonstartupvalue = [objectreference].MaximizeOnStartUp

[objectreference].MaximizeOnStartUp = maximizeonstartupvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Equivalent to the "Load Word Pro maximized" option on the General panel of the Word Pro Preferences dialog box. If this property is set to True (-1), Word Pro loads maximized. If set to False (0), Word Pro does not load maximized.

Word Pro: MaximumWordsinaSentence property

{button ,AL('H_GRAMMAR_CLASS',0)} [See list of classes](#)

{button ,AL('H_MAXIMUMWORDSINASENTENCE_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Indicates a maximum number of words that will be allowed in a sentence before Grammar Check alerts you .

Data Type

[Integer](#)

Syntax

maximumwordsinasentencevalue = [objectreference].MaximumWordsinaSentence

[objectreference].MaximumWordsinaSentence = maximumwordsinasentencevalue

Legal values

The legal values for this property range from 25 to 75. Default is 35.

Usage

Sets a maximum number of words in a sentence, from 25 to 75. Equivalent to choosing Edit - Check Grammar, clicking Options, and entering a number in the "Maximum number of words per sentence" field on the Grammatical Style panel.

Word Pro: MaxLeftBorder property

{button ,AL(^H_BASETABLE_CLASS;H_FOOTNOTETABLE_CLASS;H_PARALLELCOLUMNS_CLASS;H_GLOSSARY_CLASS;H_TABLEHEADING_CLASS;H_TABLE_CLASS',0)} [See list of classes](#)

(Read-only) Returns the specific maximum width that can be set for the left border of a row in a table object.

Data Type

Long

Syntax

maxleftbordervalue = [objectreference].MaxLeftBorder

Legal values

Data type is Long but the unit of measurement used for this property is Twips. There are 1440 Twips per inch.

Usage

Word Pro: MaxLeftGutter property

{button ,AL(^H_BASETABLE_CLASS;H_FOOTNOTETABLE_CLASS;H_PARALLELCOLUMNS_CLASS;H_GLOSSARY_CLASS;H_TABLEHEADING_CLASS;H_TABLE_CLASS',0)} [See list of classes](#)

(Read-only) Returns the specific maximum width that can be set for the left gutter of a row in a table object.

Data Type

Long

Syntax

maxleftguttervalue = [objectreference].MaxLeftGutter

Legal values

Data type is Long but the unit of measurement used for this property is Twips. There are 1440 Twips per inch.

Usage

Word Pro: MaxNumColsAllowed property

{button ,AL(^H_BASSETABLE_CLASS;H_FOOTNOTETABLE_CLASS;H_PARALLELCOLUMNS_CLASS;H_GLOSSARY_CLASS;H_TABLEHEADING_CLASS;H_TABLE_CLASS',0)} [See list of classes](#)

(Read-only) Returns the maximum number of columns that can be inserted without widening the table container. Word Pro uses the default column width to determine the number of columns inserted.

Data Type

Integer

Syntax

maxnumcolsallowedvalue = [objectreference].MaxNumColsAllowed

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Word Pro: MaxNumRowsAllowed property

{button ,AL(^H_BASETABLE_CLASS;H_FOOTNOTETABLE_CLASS;H_PARALLELCOLUMNS_CLASS;H_GLOSSARY_CLASS;H_TABLEHEADING_CLASS;H_TABLE_CLASS',0)} [See list of classes](#)

(Read-only) Returns the maximum number of rows that can be inserted without enlarging the table container.

Data Type

Integer

Syntax

maxnumrowsallowedvalue = [objectreference].MaxNumRowsAllowed

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Word Pro: MaxRightBorder property

{button ,AL(^H_BASETABLE_CLASS;H_FOOTNOTETABLE_CLASS;H_PARALLELCOLUMNS_CLASS;H_GLOSSARY_CLASS;H_TABLEHEADING_CLASS;H_TABLE_CLASS',0)} [See list of classes](#)

(Read-only) Returns the specific maximum width that can be set for the right border of a row in a table object.

Data Type

Long

Syntax

maxrightbordervalue = [objectreference].MaxRightBorder

Legal values

Data type is Long but the unit of measurement used for this property is Twips. There are 1440 Twips per inch.

Usage

Word Pro: MaxRightGutter property

{button ,AL(^H_BASETABLE_CLASS;H_FOOTNOTETABLE_CLASS;H_PARALLELCOLUMNS_CLASS;H_GLOSSARY_CLASS;H_TABLEHEADING_CLASS;H_TABLE_CLASS',0)} [See list of classes](#)

(Read-only) Returns the specific maximum width that can be set for the right gutter of a row in a table object.

Data Type

Long

Syntax

maxrightguttervalue = [objectreference].MaxRightGutter

Legal values

Data type is Long but the unit of measurement used for this property is Twips. There are 1440 Twips per inch.

Usage

Word Pro: MaxSplitCols property

{button ,AL(^H_BASSETABLE_CLASS;H_FOOTNOTETABLE_CLASS;H_PARALLELCOLUMNS_CLASS;H_GLOSSARY_CLASS;H_TABLEHEADING_CLASS;H_TABLE_CLASS',0)} [See list of classes](#)

(Read-only) Returns the maximum number of columns into which a table cell can be split.

Data Type

Integer

Syntax

maxsplitcolsvalue = [objectreference].MaxSplitCols

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

When table cells are split, Word Pro inserts the number of columns you specify into the current cell, and places the contents of the cell into the upper left cell of the newly created cells.

Word Pro: MaxSplitRows property

{button ,AL(^H_BASETABLE_CLASS;H_FOOTNOTETABLE_CLASS;H_PARALLELCOLUMNS_CLASS;H_GLOSSARY_CLASS;H_TABLEHEADING_CLASS;H_TABLE_CLASS',0)} [See list of classes](#)

(Read-only) Returns the maximum number of rows into which a table cell can be split.

Data Type

Integer

Syntax

maxsplitrowsvalue = [objectreference].MaxSplitRows

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

When table cells are split, Word Pro inserts the number of rows you specify into the current cell, and places the contents of the cell into the upper left cell of the newly created cells.

Word Pro: MaxTopBorder property

{button ,AL(^H_BASETABLE_CLASS;H_FOOTNOTETABLE_CLASS;H_PARALLELCOLUMNS_CLASS;H_GLOSSARY_CLASS;H_TABLEHEADING_CLASS;H_TABLE_CLASS',0)} [See list of classes](#)

(Read-only) Returns the specific maximum width that can be set for the top border of a row in a table object.

Data Type

Long

Syntax

maxtopbordervalue = [objectreference].MaxTopBorder

Legal values

Data type is Long but the unit of measurement used for this property is Twips. There are 1440 Twips per inch.

Usage

Word Pro: MaxTopGutter property

{button ,AL(^H_BASETABLE_CLASS;H_FOOTNOTETABLE_CLASS;H_PARALLELCOLUMNS_CLASS;H_GLOSSARY_CLASS;H_TABLEHEADING_CLASS;H_TABLE_CLASS',0)} [See list of classes](#)

(Read-only) Returns the specific maximum width that can be set for the top gutter of a row in a table object.

Data Type

Long

Syntax

maxtopguttervalue = [objectreference].MaxTopGutter

Legal values

Data type is Long but the unit of measurement used for this property is Twips. There are 1440 Twips per inch.

Usage

Word Pro: MaxVertPaneDistance property

{button ,AL(^H_DOCWINDOW_CLASS',0)} [See list of classes](#)

(Read-write)

Data Type

Long

Syntax

maxvertpanedistancevalue = [objectreference].MaxVertPaneDistance

[objectreference].MaxVertPaneDistance = maxvertpanedistancevalue

Legal values

Data type is Long but the unit of measurement used for this property is Twips. There are 1440 Twips per inch.

Usage

Word Pro: MenuPaths property

{button ,AL(`H_USERINTERFACEPREFS_CLASS',0)} [See list of classes](#)

{button ,AL(`H_MENUPATHS_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-only)

This language element has not yet been defined. Lotus provides updated Help files on its web site.

Data Type

StringCollection

Syntax

menupathsvalue = [objectreference].MenuPaths

Legal values

Always contains an instance of the StringCollection class.

Usage

Word Pro: MenuPath property

{button ,AL('H_USERINTERFACEPREFS_CLASS',0)} [See list of classes](#)

{button ,AL('H_MENUPATH_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write)

This language element has not yet been defined. You can check for a newer Help file on the Lotus web site.

Data Type

String

Syntax

menupathvalue = [objectreference].MenuPath

[objectreference].MenuPath = menupathvalue

Legal values

Any value of type String.

Usage

Word Pro: MergeFileType property

{button ,AL(^H_MERGEOPTIONS_CLASS',0)} [See list of classes](#)

(Read-write)

Data Type

The data type for this property is [Variant](#) which allows the value of this property to be one of the constants listed below or its numeric equivalent (in parentheses).

Syntax

[objectreference].MergeFileType = mergefiletypevalue

mergefiletypevalue = [objectreference].MergeFileType

Legal values

<u>Value</u>	<u>Effect</u>
\$LwpMergeTypeDefault (2258)	
\$LwpMergeTypeEnvelope (2260)	
\$LwpMergeTypeLabel (2261)	
\$LwpMergeTypeLetter (2259)	

Usage

Word Pro: MergeInfoPtr property

{button ,AL('H_MERGEOPTIONS_CLASS',0)} [See list of classes](#)

(Read-write) Do not write to this property.

Data Type

Long

Syntax

mergeinfoPtrvalue = [objectreference].MergeInfoPtr

[objectreference].MergeInfoPtr = mergeinfoPtrvalue

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Word Pro uses this property to access other Merge classes that are not available to users.

Word Pro: MergeOptions property

{button ,AL('H_TEXTDOCUMENT_CLASS',0)} [See list of classes](#)

(Read-only)

Data Type

[MergeOptions](#)

Syntax

mergeoptionsvalue = [objectreference].MergeOptions

Legal values

Always contains an instance of the MergeOptions class.

Usage

Word Pro: MergeStepNumber property

{button ,AL(^H_MERGEOPTIONS_CLASS',0)} [See list of classes](#)

(Read-write) An indicator which you can use to set the steps involved in the Merge process.

Data Type

Variant (Enumerated)

Syntax

mergestepnumbervalue = [objectreference].MergeStepNumber

[objectreference].MergeStepNumber = mergestepnumbervalue

Legal values

<u>Value</u>	<u>Effect</u>
\$LwpMergeStepNumber0 (1505)	The first step of a Merge (selecting a data file).
\$LwpMergeStepNumber1 (1506)	Inserts a merge field.
\$LwpMergeStepNumber2 (1507)	Sets a delimiter.
\$LwpMergeStepNumber3 (1508)	Prints the Merge data file.

Usage

Set these values before using the WMCommand in order to bring up the dialog box of your choice.

Word Pro: MergeToFile property

{button ,AL(^H_MERGEOPTIONS_CLASS',0)} [See list of classes](#)

(Read-write)

Data Type

String.

Syntax

[objectreference].MergeToFile = mergetofilevalue

mergetofilevalue = [objectreference].MergeToFile

Legal values**Usage**

Word Pro: MetafilePict property

{button ,AL(^H_GRAPHICOLEBJECT_CLASS;H_GRAPHIC_CLASS;H_OLEOBJECT_CLASS',0)} [See list of classes](#)

(Read-only)

Data Type

Long.

Syntax

metafilepictvalue = [objectreference].MetafilePict

Legal values**Usage**

Word Pro: MinBottomMargin property

```
{button ,AL(^H_LAYOUT_CLASS;H_CELLGROUPLAYOUT_CLASS;H_CELLLAYOUT_CLASS;H_CONNECTEDLAY  
OUT_CLASS;H_COLUMNGROUPLAYOUT_CLASS;H_FOOTERLAYOUT_CLASS;H_FRAMELAYOUT_CLASS;H  
_FRAMEGROUPLAYOUT_CLASS;H_DROP_CAPLAYOUT_CLASS;H_GROUPLAYOUT_CLASS;H_HEADERLAY  
OUT_CLASS;H_NOTELAYOUT_CLASS;H_PAGELAYOUT_CLASS;H_ROWGROUPLAYOUT_CLASS;H_ROWLA  
YOUT_CLASS;H_RUBY_LAYOUT_CLASS;H_SUPERTABLEGROUPLAYOUT_CLASS;H_SUPERTABLELAYOUT  
_CLASS;H_TABLELAYOUT_CLASS;H_ENDNOTELAYOUT_CLASS;H_FOOTNOTELAYOUT_CLASS;H_TABLE  
HEADINGLAYOUT_CLASS;H_TOCSUPERTABLELAYOUT_CLASS',0)} See list of classes
```

(Read-only) The minimum value to which a layout object's bottom margin can be set.

Data Type

Long

Syntax

minbottommarginvalue = [objectreference].MinBottomMargin

Legal values

Data type is Long but the unit of measurement used for this property is Twips. There are 1440 Twips per inch.

Usage

Word Pro: MinHeight property

{button ,AL(^H_LAYOUT_CLASS;H_CELLGROUPLAYOUT_CLASS;H_CELLLAYOUT_CLASS;H_CONNECTEDLAYOUT_CLASS;H_COLUMNGROUPLAYOUT_CLASS;H_FOOTERLAYOUT_CLASS;H_FRAMELAYOUT_CLASS;H_FRAMEGROUPLAYOUT_CLASS;H_DROP_CAPLAYOUT_CLASS;H_GROUPLAYOUT_CLASS;H_HEADERLAYOUT_CLASS;H_NOTELAYOUT_CLASS;H_PAGELAYOUT_CLASS;H_ROWGROUPLAYOUT_CLASS;H_ROWLAYOUT_CLASS;H_RUBYLAYOUT_CLASS;H_SUPERTABLEGROUPLAYOUT_CLASS;H_SUPERTABLELAYOUT_CLASS;H_TABLELAYOUT_CLASS;H_ENDNOTELAYOUT_CLASS;H_FOOTNOTELAYOUT_CLASS;H_TABLEHEADINGLAYOUT_CLASS;H_TOCSUPERTABLELAYOUT_CLASS',0)} [See list of classes](#)

(Read-write) The minimum value to which a layout object's height can be set.

Data Type

Long

Syntax

minheightvalue = [objectreference].MinHeight

[objectreference].MinHeight = minheightvalue

Legal values

Data type is Long but the unit of measurement used for this property is Twips. There are 1440 Twips per inch.

Usage

Controls the height of a layout object, so that if the content becomes smaller than a specific height, then the layout object does not become smaller than that specific height. The effect of setting this property can be seen in frames which have automatic sizing enabled.

Word Pro: MinLeftMargin property

```
{button ,AL('H_LAYOUT_CLASS;H_CELLGROUPLAYOUT_CLASS;H_CELLLAYOUT_CLASS;H_CONNECTEDLAY  
OUT_CLASS;H_COLUMNGROUPLAYOUT_CLASS;H_FOOTERLAYOUT_CLASS;H_FRAMELAYOUT_CLASS;H  
_FRAMEGROUPLAYOUT_CLASS;H_DROP_CAPLAYOUT_CLASS;H_GROUPLAYOUT_CLASS;H_HEADERLAY  
OUT_CLASS;H_NOTELAYOUT_CLASS;H_PAGELAYOUT_CLASS;H_ROWGROUPLAYOUT_CLASS;H_ROWLA  
YOUT_CLASS;H_RUBYLAYOUT_CLASS;H_SUPERTABLEGROUPLAYOUT_CLASS;H_SUPERTABLELAYOUT  
_CLASS;H_TABLELAYOUT_CLASS;H_ENDNOTELAYOUT_CLASS;H_FOOTNOTELAYOUT_CLASS;H_TABLE  
HEADINGLAYOUT_CLASS;H_TOCSUPERTABLELAYOUT_CLASS',0)} See list of classes
```

(Read-only) The minimum value to which the left margin can be set.

Data Type

Long

Syntax

minleftmarginvalue = [objectreference].MinLeftMargin

Legal values

Data type is Long but the unit of measurement used for this property is Twips. There are 1440 Twips per inch.

Usage

Word Pro: MinRightMargin property

{button ,AL('H_LAYOUT_CLASS;H_CELLGROUPLAYOUT_CLASS;H_CELLLAYOUT_CLASS;H_CONNECTEDLAYOUT_CLASS;H_COLUMNGROUPLAYOUT_CLASS;H_FOOTERLAYOUT_CLASS;H_FRAMELAYOUT_CLASS;H_FRAMEGROUPLAYOUT_CLASS;H_DROP_CAPLAYOUT_CLASS;H_GROUPLAYOUT_CLASS;H_HEADERLAYOUT_CLASS;H_NOTELAYOUT_CLASS;H_PAGELAYOUT_CLASS;H_ROWGROUPLAYOUT_CLASS;H_ROWLAYOUT_CLASS;H_RUBY_LAYOUT_CLASS;H_SUPERTABLEGROUPLAYOUT_CLASS;H_SUPERTABLELAYOUT_CLASS;H_TABLELAYOUT_CLASS;H_ENDNOTELAYOUT_CLASS;H_FOOTNOTELAYOUT_CLASS;H_TABLEHEADINGLAYOUT_CLASS;H_TOCSUPERTABLELAYOUT_CLASS',0)} [See list of classes](#)

(Read-only) The minimum value to which the right margin can be set.

Data Type

Long

Syntax

minrightmarginvalue = [objectreference].MinRightMargin

Legal values

Data type is Long but the unit of measurement used for this property is Twips. There are 1440 Twips per inch.

Usage

Word Pro: MinTopMargin property

```
{button ,AL('H_LAYOUT_CLASS;H_CELLGROUPLAYOUT_CLASS;H_CELLLAYOUT_CLASS;H_CONNECTEDLAY  
OUT_CLASS;H_COLUMNGROUPLAYOUT_CLASS;H_FOOTERLAYOUT_CLASS;H_FRAMELAYOUT_CLASS;H  
_FRAMEGROUPLAYOUT_CLASS;H_DROP_CAPLAYOUT_CLASS;H_GROUPLAYOUT_CLASS;H_HEADERLAY  
OUT_CLASS;H_NOTELAYOUT_CLASS;H_PAGELAYOUT_CLASS;H_ROWGROUPLAYOUT_CLASS;H_ROWLA  
YOUT_CLASS;H_RUBYLAYOUT_CLASS;H_SUPERTABLEGROUPLAYOUT_CLASS;H_SUPERTABLELAYOUT  
_CLASS;H_TABLELAYOUT_CLASS;H_ENDNOTELAYOUT_CLASS;H_FOOTNOTELAYOUT_CLASS;H_TABLE  
HEADINGLAYOUT_CLASS;H_TOCSUPERTABLELAYOUT_CLASS',0)} See list of classes
```

(Read-only) The minimum value to which the top margin can be set.

Data Type

Long

Syntax

mintopmarginvalue = [objectreference].MinTopMargin

Legal values

Data type is Long but the unit of measurement used for this property is Twips. There are 1440 Twips per inch.

Usage

Word Pro: MirrorName property

{button ,AL('H_CLICKHERE_CLASS;',0)} [See list of classes](#)

{button ,AL('H_MIRRORNAME_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) The name of the Click Here Block to be repeated.

Data Type

String.

Syntax

mirrornamevalue = [objectreference].MirrorName

[objectreference].MirrorName = mirrornamevalue

Legal values

Any valid Click Here Block name.

Usage

Repeating Click Here Blocks let you fill in information once and repeat it in multiple locations throughout the document.

If this property is blank, the Click Here Block is not mirrored or repeating.

If there is a valid Click Here Block name in this property, then the Click Here Block inherits its properties and data from the Click Here Block with that name.

Word Pro: ModifiedDateString property

{button ,AL('H_DOCINFO_CLASS;H_VERSION_CLASS',0)} [See list of classes](#)

{button ,AL('H_MODIFIEDDATESTRING_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-only) Returns the date the file was last changed as a String.

Data Type

String

Syntax

modifieddatestringvalue = [objectreference].ModifiedDateString

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

The format of the CreationDateString property is determined by the Windows Short date style.

Word Pro: ModifiedTimeString property

{button ,AL('H_DOCINFO_CLASS;H_VERSION_CLASS',0)} [See list of classes](#)

{button ,AL('H_MODIFIEDTIMESTRING_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-only) Returns the time the file was last changed as a String.

Data Type

[String](#)

Syntax

modifiedtimestringvalue = [objectreference].ModifiedTimeString

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

The format of the ModifiedTimeString property is determined by the Windows Time style.

Word Pro: ModifiedTimeValue property

{button ,AL(`H_DOCINFO_CLASS',0)} [See list of classes](#)

{button ,AL(`H_MODIFIEDTIMEVALUE_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Returns the time the file was last changed as a Long data type.

Data Type

Long

Syntax

modifiedtimevaluevalue = [objectreference].ModifiedTimeValue

[objectreference].ModifiedTimeValue = modifiedtimevaluevalue

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

The return value represents the number of seconds that have elapsed since midnight on January 1, 1970.

{button ,AL(`H_MODIFIEDTIMESTRING_PROPERTY_MEMDEF',0)} [See related topics](#)

Word Pro: MultiCellPasteOn property

{button ,AL('H_PREFERENCES_CLASS',0)} [See list of classes](#)

{button ,AL('H_MULTICELLPASTEON_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write)

Data Type

[Integer](#)

Syntax

multicellpasteonvalue = [objectreference].MultiCellPasteOn

[objectreference].MultiCellPasteOn = multicellpasteonvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Word Pro: MultiCompareParaTagSet property

{button ,AL(^H_CLICKHERE_CLASS;H_TEXTMARKER_CLASS;H_TEXT_CLASS',0)} [See list of classes](#)

(Read-only)

Data Type

Integer

Syntax

multicompareparatagsetvalue = [objectreference].MultiCompareParaTagSet

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Word Pro: MultiCompareParaTag property

{button ,AL(^H_CLICKHERE_CLASS;H_TEXTMARKER_CLASS;H_TEXT_CLASS',0)} [See list of classes](#)

(Read-only)

Data Type

String

Syntax

multicompareparatagvalue = [objectreference].MultiCompareParaTag

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Word Pro: NameBasedOnStyle property

```
{button ,AL(^H_LAYOUT_CLASS;H_CELLGROUPLAYOUT_CLASS;H_CELLLAYOUT_CLASS;H_CONNECTEDLAY  
OUT_CLASS;H_COLUMNGROUPLAYOUT_CLASS;H_FOOTERLAYOUT_CLASS;H_FRAMELAYOUT_CLASS;H  
_FRAMEGROUPLAYOUT_CLASS;H_DROP_CAPLAYOUT_CLASS;H_GROUPLAYOUT_CLASS;H_HEADERLAY  
OUT_CLASS;H_NOTELAYOUT_CLASS;H_PAGELAYOUT_CLASS;H_ROWGROUPLAYOUT_CLASS;H_ROWLA  
YOUT_CLASS;H_RUBYLAYOUT_CLASS;H_SUPERTABLEGROUPLAYOUT_CLASS;H_SUPERTABLELAYOUT  
_CLASS;H_TABLELAYOUT_CLASS;H_ENDNOTELAYOUT_CLASS;H_FOOTNOTELAYOUT_CLASS;H_TABLE  
HEADINGLAYOUT_CLASS;H_TOCSUPERTABLELAYOUT_CLASS',0)} See list of classes
```

(Read-write) The name of the style from which the layout object was created.

Data Type

String

Syntax

namebasedonstylevalue = [objectreference].NameBasedOnStyle

[objectreference].NameBasedOnStyle = namebasedonstylevalue

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

When setting the NameBasedOnStyle property, be sure to use the exact upper and lower case for the layout style being assigned. For example, "Default Frame" is not equivalent to "default frame".

Word Pro: NameForFilters property

{button ,AL('H_NOTELAYOUT_CLASS',0)} [See list of classes](#)

(Read-write) The name of the user who created a specific comment note.

Data Type

String

Syntax

nameforfiltersvalue = [objectreference].NameForFilters

[objectreference].NameForFilters = nameforfiltersvalue

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Default value for this property is contained in the "User Name" setting in the Personal panel of the Word Pro Preferences dialog box. Modifying the NameForFilters property of a specific comment note does not change the name setting under Word Pro Preferences.

Word Pro: NameOfExternalBorder property

{button ,AL('H_BORDER_CLASS',0)} [See list of classes](#)

{button ,AL('H_NAMEOFEXTERNALBORDER_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) The internal name of a designer border on a page, frame, or table. The name used is not the name of the file that contains the external border; it is an internal name.

Data Type

String.

Syntax

nameofexternalbordervalue = [objectreference].NameOfExternalBorder

[objectreference].NameOfExternalBorder = nameofexternalbordervalue

Legal values

String.

Usage

The internal name assigned to a designer border that is chosen in the "Designer border" field on the Lines & Colors tab of the appropriate InfoBox.

Word Pro: Name property

{button ,AL('H_BASEOBJECT_CLASS',0)} [See list of classes](#)

(Read-write) The Name property contains the name of an object. This can be useful when trying to access an object from the object's collection. Some objects allow you to change their names. Other objects are named internally by Word Pro and do not allow you to change their names.

This property is inherited by all Word Pro objects, but not every object makes use of this property.

Data Type

String

Syntax

namevalue = [objectreference].Name

[objectreference].Name = namevalue

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Some of the objects which make use of this property include WPAApplication, TextDocument, Division, DivisionInfo, Frame, Table, ClickHere, and Bookmark. The information stored in this property and the proper use of that information is detailed below.

CurrentApplication.Name

The name of the application which, in this case, is always "Lotus Word Pro".

<TextDocumentobject>.Name

The name of the document represented by the TextDocument on which the Name property is located.

<Divisionobject>.Name

The internal division name (a hexadecimal value) which uniquely identifies a division.

<DivisionInfoobject>.Name

The external division name (a string value) which represents the name seen in the division tab.

<Layoutobject>.Name

Returns the name of a frame, a table, or other object that uses a layout, such as FrameLayout or TableLayout.

<ClickHereobject>.Name

The name of the ClickHere object from which you called the Name property. You must use the ClickHereCollection object to access ClickHeres.

<Bookmarkobject>.Name

The name of the Bookmark object from which you called the Name property. You must use the BookmarkCollection object in the BookmarksByMarkerName property of a division to access the bookmarks by name.

Word Pro: NavigationLink property

{button ,AL('H_HTMLOPTIONS_CLASS',0)} [See list of classes](#)

{button ,AL('H_NAVIGATIONLINK_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write)

This language element has not yet been defined. You can check for a newer Help file on the Lotus web site.

Data Type

Integer.

Syntax

navigationlinkvalue = [objectreference].NavigationLink

[objectreference].NavigationLink = navigationlinkvalue

Legal values

Integer. The values are:

<u>Value</u>	<u>Effect</u>
0	No link provided.
1	Link appears at the bottom of the page.
2	Link appears at the top of the page.
3	Link appears at both the top and bottom of the page.

Usage

Equivalent to choosing File - Internet - HTML Export Assistant and, on the Layout panel, selecting an option in the "Navigation arrows" box.

Word Pro: Negative property

{button ,AL('H_NUMERICFORMAT_CLASS',0)} [See list of classes](#)

(Read-only) Allows you to modify the "Negative numbers" condition of the current number format.

Data Type

[NumericFormatSubset](#)

Syntax

negativevalue = [objectreference].Negative

Legal values

Always contains an instance of the NumericFormatSubset class.

Usage

Equivalent to the "Negative numbers" condition for edit in the Edit Format dialog box. You can access this dialog box by clicking the Format Options button on the Number Format panel of the InfoBox for cell layout objects.

By accessing the "Negative numbers" condition of a number format, you can modify how negative values appear within table cells, and choose prefix or suffix text.

Word Pro: NewDocMacroName property

{button ,AL('H_AUTORUNMACRO_CLASS',0)} [See list of classes](#)

(Read-write)

Data Type

String

Syntax

newdocmacrovalue = [objectreference].NewDocMacroName

[objectreference].NewDocMacroName = newdocmacrovalue

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Word Pro: NewFile property

{button ,AL(^H_REVIEWVERSIONS_CLASS',0)} [See list of classes](#)

(Write-only)

Data Type

[Integer](#)

Syntax

[objectreference].NewFile = newfilevalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Word Pro: NewVersion property

{button ,AL('H_REVIEWVERSIONS_CLASS',0)} [See list of classes](#)

{button ,AL('H_NEWVERSION_PROPERTY_EXSCRIPT',1)} [See example](#)

(Write-only)

Data Type

[Integer](#)

Syntax

[objectreference].NewVersion = newversionvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Word Pro: NextClickHere property

{button ,AL(^H_CLICKHERE_CLASS',0)} [See list of classes](#)

(Read-only) The name of the ClickHere block which is next in the division (uses Tab order).

Data Type

String

Syntax

nextclickHerevalue = [objectreference].NextClickHere

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Word Pro: NextCursorableDivision property

{button ,AL(^H_DIVISION_CLASS;H_TEXTDOCUMENT_CLASS',0)} [See list of classes](#)

(Read-only)

Gives you the name of the next division you would come to if you were cursoring through.

Data Type

String

Syntax

nextcursorabledivisionvalue = [objectreference].NextCursorableDivision

Legal values

This property is read-only. The value of this property cannot be set by a script.

Usage

Returns the internal division name

Word Pro: NextName property

{button ,AL(^H_DIVISION_CLASS;H_TEXTDOCUMENT_CLASS',0)} [See list of classes](#)

(Read-only)

Data Type

String

Syntax

nextnamevalue = [objectreference].NextName

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Word Pro: NextNeighbor property

{button ,AL(^H_DIVISION_CLASS;H_TEXTDOCUMENT_CLASS',0)} [See list of classes](#)

(Read-only)

Data Type

String

Syntax

nextneighborvalue = [objectreference].NextNeighbor

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Word Pro: NextSection property

{button ,AL(^H_SECTION_CLASS;H_INDEXSECTION_CLASS',0)} [See list of classes](#)

(Read-only)

Data Type

String

Syntax

nextsectionvalue = [objectreference].NextSection

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Word Pro: NextStyleName property

{button ,AL('H_BREAKS_CLASS',0)} [See list of classes](#)

{button ,AL('H_NEXTSTYLENAME_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Determines the paragraph style to be used following the current paragraph.

Data Type

String

Syntax

nextstylevalue = [objectreference].NextStyleName

[objectreference].NextStyleName = nextstylevalue

Legal values

The legal values for this property are any valid paragraph style name in the document.

Usage

This property is equivalent to the "Style to use for next paragraph" list of paragraph styles in the Text InfoBox.

To enable or disable this option, use the UseNextStyle property.

{button ,AL('H_USENEXTSTYLE_PROPERTY_MEMDEF',0)} [See related topics](#)

Word Pro: NextText property

{button ,AL(^H_CLICKHERE_CLASS;H_TEXTMARKER_CLASS;H_TEXT_CLASS',0)} [See list of classes](#)

(Read-only)

Data Type

String

Syntax

nexttextvalue = [objectreference].NextText

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Word Pro: NoHyphenate property

{button ,AL('H_ATTRIBUTES_CLASS',0)} [See list of classes](#)

{button ,AL('H_NOHYPHENATE_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Enables or disables the No Hyphenation attribute.

Data Type

[Integer](#)

Syntax

nohyphenatevalue = [objectreference].NoHyphenate

[objectreference].NoHyphenate = nohyphenatevalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

For the attributes object in the Text class, this property is equivalent to the No Hyphenation option in the Font panel of the Text InfoBox.

For the attributes objects (SearchAttributes and ReplaceAttributes) in the FindAndReplace class, this property is equivalent to the No Hyphenation option in the Find & Replace Text Properties dialog box.

{button ,AL('H_HYPHENATIONOPTIONS_PROPERTY_MEMDEF',0)} [See related topics](#)

Word Pro: NonUserDocument property

{button ,AL(^H_DIVISION_CLASS;H_TEXTDOCUMENT_CLASS',0)} [See list of classes](#)

(Read-write)

Data Type

Integer.

Syntax

[objectreference].NonUserDocument = nonuserdocumentvalue

nonuserdocumentvalue = [objectreference].NonUserDocument

Legal values**Usage**

Word Pro: NormalParagraph property

{button ,AL('H_CLICKHERE_CLASS;H_TEXTMARKER_CLASS;H_TEXT_CLASS',0)} [See list of classes](#)

{button ,AL('H_NORMALPARAGRAPH_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-only) Indicates whether or not any local paragraph attributes were set in the current paragraph.

Data Type

[Integer](#)

Syntax

normalparagraphvalue = [objectreference].NormalParagraph

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Word Pro: Normal property

{button ,AL('H_FONT_CLASS',0)} [See list of classes](#)

{button ,AL('H_NORMAL_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write)

Data Type

Integer

Syntax

normalvalue = [objectreference].Normal

[objectreference].Normal = normalvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Word Pro: NoteLayouts property

{button ,AL('H_FOUNDRY_CLASS',0)} [See list of classes](#)

(Read-only) An object created from the NoteLayoutCollection class. This object provides access to NoteLayout objects.

Data Type

[NoteLayoutCollection](#)

Syntax

notelayoutsvalue = [objectreference].NoteLayouts

Legal values

Always contains an instance of the NoteLayoutCollection class.

Usage

When accessed through the Foundry property on a Division object, the collection object in this property provides access to all the NoteLayout objects contained in that Division object.

When accessed through the AppFoundry property on the WPAApplication object, this collection object provides access to all the NoteLayout objects contained in the Word Pro Clipboard.

When accessed through the TempFoundry property on the WPAApplication object, this collection object provides access to all the NoteLayout objects placed in TempFoundry by WordPro or a script.

When accessed through the Foundry property on the WPAApplication object, this collection object provides access to all the NoteLayout objects contained in the currently active Division object.

This property is not used in the Foundry property on the TextDocument object.

For more information about collection classes, see [Overview: Word Pro LotusScript Collection Classes](#).

Word Pro: NotesFlow property

{button ,AL('H_DOCCONTROL_CLASS',0)} [See list of classes](#)

{button ,AL('H_NOTESFLOW_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Indicates whether or not Notes flow is currently active.

Data Type

[Integer](#)

Syntax

notesflowvalue = [objectreference].NotesFlow

[objectreference].NotesFlow = notesflowvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Use this property to determine if Notes flow is currently active. You cannot turn the property on or off because it is read-only.

Word Pro: NumberEveryNthLine property

{button ,AL(`H_LINENUMBEROPTIONS_CLASS',0)} [See list of classes](#)

{button ,AL(`H_NUMBЕРЕVERYNTHLINE_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Determines the number by which to count when numbering lines.

Data Type

[Integer](#)

Syntax

numbereverynthlinevalue = [objectreference].NumberEveryNthLine

[objectreference].NumberEveryNthLine = numbereverynthlinevalue

Legal values

Any integer value between 1 and 99.

Usage

This property is equivalent to the "Count by" box in the Line Numbering dialog box.

{button ,AL(`H_LINESPACEDEVERYNTHUNIT_PROPERTY_MEMDEF',0)} [See related topics](#)

Word Pro: NumberFound property

{button ,AL('H_FINDANDREPLACE_CLASS',0)} [See list of classes](#)

(Read-write) Enables a count of words that corresponded to the find text at the completion of Find & Replace.

Data Type

Long

Syntax

numberfoundvalue = [objectreference].NumberFound

[objectreference].NumberFound = numberfoundvalue

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Equivalent to choosing Edit - Find & Replace Text and proceeding with Find & Replace. When finished, Word Pro displays a dialog box with a count of the words that matched the find text.

Word Pro: NumberingPosition property

{button ,AL('H_PARAGRAPHSTYLE_CLASS',0)} [See list of classes](#)

{button ,AL('H_NUMBERINGPOSITION_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Determines the outline level for a paragraph style.

Data Type

[Integer](#)

Syntax

numberingpositionvalue = [objectreference].NumberingPosition

[objectreference].NumberingPosition = numberingpositionvalue

Legal values

The legal values for this property are Integer values from 0 - 9.

Usage

This property is equivalent to choosing Outline from the Text menu, choosing Outline Styles, and assigning an outline numbering position.

Word Pro: NumberingStyleName property

{button ,AL('H_PARAGRAPHSTYLE_CLASS',0)} [See list of classes](#)

(Read-write)

Data Type

String

Syntax

numberingstylevalue = [objectreference].NumberingStyleName

[objectreference].NumberingStyleName = numberingstylevalue

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Word Pro: Numbering property

{button ,AL(^H_CLICKHERE_CLASS;H_TEXTMARKER_CLASS;H_PARAGRAPHSTYLE_CLASS;H_TEXT_CLASS',
0)} [See list of classes](#)

(Read-only)

Data Type

[Numbering](#)

Syntax

numberingvalue = [objectreference].Numbering

Legal values

Always contains an instance of the Numbering class.

Usage

Word Pro: NumberOfDataFields property

{button ,AL(^H_MERGEOPTIONS_CLASS',0)} [See list of classes](#)

(Read-only)

Data Type

Integer

Syntax

numberofdatafieldsvalue = [objectreference].NumberOfDataFields

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Word Pro: NumberOfLines property

{button ,AL(^H_LAYOUT_CLASS;H_CELLGROUPLAYOUT_CLASS;H_CELLLAYOUT_CLASS;H_CONNECTEDLAY
OUT_CLASS;H_COLUMNGROUPLAYOUT_CLASS;H_FOOTERLAYOUT_CLASS;H_FRAMELAYOUT_CLASS;H
_FRAMEGROUPLAYOUT_CLASS;H_DROPCAPLAYOUT_CLASS;H_GROUPLAYOUT_CLASS;H_HEADERLAY
OUT_CLASS;H_NOTELAYOUT_CLASS;H_PAGELAYOUT_CLASS;H_ROWGROUPLAYOUT_CLASS;H_ROWLA
YOUT_CLASS;H_RUBYLAYOUT_CLASS;H_SUPERTABLEGROUPLAYOUT_CLASS;H_SUPERTABLELAYOUT
_CLASS;H_TABLELAYOUT_CLASS;H_ENDNOTELAYOUT_CLASS;H_FOOTNOTELAYOUT_CLASS;H_TABLE
HEADINGLAYOUT_CLASS;H_TOCSUPERTABLELAYOUT_CLASS',0)} [See list of classes](#)

(Read-write) Indicates the number of lines used to determine the height of a drop cap.

Data Type

Integer

Syntax

numberoflinesvalue = [objectreference].NumberOfLines

[objectreference].NumberOfLines = numberoflinesvalue

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Equivalent to the "Height of Drop Cap" option in the Drop Cap dialog box.

Word Pro: NumberOfMergeConditions property

{button ,AL(^H_MERGEOPTIONS_CLASS',0)} [See list of classes](#)

(Read-only) Sets the number of conditions for a Merge.

Data Type

Integer

Syntax

numberofmergeconditionsvalue = [objectreference].NumberOfMergeConditions

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Word Pro: NumberOfReplacements property

{button ,AL('H_FINDANDREPLACE_CLASS',0)} [See list of classes](#)

(Read-write) Enables the count of words that used replace-with text at the completion of Find & Replace.

Data Type

Long

Syntax

numberofreplacementsvalue = [objectreference].NumberOfReplacements

[objectreference].NumberOfReplacements = numberofreplacementsvalue

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Equivalent to choosing Edit - Find & Replace Text and proceeding with Find & Replace. When finished, Word Pro displays a dialog box with a count of the words that used the replace-with text.

Word Pro: NumberOfRevisions property

{button ,AL(^H_CLICKHERE_CLASS;H_TEXTMARKER_CLASS;H_TEXT_CLASS;H_VERSION_CLASS',0)} [See list of classes](#)

(Read-only)

Data Type

[Integer](#)

Syntax

numberofrevisionsvalue = [objectreference].NumberOfRevisions

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Word Pro: NumberSequenceName property

{button ,AL('H_PREFERENCES_CLASS',0)} [See list of classes](#)

(Read-write)

Data Type

String

Syntax

numbersequencenamevalue = [objectreference].NumberSequenceName

[objectreference].NumberSequenceName = numbersequencenamevalue

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Word Pro: NumberWhichLines property

{button ,AL('H_LINENUMBEROPTIONS_CLASS',0)} [See list of classes](#)

{button ,AL('H_NUMBERWHICHLINES_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Determines the behavior of line numbering in a document.

Data Type

The data type for this property is [Variant](#) which allows the value of this property to be one of the constants listed below or its numeric equivalent (in parentheses).

Syntax

numberwhichlinesvalue = [objectreference].NumberWhichLines

[objectreference].NumberWhichLines = numberwhichlinesvalue

Legal values

<u>Value</u>	<u>Effect</u>
\$LwpLineNumberOptsNone (554)	Removes line numbering from a document.
\$LwpLineNumberOptsSpecifiedLines (555)	Adds evenly spaced line numbering to a document.
\$LwpLineNumberOptsTextLinesOnly (556)	Adds line numbering that is aligned with text or blank lines in a document

Usage

This property corresponds to the "Text lines," "Equally spaced every," and Remove buttons in the Line Numbering dialog box.

When you assign a value of \$LwpLineNumberOptsSpecifiedLines to this property, you can specify the amount of space to skip between line numbers in the LinesSpacedEveryNthUnit property.

Word Pro: Number property

{button ,AL('H_FOOTNOTE_CLASS',0)} [See list of classes](#)

{button ,AL('H_NUMBER_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-only) Indicates the number of a footnote.

Data Type

[Integer](#)

Syntax

numberpropertyvalue = [objectreference].Number

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Indicates the absolute number of a footnote within a division. This means that the first footnote of each division contains a value of 1 in its Number property.

Word Pro: NumCharsInDoc property

{button ,AL('H_DOCINFO_CLASS',0)} [See list of classes](#)

{button ,AL('H_NUMCHARSINDOC_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read only) Returns the number of characters in a document.

Data Type

[Long](#)

Syntax

numcharsindocvalue = [objectreference].NumCharsInDoc

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Word Pro: NumCharsInParagraph property

{button ,AL(^H_CLICKHERE_CLASS;H_TEXTMARKER_CLASS;H_TEXT_CLASS',0)} [See list of classes](#)

(Read-only) The number of characters in the current paragraph.

Data Type

Long

Syntax

numcharsinpsaragraphvalue = [objectreference].NumCharsInParagraph

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Word Pro: NumColsSpannedOneCell property

{button ,AL(^H_LAYOUT_CLASS;H_CELLGROUPLAYOUT_CLASS;H_CELLLAYOUT_CLASS;H_CONNECTEDLAY
OUT_CLASS;H_COLUMNGROUPLAYOUT_CLASS;H_FOOTERLAYOUT_CLASS;H_FRAMELAYOUT_CLASS;H
_FRAMEGROUPLAYOUT_CLASS;H_DROP_CAPLAYOUT_CLASS;H_GROUPLAYOUT_CLASS;H_HEADERLAY
OUT_CLASS;H_NOTELAYOUT_CLASS;H_PAGELAYOUT_CLASS;H_ROWGROUPLAYOUT_CLASS;H_ROWLA
YOUT_CLASS;H_RUBYLAYOUT_CLASS;H_SUPERTABLEGROUPLAYOUT_CLASS;H_SUPERTABLELAYOUT
_CLASS;H_TABLELAYOUT_CLASS;H_ENDNOTELAYOUT_CLASS;H_FOOTNOTELAYOUT_CLASS;H_TABLE
HEADINGLAYOUT_CLASS;H_TOCSUPERTABLELAYOUT_CLASS',0)} [See list of classes](#)

(Read-only) The number of table columns spanned by a connected cell.

Data Type

Integer

Syntax

numcolsspannedonecellvalue = [objectreference].NumColsSpannedOneCell

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

In an unconnected cell, this property contains a value of 1.

Word Pro: NumCols property

```
{button ,AL('H_LAYOUT_CLASS;H_CELLGROUPLAYOUT_CLASS;H_CELLLAYOUT_CLASS;H_CONNECTEDLAY  
OUT_CLASS;H_COLUMNGROUPLAYOUT_CLASS;H_FOOTERLAYOUT_CLASS;H_FRAMELAYOUT_CLASS;H  
_FRAMEGROUPLAYOUT_CLASS;H_DROP_CAPLAYOUT_CLASS;H_GROUPLAYOUT_CLASS;H_HEADERLAY  
OUT_CLASS;H_NOTELAYOUT_CLASS;H_PAGELAYOUT_CLASS;H_ROWGROUPLAYOUT_CLASS;H_ROWLA  
YOUT_CLASS;H_RUBYLAYOUT_CLASS;H_SUPERTABLEGROUPLAYOUT_CLASS;H_SUPERTABLELAYOUT  
_CLASS;H_TABLELAYOUT_CLASS;H_ENDNOTELAYOUT_CLASS;H_FOOTNOTELAYOUT_CLASS;H_TABLE  
HEADINGLAYOUT_CLASS;H_TOCSUPERTABLELAYOUT_CLASS;H_MARKER_CLASS;H_POWERFIELD_CLA  
SS;H_CLICKHERE_CLASS;H_TEXTMARKER_CLASS;H_RBYMARKER_CLASS;H_TABLEMARKER_CLASS;  
H_BASETABLE_CLASS;H_FOOTNOTETABLE_CLASS;H_PARALLELCOLUMNS_CLASS;H_GLOSSARY_CLAS  
S;H_TABLEHEADING_CLASS;H_TABLE_CLASS;H_DOCWINDOW_CLASS;H_WINVIEWPREFS_CLASS',0)}
```

[See list of classes](#)

```
{button ,AL('H_NUMCOLS_PROPERTY_EXSCRIPT',1)} See example
```

[Layout]

(Read-write) Allows you to control the number of columns in a layout object.

[Table]

(Read-only) Indicates the number of columns in a table.

[WinViewPrefs]

(Read-write) Determines the number of pages to be displayed across the screen in the layout view that shows multiple pages.

Data Type

Integer

Syntax

numcolsvalue = [objectreference].NumCols

[objectreference].NumCols = numcolsvalue

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

[Layout]

Equivalent to the "Number of newspaper columns" option on the Columns panel of the InfoBox for certain layout objects.

[WinViewPrefs]

Equivalent to the "Pages across screen" option on the Zoom panel of the View Preferences dialog box. Controls the number of pages that are displayed across the screen when the "View to show multiple pages" option is selected.

For more information on setting different view types, see the [ViewType](#) property.

Word Pro: NumContainers property

```
{button ,AL(^H_BASECONTAINER_CLASS;H_CELLCONTAINER_CLASS;H_FRAMECONTAINER_CLASS;H_NOTECONTAINER_CLASS;H_RUBYCONTAINER_CLASS;H_DROPDOWNCONTAINER_CLASS;H_PAGECONTAINER_CLASS;H_SUBPAGECONTAINER_CLASS;H_SUPERPAGECONTAINER_CLASS;H_ROWCONTAINER_CLASS;H_SUPERTABLECONTAINER_CLASS;H_TABLECONTAINER_CLASS;H_PARALLELCOLSCONTAINER_CLASS;H_TABLEONLYCONT_CLASS',0)} See list of classes
```

(Read-only) Returns the number of containers currently selected together.

Data Type

Integer

Syntax

numcontainersvalue = [objectreference].NumContainers

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Word Pro: NumDecimalPlaces property

{button ,AL(^H_NUMERICFORMAT_CLASS',0)} [See list of classes](#)

{button ,AL(^H_NUMDECIMALPLACES_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Allows you to access the number of decimal places that are used for numeric values within table cells.

Data Type

[Integer](#)

Syntax

numdecimalplacesvalue = [objectreference].NumDecimalPlaces

[objectreference].NumDecimalPlaces = numdecimalplacesvalue

Legal values

The legal values for this property are Integers 0 - 15.

Usage

Equivalent to the "Decimal Places" option on the Number Format panel of the InfoBox for cell layout objects.

Word Pro: NumDisplayablePagesInDoc property

{button ,AL(^H_DIVISION_CLASS;H_TEXTDOCUMENT_CLASS',0)} [See list of classes](#)

(Read-only)

Gives you the number of pages in a document excluding OLE divisions

Data Type

Integer

Syntax

numdisplayablepagesindocvalue = [objectreference].NumDisplayablePagesInDoc

Legal values

This property is read-only. The value of this property cannot be set by a script.

Usage

Word Pro: NumericFormat property

```
{button ,AL('H_LAYOUT_CLASS;H_CELLGROUPLAYOUT_CLASS;H_CELLLAYOUT_CLASS;H_CONNECTEDLAYOUT_CLASS;H_COLUMNGROUPLAYOUT_CLASS;H_FOOTERLAYOUT_CLASS;H_FRAMELAYOUT_CLASS;H_FRAMEGROUPLAYOUT_CLASS;H_DROPDOWNLAYOUT_CLASS;H_GROUPLAYOUT_CLASS;H_HEADERLAYOUT_CLASS;H_NOTELAYOUT_CLASS;H_PAGELAYOUT_CLASS;H_ROWGROUPLAYOUT_CLASS;H_ROWLAYOUT_CLASS;H_RUBYLAYOUT_CLASS;H_SUPERTABLEGROUPLAYOUT_CLASS;H_SUPERTABLELAYOUT_CLASS;H_TABLELAYOUT_CLASS;H_ENDNOTELAYOUT_CLASS;H_FOOTNOTELAYOUT_CLASS;H_TABLEHEADINGLAYOUT_CLASS;H_TOCSUPERTABLELAYOUT_CLASS',0)} See list of classes
```

```
{button ,AL('H_NUMERICFORMAT_PROPERTY_EXSCRIPT',1)} See example
```

(Read-only) Returns a numeric format object in a specific layout object.

Data Type

[NumericFormat](#)

Syntax

numericformatvalue = [objectreference].NumericFormat

Legal values

Always contains an instance of the NumericFormat class.

Usage

Word Pro: NumFields property

{button ,AL(^H_DOCINFOFIELDMANAGER_CLASS',0)} [See list of classes](#)

(Read-only) The number of document fields that currently exist in a document.

Data Type

Integer

Syntax

numfieldsvalue = [objectreference].NumFields

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Indicates the number of document fields that exist in a document. Word Pro's default document fields, such as Filename and Path, are not included in this value.

Word Pro: NumLinesOfSpaceAboveLine property

{button ,AL(^H_SPACING_CLASS',0)} [See list of classes](#)

(Read-write)

Data Type

Long.

Syntax

[objectreference].NumLinesOfSpaceAboveLine = numlinesofspaceabovelinevalue

numlinesofspaceabovelinevalue = [objectreference].NumLinesOfSpaceAboveLine

Legal values

Data type is Long but the unit of measurement used for this property is Twips. There are 1440 Twips per inch.

Usage

Word Pro: NumLinesOfSpaceAbove property

{button ,AL('H_SPACING_CLASS',0)} [See list of classes](#)

{button ,AL('H_NUMLINESOFSPACEABOVE_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write)

Data Type

Long

Syntax

numlinesofspaceabovevalue = [objectreference].NumLinesOfSpaceAbove

[objectreference].NumLinesOfSpaceAbove = numlinesofspaceabovevalue

Legal values

Data type is Long but the unit of measurement used for this property is Twips. There are 1440 Twips per inch.

Usage

Word Pro: NumLinesOfSpaceBelow property

{button ,AL('H_SPACING_CLASS',0)} [See list of classes](#)

{button ,AL('H_NUMLINESOFSPACEBELOW_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write)

Data Type

Long

Syntax

numlinesofspacebelowvalue = [objectreference].NumLinesOfSpaceBelow

[objectreference].NumLinesOfSpaceBelow = numlinesofspacebelowvalue

Legal values

Data type is Long but the unit of measurement used for this property is Twips. There are 1440 Twips per inch.

Usage

Word Pro: NumLinesOfSpace property

{button ,AL('H_SPACING_CLASS',0)} [See list of classes](#)

{button ,AL('H_NUMLINESOFSPACE_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write)

Data Type

Long

Syntax

numlinesofspacevalue = [objectreference].NumLinesOfSpace

[objectreference].NumLinesOfSpace = numlinesofspacevalue

Legal values

Data type is Long but the unit of measurement used for this property is Twips. There are 1440 Twips per inch.

Usage

Word Pro: NumOfRecentFiles property

{button ,AL(^H_USERINTERFACEPREFS_CLASS',0)} [See list of classes](#)

{button ,AL(^H_NUMOFRECENTFILES_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) The maximum number of recent files displayed at the end of the File menu.

Data Type

[Integer](#)

Syntax

numofrecentfilesvalue = [objectreference].NumOfRecentFiles

[objectreference].NumOfRecentFiles = numofrecentfilesvalue

Legal values

Integer between 0 and 5.

Usage

Equivalent to the "Recent files" option on the General panel of the Word Pro Preferences dialog box.

Word Pro: NumPagesInDoc property

{button ,AL(`H_DIVISION_CLASS;H_DOCINFO_CLASS;H_TEXTDOCUMENT_CLASS',0)} [See list of classes](#)

{button ,AL(`H_NUMPAGESINDOC_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-only) Returns the number of pages in a document.

Data Type

[Integer](#)

Syntax

numpagesindocvalue = [objectreference].NumPagesInDoc

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Word Pro: NumParagraphs property

{button ,AL(^H_SORTOPTIONS_CLASS',0)} [See list of classes](#)

(Read-write) Allows you to specify the number of paragraphs or rows that constitute a record. All records that are to be sorted must contain the same number of paragraphs or rows.

Data Type

[Integer](#)

Syntax

numparagraphsvalue = [objectreference].NumParagraphs

[objectreference].NumParagraphs = numparagraphsvalue

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Equivalent to choosing Text - Sort and selecting a number in the "Number of paragraphs/row in record" box. If you use this property in a table, it allows you to specify the number of rows you want to sort.

Word Pro: NumRowsSpannedOneCell property

```
{button ,AL(^H_LAYOUT_CLASS;H_CELLGROUPLAYOUT_CLASS;H_CELLLAYOUT_CLASS;H_CONNECTEDLAY  
OUT_CLASS;H_COLUMNGROUPLAYOUT_CLASS;H_FOOTERLAYOUT_CLASS;H_FRAMELAYOUT_CLASS;H  
_FRAMEGROUPLAYOUT_CLASS;H_DROP_CAPLAYOUT_CLASS;H_GROUPLAYOUT_CLASS;H_HEADERLAY  
OUT_CLASS;H_NOTELAYOUT_CLASS;H_PAGELAYOUT_CLASS;H_ROWGROUPLAYOUT_CLASS;H_ROWLA  
YOUT_CLASS;H_RUBYLAYOUT_CLASS;H_SUPERTABLEGROUPLAYOUT_CLASS;H_SUPERTABLELAYOUT  
_CLASS;H_TABLELAYOUT_CLASS;H_ENDNOTELAYOUT_CLASS;H_FOOTNOTELAYOUT_CLASS;H_TABLE  
HEADINGLAYOUT_CLASS;H_TOCSUPERTABLELAYOUT_CLASS',0)} See list of classes
```

(Read-only) The number of table rows spanned by a connected cell.

Data Type

Integer

Syntax

numrowsspannedonecellvalue = [objectreference].NumRowsSpannedOneCell

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

In an unconnected cell, this property contains a value of 1.

Word Pro: NumRowsThatFit property

{button ,AL('H_DOCWINDOW_CLASS',0)} [See list of classes](#)

(Read-write)

Data Type

Integer

Syntax

numrowsthatfitvalue = [objectreference].NumRowsThatFit

[objectreference].NumRowsThatFit = numrowsthatfitvalue

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Word Pro: NumRows property

{button ,AL(^H_MARKER_CLASS;H_POWERFIELD_CLASS;H_CLICKHERE_CLASS;H_TEXTMARKER_CLASS;H_RUBYMARKER_CLASS;H_TABLEMARKER_CLASS;H_BASetable_CLASS;H_FOOTNOTETABLE_CLASS;H_PARALLELCOLUMNS_CLASS;H_GLOSSARY_CLASS;H_TABLEHEADING_CLASS;H_TABLE_CLASS',0)} [See list of classes](#)

(Read-only) Indicates the number of rows in a table.

Data Type

[Integer](#)

Syntax

numrowsvalue = [objectreference].NumRows

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Word Pro: NumTabs property

{button ,AL(^H_TABRACK_CLASS',0)} [See list of classes](#)

(Read-only)

Data Type

Integer

Syntax

numtabsvalue = [objectreference].NumTabs

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Word Pro: NumUndoLevels property

{button ,AL(^H_PREFERENCES_CLASS',0)} [See list of classes](#)

{button ,AL(^H_NUMUNDOLEVELS_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Sets or returns the number of actions Word Pro can undo.

Data Type

[Integer](#)

Syntax

numundolevelsvalue = [objectreference].NumUndoLevels

[objectreference].NumUndoLevels = numundolevelsvalue

Legal values

The legal values for this property are 0 - 99.

Usage

Equivalent to the "Undo levels" option, located on the General tab of the Word Pro Preferences dialog box.

Word Pro: NumWindowsViewingDoc property

{button ,AL(^H_DIVISION_CLASS;H_TEXTDOCUMENT_CLASS',0)} [See list of classes](#)

{button ,AL(^H_NUMWINDOWSVIEWINGDOC_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-only)

Data Type

[Integer](#)

Syntax

numwindowsviewingdocvalue = [objectreference].NumWindowsViewingDoc

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Word Pro: NumWordsInDoc property

{button ,AL('H_DOCINFO_CLASS;H_TEXTDOCUMENT_CLASS',0)} [See list of classes](#)

{button ,AL('H_NUMWORDSINDOC_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-only) Returns the number of words in a document.

Data Type

Long

Syntax

numwordsindocvalue = [objectreference].NumWordsInDoc

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Word Pro: ObjectType property

{button ,AL(^H_CLICKHERE_CLASS;H_TEXTMARKER_CLASS;H_TEXT_CLASS',0)} [See list of classes](#)

(Read-only) Fribtype.

Data Type

String

Syntax

objectypevalue = [objectreference].ObjectType

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Word Pro: Object property

{button ,AL(^H_OLEOBJECT_CLASS',0)} [See list of classes](#)

(Read-only)

Note This property is not supported for the class OLEObject within OS/2.

Data Type

Variant

Syntax

objectvalue = [objectreference].Object

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Word Pro: Oblique property

{button ,AL(^H_FONTMETRICS_CLASS',0)} [See list of classes](#)

(Read-only)

Data Type

Integer

Syntax

obliquevalue = [objectreference].Oblique

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Word Pro: OCXDesignMode property

{button ,AL('H_DOCCONTROL_CLASS',0)} [See list of classes](#)

{button ,AL('H_OCXDESIGNMODE_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) This property enables or disables OCX design mode.

Data Type

[Integer](#)

Syntax

ocxdesignmodevalue = [objectreference].OCXDesignMode

[objectreference].OCXDesignMode = ocxdesignmodevalue

Legal values

The legal values for this property are -1 and 0 but you may use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Equivalent to choosing View - Design Mode.

Setting this property to True enables OCX design mode. In OCX design mode, you can select OCX controls and modify their properties. A False setting enables OCX run mode. In run mode, OCX controls are functional and their properties cannot be modified.

Word Pro: Ole1Object property

{button ,AL(^H_OLEOBJECT_CLASS',0)} [See list of classes](#)

(Read-only)

Data Type

Integer.

Syntax

ole1objectvalue = [objectreference].Ole1Object

Legal values**Usage**

Word Pro: OleContainerDocName property

{button ,AL(^H_TEXTDOCUMENT_CLASS',0)} [See list of classes](#)

(Read-only)

Data Type

String

Syntax

olecontainerdocnamevalue = [objectreference].OleContainerDocName

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Word Pro: OleContainer property

{button ,AL(^H_TEXTDOCUMENT_CLASS',0)} [See list of classes](#)

(Read-only)

Returns the automation object for the container application in which Word Pro is embedded.

Data Type

Variant

Syntax

olecontainervalue = [objectreference].OleContainer

Legal values

This property is read-only. The value of this property cannot be set by a script.

Usage

The legal value for OleContainer is any valid automation object.

Word Pro: OleEmbeddedSize property

{button ,AL(^H_TEXTDOCUMENT_CLASS',0)} [See list of classes](#)

(Read-only)

When we are OLE embedded - the size in bytes of the OLE object

Data Type

Long

Syntax

oleembeddedsizevalue = [objectreference].OleEmbeddedSize

Legal values

This property is read-only. The value of this property cannot be set by a script.

Usage

Word Pro: OLEEnabled property

{button ,AL(^H_PREFERENCES_CLASS',0)} [See list of classes](#)

{button ,AL(^H_OLEENABLED_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write)

Data Type

Integer.

Syntax

[objectreference].OLEEnabled = oleenabledvalue

oleenabledvalue = [objectreference].OLEEnabled

Legal values**Usage**

Word Pro: OleMinHeight property

{button ,AL('H_TEXTDOCUMENT_CLASS',0)} [See list of classes](#)

(Read-write)

Data Type

Long.

Syntax

[objectreference].OleMinHeight = oleminheightvalue

oleminheightvalue = [objectreference].OleMinHeight

Legal values

Usage

Word Pro: OleObjectSize property

{button ,AL(^H_OLEOBJECT_CLASS',0)} [See list of classes](#)

(Read-only)

Note This property is not supported for the class OLEObject within OS/2.

Data Type

Long.

Syntax

oleobjectsizevalue = [objectreference].OleObjectSize

Legal values**Usage**

Word Pro: OleObjects property

{button ,AL('H_FOUNDRY_CLASS',0)} [See list of classes](#)

(Read-only) An object created from the OleObjectCollection class. This object provides access to OleObject objects.

Note OLE is not supported under OS/2.

Data Type

[OleObjectCollection](#)

Syntax

oleobjectsvalue = [objectreference].OleObjects

Legal values

Always contains an instance of the OleObjectCollection class.

Usage

Use this property to determine if any OLE objects exist in a particular object.

When accessed through the Foundry property on a Division object, the collection object in this property provides access to all the OleObject objects contained in that Division object.

When accessed through the AppFoundry property on the WPApplication object, this collection object provides access to all the OleObject objects contained in the Word Pro Clipboard.

When accessed through the TempFoundry property on the WPApplication object, this collection object provides access to all the OleObject objects placed in TempFoundry by WordPro or a script.

When accessed through the Foundry property on the WPApplication object, this collection object provides access to all the OleObject objects contained in the currently active Division object.

This property is not used in the Foundry property on the TextDocument object.

For more information about collection classes, see [Overview: Word Pro LotusScript Collection Classes](#).

Word Pro: OleObject property

{button ,AL('H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

(Read-only) The OleObject object which is uppermost in the focus of the currently active document.

Note This property is not supported for the class OLEObject within OS/2.Data Type

OleObject

Syntax

oleobjectvalue = [objectreference].OleObject

Legal values

Always contains an instance of the OleObject class.

Usage

Use this property when you want to access the OLE object that currently has the focus. If you want to access a graphic that is not an OLE object, use the Graphic property on WPAApplication. If you are not sure if a graphic is an OLE object, use the GraphicOleObject property that is capable of containing both Graphic objects and OleObject objects.

Word Pro: OpenDocMacroName property

{button ,AL('H_AUTORUNMACRO_CLASS',0)} [See list of classes](#)

(Read-write)

Data Type

String

Syntax

opendocmacrovalue = [objectreference].OpenDocMacroName

[objectreference].OpenDocMacroName = opendocmacrovalue

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Word Pro: OpenDocsVisible property

{button ,AL('H_USERINTERFACEPREFS_CLASS',0)} [See list of classes](#)

{button ,AL('H_OPENDOCSVISIBLE_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Displays all documents that are opened in the current Word Pro session.

Data Type

[Integer](#)

Syntax

opendocsvisiblevalue = [objectreference].OpenDocsVisible

[objectreference].OpenDocsVisible = opendocsvisiblevalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values. Default is 0.

Usage

For some features, Word Pro creates, opens, modifies, and saves files without the user actually viewing them. When a user performs a merge, for example, Word Pro creates, opens, modifies, and/or saves the Merge data file without displaying it.

If this property is set to True (-1), you can see all documents that are opened subsequently. If set to False (0), files that are opened subsequently are not visible.

Word Pro: OpenExistingFileInWelcomeBox property

{button ,AL('H_USERINTERFACEPREFS_CLASS',0)} [See list of classes](#)

(Read-write) Determines what panel displays in the Welcome dialog box.

Data Type

Integer

Syntax

openexistingfileinwelcomeboxvalue = [objectreference].OpenExistingFileInWelcomeBox

[objectreference].OpenExistingFileInWelcomeBox = openexistingfileinwelcomeboxvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values. Default is False (0).

Usage

In order to use this property, the ShowNoWelcomeBox property must be set to False (0).

If this property is set to True (-1), Word Pro displays the Welcome dialog box with the Open an Existing Document panel in the foreground. If set to False (0), Word Pro displays the Welcome dialog box with the Create a New Document from a SmartMaster panel in the foreground.

Word Pro: OpenReadOnly property

{button ,AL('H_USERINTERFACEPREFS_CLASS',0)} [See list of classes](#)

{button ,AL('H_OPENREADONLY_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Opens a file as read-only.

Data Type

[Integer](#)

Syntax

openreadonlyvalue = [objectreference].OpenReadOnly

[objectreference].OpenReadOnly = openreadonlyvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values. Default is False (0).

Usage

Use this property to open any document in read-only mode.

Word Pro: Options property

{button ,AL(^H_FORMATCHECKPREF_CLASS;H_MERGEOPTIONS_CLASS',0)} [See list of classes](#)

(Read-write)

[MergeOptions]

A flag that sets various options for a Merge.

Data Type

Variant

Syntax

optionsvalue = [objectreference].Options

[objectreference].Options = optionsvalue

Legal values

<u>Value</u>	<u>Effect</u>
LwpFormatCheckOptionsAll (&H1F)	
LwpFormatCheckOptionsFixacronyms (&H1)	
LwpFormatCheckOptionsFixbulletlist (&H4)	
LwpFormatCheckOptionsFixmargins (&H8)	
LwpFormatCheckOptionsFixspaces (&H2)	
LwpFormatCheckOptionsTwospaces (&H10)	
LwpMergeOptFlgMergeAndPrint (&H1)	Prints the Merge data file. If you set this flag, you cannot view any data records as you merge.
LwpMergeOptFlgMergeViewAndPrint (&H2)	Allows you to view each data record in the Merge process.
LwpMergeOptFlgMergeAndSaveAs (&H4)	
LwpMergeOptFlgMergeInsertFields (&H8)	Indicates you inserted a Merge data field.
LwpMergeOptFlgMergeLabels (&H10)	
LwpMergeOptFlgMergeMacro (&H20)	Prevents the Merge bar from displaying during the Merge process.

Usage

Word Pro: OrigFileType property

{button ,AL('H_TEXTDOCUMENT_CLASS',0)} [See list of classes](#)

(Read-write)

Data Type

String

Syntax

origfiletypevalue = [objectreference].OrigFileType

[objectreference].OrigFileType = origfiletypevalue

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Word Pro: OrigHeight property

{button ,AL('H_GRAPHICOLEBJECT_CLASS;H_GRAPHIC_CLASS;H_OLEOBJECT_CLASS',0)} [See list of classes](#)

{button ,AL('H_ORIGHEIGHT_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-only)

Data Type

[Long](#)

Syntax

origheightvvalue = [objectreference].OrigHeight

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Word Pro: OrigWidth property

{button ,AL('H_GRAPHICOLEBJECT_CLASS;H_GRAPHIC_CLASS;H_OLEOBJECT_CLASS',0)} [See list of classes](#)

{button ,AL('H_ORIGWIDTH_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-only)

Data Type

[Long](#)

Syntax

origwidthvalue = [objectreference].OrigWidth

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Word Pro: OStype property

{button ,AL('H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL('H_OSTYPE_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-only)

This language element has not yet been defined. Lotus provides updated Help files on its web site.

Data Type

The data type for this property is Variant.

Syntax

ostypevalue = [objectreference].OStype

Legal values

This property is read-only. The value of this property cannot be set by a script. The possible values for this property are listed below.

\$LwpPlatformOs2 (2486)

\$LwpPlatformWin16 (2482)

\$LwpPlatformWin32 (2483)

\$LwpPlatformWin95 (2484)

\$LwpPlatformWinnt (2485)

Usage

Word Pro: OutlineBorderLines property

{button ,AL(^H_TABLELINE_CLASS',0)} [See list of classes](#)

(Read-Write) Allows you to return or set the style of the outside border line of a table object.

Data Type

[BorderLines](#)

Syntax

outlineborderlinesvalue = [objectreference].OutlineBorderLines

[objectreference].OutlineBorderLines = outlineborderlinesvalue

Legal values

Always contains an instance of the BorderLines class.

Usage

Word Pro: OutlineButtons property

{button ,AL('H_WINVIEWPREFS_CLASS',0)} [See list of classes](#)

{button ,AL('H_OUTLINEBUTTONS_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Determines whether Word Pro displays outline buttons in outline mode.

Data Type

[Integer](#)

Syntax

outlinebuttonsvalue = [objectreference].OutlineButtons

[objectreference].OutlineButtons = outlinebuttonsvalue

Legal values

Data type is Integer. The legal values for this property are 1 and 0.

Usage

Equivalent to the "Show outline buttons" option on the Outline panel of the View Preferences dialog box.

If this property is set to 1, Word Pro displays outline buttons in outline mode. If set to 0, Word Pro does not display outline buttons in outline mode.

Word Pro: OutlineHeadingButtonsOnly property

{button ,AL(^H_WINVIEWPREFS_CLASS',0)} [See list of classes](#)

(Read-write) Determines whether Word Pro displays buttons in outline mode only for paragraphs which are assigned heading paragraph styles.

Data Type

[Integer](#)

Syntax

outlineheadingbuttonsonlyvalue = [objectreference].OutlineHeadingButtonsOnly

[objectreference].OutlineHeadingButtonsOnly = outlineheadingbuttonsonlyvalue

Legal values

Data type is Integer.

The legal values for this property are 1 and 0.

0 = Don't show buttons

1 = Show buttons

Usage

Equivalent to the "Show outline button for headings only" option on the Outline panel of the View Preferences dialog box.

If this property is set to True, Word Pro displays buttons in outline mode only for paragraphs which are assigned heading paragraph styles. In order for these buttons to display, the "Show outline buttons" option must also be selected (see the [OutlineButtons](#) topic for more information).

If set to False, Word Pro does not display buttons in outline mode only for paragraphs which are assigned heading paragraph styles. The OutlineButtons property determines whether any outline buttons display.

Word Pro: OutlineLevel property

{button ,AL('H_PRESENTATION_CLASS',0)} [See list of classes](#)

{button ,AL('H_OUTLINELEVEL_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Determines the highest outline level displayed.

Data Type

[Integer](#)

Syntax

outlinelevelvalue = [objectreference].OutlineLevel

[objectreference].OutlineLevel = outlinelevelvalue

Legal values

The legal values for this property are 0 - 9.

Usage

This property is equivalent to choosing Outline from the Text menu, then choosing Collapse to Level, and then choosing an outline level to display.

Word Pro: OutlineOnlyHeadingsWhenCollapsed property

{button ,AL('H_WINVIEWPREFS_CLASS',0)} [See list of classes](#)

{button ,AL('H_OUTLINEONLYHEADINGSWHENCOLLAPSED_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Determines whether Word Pro displays only outline headings in outline mode.

Data Type

[Integer](#)

Syntax

outlineonlyheadingswhencollapsedvalue = [objectreference].OutlineOnlyHeadingsWhenCollapsed

[objectreference].OutlineOnlyHeadingsWhenCollapsed = outlineonlyheadingswhencollapsedvalue

Legal values

Data type is Integer. The legal values for this property are 1 and 0.

Usage

Equivalent to the "Show only headings when collapsed to level" option on the Outline panel of the View Preferences dialog box.

If this property is set to 1, when a paragraph level is collapsed in outline mode, Word Pro displays only the heading paragraphs. If set to 0, when a paragraph level is collapsed in outline mode, Word Pro displays heading paragraphs and any subordinate text beneath a higher outline level.

For example, if this property is set to 1 and outline level 2 is collapsed, Word Pro displays level 1 and level 2 paragraph headings. If set to 0 and outline level 2 is collapsed, Word Pro displays level 1 and level 2 paragraph headings, as well as any subordinate text beneath the level 1 paragraph headings.

Word Pro: OutlineSeqItems property

{button ,AL('H_OUTLINESTYLESEQUENCE_CLASS',0)} [See list of classes](#)

{button ,AL('H_OUTLINESEQITEMS_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-only)

Data Type

[OutlineSeqItemCollection](#)

Syntax

outlineseqitemsvalue = [objectreference].OutlineSeqItems

Legal values

Always contains an instance of the OutlineSeqItemCollection class.

Usage

Word Pro: OutlineStyleSequences property

{button ,AL('H_FOUNDRY_CLASS',0)} [See list of classes](#)

{button ,AL('H_OUTLINESTYLESEQUENCES_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-only) An object created from the OutlineSeqCollection class. This object provides access to OutlineStyleSequence objects.

Data Type

[OutlineSeqCollection](#)

Syntax

outlinestylesequencesvalue = [objectreference].OutlineStyleSequences

Legal values

Always contains an instance of the OutlineSeqCollection class.

Usage

When accessed through the Foundry property on a Division object, the collection object in this property provides access to all the OutlineStyleSequence objects contained in that Division object.

When accessed through the AppFoundry property on the WPAApplication object, this collection object provides access to all the OutlineStyleSequence objects contained in the Word Pro Clipboard.

When accessed through the TempFoundry property on the WPAApplication object, this collection object provides access to all the OutlineStyleSequence objects placed in TempFoundry by WordPro or a script.

When accessed through the Foundry property on the WPAApplication object, this collection object provides access to all the OutlineStyleSequence objects contained in the currently active Division object.

This property is not used in the Foundry property on the TextDocument object.

For more information about collection classes, see [Overview: Word Pro LotusScript Collection Classes](#).

Word Pro: Outline property

{button ,AL(^H_FONTMETRICS_CLASS',0)} [See list of classes](#)

(Read-only)

Data Type

[Integer](#)

Syntax

outlinevalue = [objectreference].Outline

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Word Pro: OutputToFile property

{button ,AL('H_PRINTSETTINGS_CLASS',0)} [See list of classes](#)

{button ,AL('H_OUTPUTTOFILE_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Indicates whether or not Word Pro outputs the document to a file when it prints.

Data Type

[Integer](#)

Syntax

outputtofilevalue = [objectreference].OutputToFile

[objectreference].OutputToFile = outputtofilevalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Equivalent to choosing File - Print and selecting "Print to file."

Word Pro: OverrideCheckAndEnable property

{button ,AL('H_MENUITEM_CLASS',0)} [See list of classes](#)

(Read-write)

This language element has not yet been defined. You can check for a newer Help file on the Lotus web site.

Data Type

Integer

Syntax

overridecheckandenablevalue = [objectreference].OverrideCheckAndEnable

[objectreference].OverrideCheckAndEnable = overridecheckandenablevalue

Legal values

The legal values for this property are -1 and 0 but you may use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Word Pro: Override property

{button ,AL('H_COLOR_CLASS',0)} [See list of classes](#)

{button ,AL('H_OVERRIDE_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Determines what information is used to define an object's color.

Data Type

Data type is [Variant](#), which allows the value of this property to be one of the constants listed below or its numeric equivalent (in parentheses).

Syntax

overridevalue = [objectreference].Override

[objectreference].Override = overridevalue

Legal values

<u>Value</u>	<u>Effect</u>
\$LwpColorOverrideBlack (2017)	Assigns the predefined WordPro color black to an object and ignores the red, green, and blue properties of the color object.
\$LwpColorOverrideBlue (2024)	Assigns the predefined WordPro color blue to an object and ignores the red, green, and blue properties of the color object.
\$LwpColorOverrideDarkGray (2021)	Assigns the predefined WordPro color dark gray to an object and ignores the red, green, and blue properties of the color object.
\$LwpColorOverrideGray (2019)	Assigns the predefined WordPro color gray to an object and ignores the red, green, and blue properties of the color object.
\$LwpColorOverrideGreen (2023)	Assigns the predefined WordPro color green to an object and ignores the red, green, and blue properties of the color object.
\$LwpColorOverrideInvalid (2026)	Only assigned internally by Word Pro.
\$LwpColorOverrideLightGray (2020)	Assigns the predefined WordPro color light gray to an object and ignores the red, green, and blue properties of the color object.
\$LwpColorOverrideRed (2022)	Assigns the predefined WordPro color red to an object and ignores the red, green, and blue properties of the color object.
\$LwpColorOverrideReserved (2025)	Not implemented.
\$LwpColorOverrideRgb (2016)	Assigns the values in the red, green, and blue properties of the color object.
\$LwpColorOverrideWhite (2018)	Assigns the predefined WordPro color white to an object and ignores the red, green, and blue properties of the color object.

Usage

If the Override property value is set to RGB, then the object's color is defined by the values in the red, green, and blue properties of the color object. All other Override values use a predefined WordPro color, and ignore the red, green, and blue properties of the color object.

Word Pro: OverstrikeCharacter property

{button ,AL('H_FONT_CLASS',0)} [See list of classes](#)

{button ,AL('H_OVERSTRIKECHARACTER_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write)

Data Type

Long

Syntax

overstrikecharactervalue = [objectreference].OverstrikeCharacter

[objectreference].OverstrikeCharacter = overstrikecharactervalue

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Word Pro: Overstrike property

{button ,AL('H_FONT_CLASS',0)} [See list of classes](#)

{button ,AL('H_OVERSTRIKE_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Determines whether the overstrike attribute is on (True) or off (False) for the specified Font object.

Data Type

[Integer](#)

Syntax

overstrikevalue = [objectreference].Overstrike

[objectreference].Overstrike = overstrikevalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

For the Font object in the Text class, this property is equivalent to the "Strikethrough" attribute, located on the Text Font properties panel of the InfoBox.

Word Pro: PageNo property

{button ,AL(^H_USEWHEN_CLASS',0)} [See list of classes](#)

(Read-write)

Data Type

Integer

Syntax

pagenovalue = [objectreference].PageNo

[objectreference].PageNo = pagenovalue

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Word Pro: PageNumberAsText property

{button ,AL('H_MARKER_CLASS;H_POWERFIELD_CLASS;H_CLICKHERE_CLASS;H_TEXTMARKER_CLASS;H_RUBYMARKER_CLASS;H_TABLEMARKER_CLASS',0)} [See list of classes](#)

{button ,AL('H_PAGENUMBERASTEXT_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-only)

Data Type

[String](#)

Syntax

pagenumberastextvalue = [objectreference].PageNumberAsText

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Word Pro: PageNumber property

{button ,AL(^H_MARKER_CLASS;H_POWERFIELD_CLASS;H_CLICKHERE_CLASS;H_TEXTMARKER_CLASS;H_RUBYMARKER_CLASS;H_TABLEMARKER_CLASS',0)} [See list of classes](#)

{button ,AL(^H_PAGENUMBER_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-only) The page number on which a marker is located.

Data Type

[Integer](#)

Syntax

pagenumbervalue = [objectreference].PageNumber

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

This property is valid for all marker types except markers created with the \$LWPMarkerTypeDefault type.

Word Pro: PageNumFirstPageShowing property

{button ,AL(^H_DOCWINDOW_CLASS',0)} [See list of classes](#)

(Read-only)

Data Type

[Integer](#)

Syntax

pagenumfirstpageshowingvalue = [objectreference].PageNumFirstPageShowing

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Word Pro: PageNumLastPageShowing property

{button ,AL(^H_DOCWINDOW_CLASS',0)} [See list of classes](#)

(Read-only)

Data Type

[Integer](#)

Syntax

pagenumlastpageshowingvalue = [objectreference].PageNumLastPageShowing

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Word Pro: PageNum property

```
{button ,AL('H_BASECONTAINER_CLASS;H_CELLCONTAINER_CLASS;H_FRAMECONTAINER_CLASS;H_NOTECONTAINER_CLASS;H_RUBYCONTAINER_CLASS;H_DROPDOWNCONTAINER_CLASS;H_PAGECONTAINER_CLASS;H_SUBPAGECONTAINER_CLASS;H_SUPERPAGECONTAINER_CLASS;H_ROWCONTAINER_CLASS;H_SUPERTABLECONTAINER_CLASS;H_TABLECONTAINER_CLASS;H_PARALLELCOLSCONTAINER_CLASS;H_TABLEONLYCONT_CLASS',0)} See list of classes
```

```
{button ,AL('H_PAGENUM_PROPERTY_EXSCRIPT',1)} See example
```

(Read-only) Returns the absolute page number on which the current container is located.

Data Type

Integer

Syntax

pagenumvalue = [objectreference].PageNum

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Absolute page numbers continue sequentially throughout the entire document, across all divisions. Relative page numbers usually number pages sequentially, within a single division.

Word Pro: PageOrder property

{button ,AL('H_PRINTSETTINGS_CLASS',0)} [See list of classes](#)

{button ,AL('H_PAGEORDER_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Specifies if a document prints from front to back, or from back to front.

Data Type

Variant (Enumerated)

PageOrder

Syntax

pageordervalue = [objectreference].PageOrder

[objectreference].PageOrder = pageordervalue

Legal value

<u>Value</u>	<u>Effect</u>
\$LtsPageOrderBackToFront (1056964852)	The document prints from back to front.
\$LtsPageOrderFrontToBack (1056964851)	The document prints from front to back.

Usage

Equivalent to the "In reverse order" option on the Print Options dialog box.

Word Pro: PageStyleName property

{button ,AL('H_PREFERENCES_CLASS',0)} [See list of classes](#)

(Read-write)

Data Type

String

Syntax

pagestylevalue = [objectreference].PageStyleName

[objectreference].PageStyleName = pagestylevalue

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Word Pro: PageStyles property

{button ,AL('H_FOUNDRY_CLASS',0)} [See list of classes](#)

(Read-only) An object created from the PageLayoutCollection class. This object provides access to PageLayout objects which are used as page styles.

Data Type

[PageLayoutCollection](#)

Syntax

pagestylesvalue = [objectreference].PageStyles

Legal values

Always contains an instance of the PageLayoutCollection class.

Usage

When accessed through the Foundry property on a Division object, the collection object in this property provides access to all the PageLayout objects which are used as page styles and contained in that Division object.

When accessed through the AppFoundry property on the WPAApplication object, this collection object provides access to all the PageLayout objects which are used as page styles and contained in the Word Pro Clipboard.

When accessed through the TempFoundry property on the WPAApplication object, this collection object provides access to all the PageLayout objects which are used as page styles and placed in TempFoundry by WordPro or a script.

When accessed through the Foundry property on the WPAApplication object, this collection object provides access to all the PageLayout objects which are used as page styles and contained in the currently active Division object.

This property is not used in the Foundry property on the TextDocument object.

For more information about collection classes, see [Overview: Word Pro LotusScript Collection Classes](#).

Word Pro: Pages property

{button ,AL('H_FOUNDRY_CLASS',0)} [See list of classes](#)

(Read-only) An object created from the PageLayoutCollection class. This object provides access to PageLayout objects.

Data Type

[PageLayoutCollection](#)

Syntax

pagesvalue = [objectreference].Pages

Legal values

Always contains an instance of the PageLayoutCollection class.

Usage

When accessed through the Foundry property on a Division object, the collection object in this property provides access to all the PageLayout objects contained in that Division object.

When accessed through the AppFoundry property on the WPApplication object, this collection object provides access to all the PageLayout objects contained in the Word Pro Clipboard.

When accessed through the TempFoundry property on the WPApplication object, this collection object provides access to all the PageLayout objects placed in TempFoundry by WordPro or a script.

When accessed through the Foundry property on the WPApplication object, this collection object provides access to all the PageLayout objects contained in the currently active Division object.

This property is not used in the Foundry property on the TextDocument object.

For more information about collection classes, see [Overview: Word Pro LotusScript Collection Classes](#).

Word Pro: PageToUseLayoutOn property

```
{button ,AL('H_LAYOUT_CLASS;H_CELLGROUPLAYOUT_CLASS;H_CELLLAYOUT_CLASS;H_CONNECTEDLAYOUT_CLASS;H_COLUMNGROUPLAYOUT_CLASS;H_FOOTERLAYOUT_CLASS;H_FRAMELAYOUT_CLASS;H_FRAMEGROUPLAYOUT_CLASS;H_DROP_CAP_LAYOUT_CLASS;H_GROUPLAYOUT_CLASS;H_HEADERLAYOUT_CLASS;H_NOTELAYOUT_CLASS;H_PAGELAYOUT_CLASS;H_ROWGROUPLAYOUT_CLASS;H_ROWLAYOUT_CLASS;H_RUBY_LAYOUT_CLASS;H_SUPERTABLEGROUPLAYOUT_CLASS;H_SUPERTABLELAYOUT_CLASS;H_TABLELAYOUT_CLASS;H_ENDNOTELAYOUT_CLASS;H_FOOTNOTELAYOUT_CLASS;H_TABLEHEADINGLAYOUT_CLASS;H_TOCSUPERTABLELAYOUT_CLASS',0)} See list of classes
```

```
{button ,AL('H_PAGETOUSELAYOUTON_PROPERTY_EXSCRIPT',1)} See example
```

(Read-write) Allows you to set the page on which a specific layout will be used.

Data Type

[Integer](#)

Syntax

pagetouselayoutonvalue = [objectreference].PageToUseLayoutOn

[objectreference].PageToUseLayoutOn = pagetouselayoutonvaluew

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

[HeaderLayout] [FooterLayout]

Controls the page on which the headers or footers begin.

```
{button ,AL('H_CONDITIONTYPE_PROPERTY_MEMDEF',0)} See related topics
```

Word Pro: Page property

{button ,AL('H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL('H_PAGE_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-only) An instance of the PageContainer class. This is a current context property which only contains an object when the focus of Word Pro includes a page. If there is no page in the focus, this property is empty.

Data Type

[PageContainer](#)

Syntax

pagevalue = [objectreference].Page

Legal values

An instance of the PageContainer class.

Usage

When the focus includes a page, this property contains the PageContainer object which groups together objects that comprise the page that has the focus. Use this property to access the Layout or other objects related to that page.

Word Pro: PairKerning property

{button ,AL(^H_OPTIONS_CLASS',0)} [See list of classes](#)

(Read-write)

Data Type

Integer

Syntax

pairkerningvalue = [objectreference].PairKerning

[objectreference].PairKerning = pairkerningvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Word Pro: PaperNames property

{button ,AL(`H_PRINTMANAGER_CLASS',0)} [See list of classes](#)

{button ,AL(`H_PAPERNAME_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-only) A collection of the names of the available page sizes.

Data Type

[StringCollection](#)

Syntax

papernamesvalue = [objectreference].PaperNames

Legal values

Always contains an instance of the StringCollection class.

Usage

This collection of values is equivalent to the list of sizes in the "Page size" box on the Size and margins panel of the Page InfoBox.

{button ,AL(`H_PAPERNAME_PROPERTY_MEMDEF',0)} [See related topics](#)

Word Pro: PaperName property

{button ,AL(`H_LAYOUT_CLASS;H_CELLGROUPLAYOUT_CLASS;H_CELLLAYOUT_CLASS;H_CONNECTEDLAYOUT_CLASS;H_COLUMNGROUPLAYOUT_CLASS;H_FOOTERLAYOUT_CLASS;H_FRAMELAYOUT_CLASS;H_FRAMEGROUPLAYOUT_CLASS;H_DROP_CAPLAYOUT_CLASS;H_GROUPLAYOUT_CLASS;H_HEADERLAYOUT_CLASS;H_NOTELAYOUT_CLASS;H_PAGELAYOUT_CLASS;H_ROWGROUPLAYOUT_CLASS;H_ROWLAYOUT_CLASS;H_RUBYLAYOUT_CLASS;H_SUPERTABLEGROUPLAYOUT_CLASS;H_SUPERTABLELAYOUT_CLASS;H_TABLELAYOUT_CLASS;H_ENDNOTELAYOUT_CLASS;H_FOOTNOTELAYOUT_CLASS;H_TABLEHEADINGLAYOUT_CLASS;H_TOCSUPERTABLELAYOUT_CLASS',0)} [See list of classes](#)

{button ,AL(`H_PAPERNAME_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write)

This language element has not yet been defined. You can check for a newer Help file on the Lotus web site.

Data Type

String

Syntax

papernamevalue = [objectreference].PaperName

[objectreference].PaperName = papernamevalue

Legal values

Any value of type String.

Usage

{button ,AL(`H_PAPERNAME_PROPERTY_MEMDEF',0)} [See related topics](#)

Word Pro: ParagraphBackground property

{button ,AL(^H_CLICKHERE_CLASS;H_TEXTMARKER_CLASS;H_PARAGRAPHSTYLE_CLASS;H_TEXT_CLASS',
0)} [See list of classes](#)

(Read-only)

This language element has not yet been defined. Lotus provides updated Help files on its web site.

Data Type

ParagraphBackground

Syntax

paragraphbackgroundvalue = [objectreference].ParagraphBackground

Legal values

Always contains an instance of the ParagraphBackground class.

Usage

Word Pro: ParagraphBorder property

{button ,AL(^H_CLICKHERE_CLASS;H_TEXTMARKER_CLASS;H_PARAGRAPHSTYLE_CLASS;H_TEXT_CLASS',
0)} [See list of classes](#)

{button ,AL(^H_PARAGRAPHBORDER_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-only)

Data Type

[ParagraphBorder](#)

Syntax

paragraphbordervalue = [objectreference].ParagraphBorder

Legal values

Always contains an instance of the ParagraphBorder class.

Usage

Word Pro: ParagraphHasDropCap property

{button ,AL('H_CLICKHERE_CLASS;H_TEXTMARKER_CLASS;H_TEXT_CLASS',0)} [See list of classes](#)

{button ,AL('H_PARAGRAPHHASDROPCAP_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-only)

Data Type

Integer.

Syntax

paragraphhasdropcapvalue = [objectreference].ParagraphHasDropCap

Legal values

Boolean.

Usage

Word Pro: ParagraphHasText property

{button ,AL(`H_CLICKHERE_CLASS;H_TEXTMARKER_CLASS;H_TEXT_CLASS',0)} [See list of classes](#)

{button ,AL(`H_PARAGRAPHHASTEXT_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-only) Indicates whether there are any Text fribs present in the current paragraph.

Data Type

Integer.

Syntax

paragraphhasertextvalue = [objectreference].ParagraphHasText

Legal values

Usage

```
'Example: Merge method
' This example merges data for the current merge document.
' RUNTIME DEPENDENCIES: You must have a document open which has been assigned
' to a merge data file has inserted merge fields for this script to work.

Dim stat As Integer

' Set up to merge and view
.ApplicationWindow.ActiveDocument.MergeOptions.Options = &H2
.ApplicationWindow.ActiveDocument.MergeOptions.MergeStepNumber = $LwpMergeStepNumber3
.MergeStart

Do
    stat = .Merge($LwpMergeActionNextRecord)
    .Merge $LwpMergeActionMergeOne
Loop Until stat = False
```

'Example: Messages method

' This example shows how to provide a default response to a dialog box. The

' MessageBox dialog is never displayed here.

' RUNTIME DEPENDENCIES: You must have a document open for this script to work.

.Messages \$LwpTakeDefaultMsgboxAnswer

MessageBox "Click OK ",MB_OK,"Example Script"

'Example: Minimize method

' This example minimizes the main application window.

.ApplicationWindow.Minimize

```

'Example: MirrorName property
' This example inserts two Click Here blocks in the current document.
'The second Click Here block repeats the first one.

' Insert the first block, and set its properties
MyClick = .InsertClickHere()
.Division.Foundry.ClickHeres(MyClick).Prompt.Clear
.Division.Foundry.ClickHeres(MyClick).Prompt.InsertText "Click here to type Text",
False, $LwpTextTypeNative
' This click here is the 'parent', so it doesn't mirror anything
.Division.Foundry.ClickHeres(MyClick).MirrorName = ""
.Division.Foundry.ClickHeres(MyClick).Action = 1
.Division.Foundry.ClickHeres(MyClick).HelpText = "Click here to type Text"
.Division.Foundry.ClickHeres(MyClick).UsesHelp = False
.Division.Foundry.ClickHeres(MyClick).TabOrder = 3
.Division.Foundry.ClickHeres(MyClick).TabExits = True
.Division.Foundry.ClickHeres(MyClick).ReturnExits = False
.Division.Foundry.ClickHeres(MyClick).Name = MyClick
.Division.Foundry.ClickHeres(MyClick).AllowListEdit = True
.Division.Foundry.ClickHeres(MyClick).AllowListMultiValues = False
.Division.Foundry.ClickHeres(MyClick).RemoveNamedProperty
.Division.Foundry.ClickHeres(MyClick).SetNamedProperty "Collect", "Off"
.Division.Foundry.ClickHeres(MyClick).SetNamedProperty "Required", "Off"
.Division.Foundry.ClickHeres(MyClick).SetNamedProperty "Notes/FX", "Off"
.UpdateUI

'Insert some text, then the CHB we will repeat.
.Text.InsertText "This is some sample text. ", False
.Text.CanRepeatClickHere MyClick.Name
RepeatClick = .InsertClickHere()
.Division.Foundry.ClickHeres(RepeatClick).Prompt.Clear
.Division.Foundry.ClickHeres(RepeatClick).Prompt.InsertText "Click here to type Text",
False, $LwpTextTypeNative
' Insert the Division Name and the name of the block to repeat, for
'example "Body:ClickHere1"
.Division.Foundry.ClickHeres(RepeatClick).MirrorName = .Container.DivisionName & ":" &
MyClick
.Division.Foundry.ClickHeres(RepeatClick).Action = 1
.Division.Foundry.ClickHeres(RepeatClick).HelpText = "Click here to type Text"
.Division.Foundry.ClickHeres(RepeatClick).UsesHelp = False
.Division.Foundry.ClickHeres(RepeatClick).TabOrder = 3
.Division.Foundry.ClickHeres(RepeatClick).TabExits = True
.Division.Foundry.ClickHeres(RepeatClick).ReturnExits = False
.Division.Foundry.ClickHeres(RepeatClick).Name = RepeatClick
.Division.Foundry.ClickHeres(RepeatClick).AllowListEdit = True
.Division.Foundry.ClickHeres(RepeatClick).AllowListMultiValues = False
.Division.Foundry.ClickHeres(RepeatClick).RemoveNamedProperty
.Division.Foundry.ClickHeres(RepeatClick).SetNamedProperty "Collect", "Off"

```

```
.Division.Foundry.ClickHeres (RepeatClick).SetNamedProperty "Required", "Off"  
.Division.Foundry.ClickHeres (RepeatClick).SetNamedProperty "Notes/FX", "Off"  
.UpdateUI
```

```
'Example: MirrorPage method
' This example sets up a left/right (complex) page.
' RUNTIME DEPENDENCIES: You must have a document open for this script to work.
.Page.Layout.RightPage.MarginLeft = 720
.Page.Layout.IsComplex = True
.Page.Layout.LeftPage.MirrorPage
```

```
'Example: ModifiedDateString property
'This example prints the date and time the document was modified,
'and the total editing time to the LotusScript output panel.
MDate = .ActiveDocument.Docinfo.ModifiedDateString
MTime = .ActiveDocument.Docinfo.ModifiedTimeString
ETime = .ActiveDocument.Docinfo.TotalEditingTime
Print "The current document was modified on " & MDate & " at " & MTime & ". " & ETime
& " minutes have been spent editing the document."
```



```
'Example: ModifiedTimeString property
'This example prints the date and time the document was modified,
'and the total editing time to the LotusScript output panel.
MDate = .ActiveDocument.Docinfo.ModifiedDateString
MTime = .ActiveDocument.Docinfo.ModifiedTimeString
ETime = .ActiveDocument.Docinfo.TotalEditingTime
Print "The current document was modified on " & MDate & " at " & MTime & ". " & ETime
& " minutes have been spent editing the document."
```

'Example: ModifiedTimeValue property
'This example determines the time between when the current document was
'created, and when it was last modified, and displays the results in a message box.
'In order to return a reasonable result, the document must have been saved
'at least once.

```
Dim Days As Long, Hours As Integer, Minutes As Integer
CTime = .ActiveDocument.DocInfo.CreationTimeValue
MTime = .ActiveDocument.DocInfo.ModifiedTimeValue
TotalSec = MTime - CTime ' The total number of seconds since document created
Days = Int(TotalSec / 86400) ' There 86,400 seconds in a day
Hours = Int((TotalSec Mod 86400) / 3600)
Minutes = Int((TotalSec Mod 3600) / 60)
Seconds = Int(TotalSec Mod 60)
MessageBox "This document is " & Days & " days " & Hours & " hours " & Minutes & "
minutes " & Seconds & " seconds old." , MB_OK, "Example Script"
```

```
'Example: MorphSelectionToTable method
' This example inserts 5 rows and 5 columns of text into the current document
' where each column is separated by a tab and each row by a paragraph break.
' The text is then selected and morphed into a table.
' RUNTIME DEPENDENCIES: You must have a document open for this script to work.
```

```
Dim RowCount as Integer
Dim ColumnCount as Integer
Dim OutputText as string
For RowCount = 1 To 5
    For ColumnCount = 1 To 5
        OutputText = "r" & RowCount & "c" & ColumnCount
        .Text.InsertText OutputText
        .Text.InsertTab
    Next
    .Text.SplitParagraph
Next
.Text.Shade $LwpLocationTypeParagraph,$LwpNavigateDirectionLeft,5
.Text.MorphSelectionToTable
```

```
'Example: MoveDivision method
' This example moves the division named 'Division1' to last division in the
' current document.
' RUNTIME DEPENDENCIES: You must have a document open containing 2 or more
' divisions with one of the divisions named 'Division1' for this script to
' work.
```

```
Dim Div As Division
```

```
Set Div = Bind("!Division1")
```

```
DivName = Div.Name
```

```
.MoveDivision DivName
```

```
'Example: MoveDown method
' This example inserts 5 lines into the current document and then moves the
' cursor up 4 lines and then down 2.
' RUNTIME DEPENDENCIES: You must have a document open for this script to work.
Dim LineNumber as Integer
For LineNumber = 1 To 5
    .Type("Line " & LineNumber)
    .Text.SplitParagraph
Next
MessageBox "Click OK to move up 4 lines.",MB_OK,"Example Script"
.Text.MoveUp(4)
MessageBox "Click OK to move down 2 lines.",MB_OK,"Example Script"
.Text.MoveDown(2)
```

```

'Example: MoveItem method
' This example adds a new menu item name 'New Menu' to the File menu just
' below the Save item. All items from the Edit menu are then moved to
' 'New Menu'.
' RUNTIME DEPENDENCIES: You must have not deleted the Edit or File menus
' for this script to work.

Dim MenuName as String
Dim SourceMenu As MenuItem
Dim DestinationMenu As MenuItem
Dim MenuSpacer as String
MenuSpacer = Chr$(8)
MenuName = "&New Edit"

' Set DestinationMenu to the File menu
Set DestinationMenu=.Applicationwindow.LwpMenuBar.Items.Item("&File")

' Set SourceMenu to the Edit Menu
Set SourceMenu = .ApplicationWindow.LwpMenuBar.Items.Item("&Edit")

' Create a new Edit Menu
DestinationMenu.DeleteItem "My Edit"
DestinationMenu.NewItem MenuName,,0,"&Save" & MenuSpacer & "Ctrl+S"

' Copy all the items from the Edit Menu to My new Menu
Forall Items In SourceMenu.Items
    DestinationMenu.Items(MenuName).MoveItem Items, True, ,
End Forall

```

```
'Example: MoveParagraph method
' This example inserts two paragraphs into the current document and then
' moves the second paragraph above the first.
' RUNTIME DEPENDENCIES: You must have a document open for this script to work.

.Type "This is the first paragraph inserted.[Enter]"
.Type "This is the second paragraph inserted. It "
.Type "will be moved before the first paragraph."
.Text.MoveParagraph $LwpMoveDirectionUp
```

```
'Example: MoveToEnd method
' This example inserts 5 sentences of text into the current document.
' The cursor is then positioned to the beginning of the paragraph, and then advanced
' to the next word, sentence, line and paragraph.
' RUNTIME DEPENDENCIES: You must have a document open and the cursor located
' in a multiline paragraph for this script to work.
```

```
Dim SentenceNumber as Integer
Dim WordNumber as Integer
For SentenceNumber = 1 To 5
    For WordNumber = 1 To 5
        .Text.InsertText "Word" & Format$(WordNumber) & " "
    Next
    .Type (". ")
Next
.Text.MoveToStart $LwpLocationTypeParagraph
.Text.MoveToEnd $LwpLocationTypeWord
.Text.MoveToEnd $LwpLocationTypeSentence
.Text.MoveToEnd $LwpLocationTypeLine
.Text.MoveToEnd $LwpLocationTypeParagraph
```



```
'Example: MoveToStart method
' This example inserts 5 sentences of text into the current document.
' The cursor is then positioned to the beginning of the paragraph.
' RUNTIME DEPENDENCIES: You must have a document open for this script to work.

Dim SentenceNumber as Integer
Dim WordNumber as Integer
For SentenceNumber = 1 To 5
    For WordNumber = 1 To 5
        .Text.InsertText "Word" & Format$(WordNumber) & " "
    Next
    .Type (". ")
Next
.Text.MoveToStart $LwpLocationTypeParagraph
```

```
'Example: MoveUp method
' This example inserts 5 lines into the current document and then moves the
' cursor up 4 lines.
' RUNTIME DEPENDENCIES: You must have a document open for this script to work.
Dim LineNumber as Integer
For LineNumber = 1 To 5
    .Type("Line " & LineNumber )
    .Text.SplitParagraph
Next
MessageBox "Click OK to move up 4 lines.",MB_OK,"Example Script"
.Text.MoveUp(4)
```

```
'Example: Move method
' This example moves the Word Pro's application window.
' RUNTIME DEPENDENCIES: You must have a document open for this script to work.

Dim XPosition as Long
Dim YPosition as Long
XPosition = 1440
YPosition = 2880
.ApplicationWindow.Move XPosition,YPosition
```

```
'Example: MultiCellPasteOn property
'This example asks the user whether to paste text into one or multiple cells,
'and then sets the appropriate option.

stat = MessageBox ("Do you want to paste all text into one cell?", 36, "Example
Script")
If stat = 6 Then ' user said yes
    .Application.Preferences.MultiCellPasteOn = False
Else
    .Application.Preferences.MultiCellPasteOn = True
End If
```

'Example: NameOfExternalBorder property

'This example creates an custom border for the current page layout.

'The border is then changed to another pattern.

```
.Page.Layout.RightPage.BorderLines.LinePlacement = &Hf
.Page.Layout.RightPage.BorderLines.AllBorders.Pattern = -32768
.Page.Layout.RightPage.BorderLines.AllBorders.NameOfExternalBorder =
"1998327121627.EDB"
.Page.Layout.RightPage.BorderLines.AllBorders.WidthInTwips = 720
.Page.Layout.RightPage.BorderLines.AllBorders.Color.Red = 176
.Page.Layout.RightPage.BorderLines.AllBorders.Color.Blue = 0
.Page.Layout.RightPage.BorderLines.AllBorders.Color.Green = 198
.Page.Layout.RightPage.BorderLines.AllBorders.Color.Override =
$LwpColorOverrideBlack
.Page.Layout.RightPage.Join.JoinType = -32768
.Page.Layout.RightPage.Join.Clear
.Page.Layout.RightPage.Join.JoinName = "1998326125038.EDB"
.Page.Layout.RightPage.Join.ScaleMode = $LwpJoinScaleMatchborder
.Page.Layout.RightPage.Join.Percentage = 30

MessageBox "Click OK to change the designer border.", MB_OK, "Example Script"
.Page.Layout.RightPage.BorderLines.AllBorders.NameOfExternalBorder = "\Borders\
emf\1713.emf"
.Page.Layout.RightPage.Join.Clear
.Page.Layout.RightPage.Join.JoinName = "\Borders\emf\17c013.emf"
```

'Example: NavigationLink property

'This example sets the NavigationLink property so that navigation arrows
'appear at the bottom of pages exported to HTML.

.ApplicationWindow.UserInterfacePrefs.HTMLOptions.**NavigationLink** = 1

'Example: NewDivision method

' This example creates a new division in the active document.

' RUNTIME DEPENDENCIES: You must have a document open for this script to work.

.NewDivision

'Example: NewDocument method

' This example creates a new document based on the 'DEFAULT.MWP' SmartMaster.

.NewDocument , , "DEFAULT.MWP", ,

'Example: NewFrame method

' This example creates a new frame 2 inches by 2 inches in the current
' document. The frame's upper left corner is positioned 1 inch from the left
' edge of the page and 2 inches down from the top of the page.
' RUNTIME DEPENDENCIES: You must have a document open for this script to work.

.NewFrame 2880, 2880, 1440, 2880

```
'Example: NewItem method
' This example creates a new menu item named 'Example Menu' to the File
' menu. The mnuMenuSub subroutine is assigned to run each time the new
' menu item is selected
' RUNTIME DEPENDENCIES: You must have a document open for this script to work.

Dim NewMenu As MenuItem
Dim MenuName as String
Dim MenuSpacer as String

MenuName = "&Example Menu"
MenuSpacer = Chr$(8)

' Set menu object
Set NewMenu = .ApplicationWindow.LwpMenuBar.Items.Item("&File")

' Create a new menu off of the File Menu and before the Save option
' Delete it first to prevent duplicates
NewMenu.DeleteItem MenuName
NewMenu.NewItem MenuName, "!mnuMenuSub", 0, "&Save" & MenuSpacer & "Ctrl+S"

Sub mnuMenuSub
' this is the sub which runs when the user selects the example menu.
    MessageBox "You selected the Example Menu", MB_OK, "Example Script"
End Sub
```

'Example: NewVersion property

'This example creates two documents, File1 and File2, with differences
'between them so that they can be compared.

'The two documents are then compared, with the differences placed into a
'new version

```
.NewDocument "", "", "default.mwp", "", "", ""  
.Type "This is the first version of the document"  
.SaveAs "file1.lwp", "", "Lotus Word Pro", False, True, False  
.Type "[Enter]This is some added text.[Up][Left][Left]"  
.Type "[Left][Left][Left][Left][Left][Left][Left][Left]"  
.Type "[shiftLeft][shiftLeft][shiftLeft][shiftLeft][shiftLeft]second"  
.SaveAs "file2.lwp", "", "Lotus Word Pro", False, True, False  
.Close  
.OpenDocument "file1.lwp", "", ""  
.ApplicationWindow.ReviewVersions.FilesToCompare = "file2.lwp"  
.ApplicationWindow.ReviewVersions.NewVersion = True  
.ApplicationWindow.ReviewVersions.NewFile = False  
.ActiveDocument.VersionManager.CreateVersion "Markup version"  
.ActiveDocument.RevisionMarkMode = 0  
.CompareFiles "file2.lwp", "", False, 1  
.ActiveDocument.RevisionMarkMode = 0
```

'Example: NewWindow method

' This example creates a new document window based upon the current document.
' A message box then prompts you to close the new window.
' RUNTIME DEPENDENCIES: You must have a document open for this script to work.

.NewWindow

```
Messagebox "Click OK to close the current window.",MB_OK,"Example Script"  
.CloseDocWindow
```

```
'Example: NextStyleName property
'This example types some text into the current document.
'The option to use a specific style following the current paragraph is set and
demonstrated.
.Type "This is some text using the Default Text Style.[Enter]"
.Text.Breaks.NextStyleName = "Bullet 1" ' specify the style name to use for the next
paragraph.
.Text.Breaks.UseNextStyle = True ' set the option to use the style name.
.Type "The next paragraph will be Bullet 1[Enter]"
.Type "This paragraph uses Bullet 1"
```

```
'Example: NoHyphenate property
'This example sets hyphenation options for a new document.

.NewDocument
For i = 1 To 20 ' put some text in to demonstrate
    .type "testing hyphenation "
Next
' turn on auto hyphenation
.ActiveDocument.DivisionOptions.HyphenationOptions.AutoHyphenate = True
.Text.Attributes.NoHyphenate = False ' turn on local hyphenation
MessageBox "Click OK to disable hyphenation", MB_OK, "Example Script"
.ActiveDocument.DivisionOptions.HyphenationOptions.AutoHyphenate = False
```

```
'Example: NormalParagraph property
'This example inserts some text in the current document, then tests to see
'there are local paragraph overrides. The paragraph is then centered,
'and the test is run again.

.Text.InsertText "This is an example of a paragraph. ", False
If .Text.NormalParagraph = True Then
    MessageBox "There are no local paragraph overrides in this text.", MB_OK, "Example
Script"
End If
.Text.Alignment.AlignmentType = $LtsAlignmentHorizCenter
If .Text.NormalParagraph = False Then
    MessageBox "There are local paragraph overrides in this text.", MB_OK, "Example
Script"
End If
```

```
'Example: NormalText method
' This example first inserts sample text in the current document, selects
' the paragraph, and makes that paragraph bold. The script then uses the
' NormalText method to return the selected paragraph to the normal attributes
' for that paragraph style.
' RUNTIME DEPENDENCIES: You must have a document open for this script to work.

.Text.InsertText "This is some sample text."
.SelectParagraph
.Bold
MessageBox "Click OK to revert to normal attributes", MB_OK, "Example Script"
.NormalText
```



```
'Example: Normal property
'This example inserts some bold text in the document, then removes the
'attributes from the text.

.Bold
.Text.InsertText "This is an example of bold text. ", False
.Text.Select $LwpSelectObjectTypeParagraph
MessageBox "Click OK to make this text normal.", MB_OK, "Example Script"
.Text.Font.Normal = True
```

```
'Example: NotesFlow property
'This example tests to see if NotesFlow is active in the current document,
'and prints the result in a message box.

If .ActiveDocument.DocControl.NotesFlow = True Then
    MessageBox "NotesFlow is currently active.", MB_OK, "Example Script"
Else
    MessageBox "NotesFlow is currently disabled.", MB_OK, "Example Script"
End If
```

```
'Example: NumberEveryNthLine property
'This example creates a new document, and sets line numbering to number every other
line, with a spacing of 1/4 inch per line.
.newdocument
Set LineNumbers = .Division.LineNumberOptions
LineNumbers.NumberWhichLines = $LwpLineNumberOptsSpecifiedLines
LineNumbers.NumberEveryNthLine = 2
LineNumbers.LinesSpacedEveryNthUnit = 360 ' 1/4 inch
```

'Example: NumberingPosition property

'This example assigns an outline level of 1 and a numbering position of 1

'to the Heading 1 paragraph style and makes it a heading style.

'It assigns an outline level and numbering position of 2 to the Heading 2 paragraph style,

'and makes it not a heading style.

```
.Division.Foundry.ParagraphStyles("Heading 1").DocumentLevel = 1
```

```
.Division.Foundry.ParagraphStyles("Heading 1").Heading = 1
```

'Assign the Default Outline numbering sequence, and set the numbering position to 1.

```
.Division.Foundry.ParagraphStyles("Heading 1").Bullet.Name = "Default Outline"
```

```
.Division.Foundry.ParagraphStyles("Heading 1").NumberingPosition = 1
```

```
.Division.Foundry.ParagraphStyles("Heading 2").DocumentLevel = 2
```

```
.Division.Foundry.ParagraphStyles("Heading 2").Heading = 0
```

```
.Division.Foundry.ParagraphStyles("Heading 2").Bullet.Name = "Default Outline"
```

```
.Division.Foundry.ParagraphStyles("Heading 2").NumberingPosition = 2
```

'Example: NumberOfVersions method

' This example prints the number of versions for the current document to the

' Lotus Script Output panel.

' RUNTIME DEPENDENCIES: You must have a document open for this script to work.

Print .ActiveDocument.VersionManager.NumberOfVersions(\$LwpVersionObjectType)

'Example: NumberWhichLines property

'This example creates a new document, and sets line numbering to number every other line, with a spacing of 1/4 inch per line.

.NewDocument

Set LineNumbers = .Division.LineNumberOptions

LineNumbers.**NumberWhichLines** = \$LwpLineNumberOptsSpecifiedLines

LineNumbers.NumberEveryNthLine = 2

LineNumbers.LinesSpacedEveryNthUnit = 360 ' 1/4 inch

'Example: Number property

'This example inserts several footnotes in the current document. The first
'footnote is then located using the Number property, and text is inserted
'into the note.

```
.InsertFootnote $LwpFnTypeAtBottomOfPage
.Type "This is footnote number 1.[Esc] This is some text."
.InsertFootnote $LwpFnTypeAtBottomOfPage
.Type "This is footnote number 2.[Esc]"
Forall Footnote In .Division.Foundry.Footnotes
    If Footnote.Number = 1 Then
        Footnote.GoTo
            .Text.InsertText "New text for footnote number one. "
    End If
End Forall
```

```
'Example: NumCharsInDoc property
'This example prints the number of pages, words, and characters to the LotusScript
output panel.
NPages = .ActiveDocument.Docinfo.NumPagesInDoc
NWords = .ActiveDocument.Docinfo.NumWordsInDoc
NChars = .ActiveDocument.Docinfo.NumCharsInDoc
Print "The current document contains " & NPages & " pages, with " & NWords & " words,
and " & NChars & " characters."
```



```
'Example: NumCols property
'This example creates newspaper columns and sets options for column breaks.
.Layout.NumCols = 3 ' go into a newspaper column format
.Type "This is some text in the column.[Enter]"
.Text.Breaks.IsColumnBreakBefore = True ' put a column break before the next paragraph
.Type "There is a column break before this paragraph.[Enter]"
.Text.Breaks.IsColumnBreakBefore = False
```

'Example: NumDecimalPlaces property

'This example creates a table, and types a number into the first cell.

'The number of decimal places is set to 3, and the number is displayed.

```
.CreateTable False, "Default Table", 4, 4
```

```
.Type "123.45678"
```

```
.Table.CurrentCell.NumericFormat.NumDecimalPlaces = 3
```

```
.type "[Down]"
```

```
'Example: NumericFormat property
'This example creates a table, and types a number in the first cell.
'It then changes the numeric format of the cell to percentage.

.CreateTable
.Type "123.45"
.CurrentCell.NumericFormat.FormatType = $LtsNumberFormatPercent
```

'Example: NumLinesOfSpaceAbove property

'This example sets the paragraph spacing to 1 1/2 lines of space.

.Text.InsertText "This is a sample paragraph of text.", True

.Text.InsertText "This is a sample paragraph of text."

.Text.Spacing.TypeAbove = \$LwpSpacingTypeDynamic

.Text.Spacing.**NumLinesOfSpaceAbove** = 30

'Example: NumLinesOfSpaceBelow property

'This example sets spacing below the paragraph to 1/2 line of text.

```
.Text.InsertText "This is a sample paragraph of text. "
```

```
.Text.Spacing.TypeBelow = $LwpSpacingTypeDynamic
```

```
.Text.Spacing.NumLinesOfSpaceBelow = 10
```

```
.Text.InsertText "This is a sample paragraph of text.", True
```

```
.Text.InsertText "This is a sample paragraph of text."
```

```
'Example: NumLinesOfSpace property  
'This example inserts a sample paragraph of text into the current document,  
'and sets the line spacing to double spacing.
```

```
For I = 1 to 4
```

```
    .Text.InsertText "This is an example paragraph to test spacing. "
```

```
Next
```

```
.Text.Spacing.NumLinesOfSpace = 40
```

'Example: NumOfRecentFiles property

'This example sets the number of recent files to display to four.

.ApplicationWindow.UserInterfacePrefs.**NumOfRecentFiles** = 4

```
'Example: NumPagesInDoc property
'This example prints the number of pages, words, and characters to the LotusScript
output panel.
NPages = .ActiveDocument.Docinfo.NumPagesInDoc
NWords = .ActiveDocument.Docinfo.NumWordsInDoc
NChars = .ActiveDocument.Docinfo.NumCharsInDoc
Print "The current document contains " & NPages & " pages, with " & NWords & " words,
and " & NChars & " characters."
```


'Example: NumUndoLevels property

'This example sets the number of Undo levels to five.

```
.Application.Preferences.NumUndoLevels = 5
```

'Example: NumWindowsViewingDoc property

'This example splits the document window several times. Each time the
'window is split, a message box displays indicating the number of splits.

```
MessageBox "There are " & .ActiveDocument.NumWindowsViewingDoc & " instances of this  
document displayed.", MB_OK, "Example Script"
```

```
.SplitWindow True
```

```
MessageBox "There are " & .ActiveDocument.NumWindowsViewingDoc & " instances of this  
document displayed.", MB_OK, "Example Script"
```

```
.SplitWindow False
```

```
MessageBox "There are " & .ActiveDocument.NumWindowsViewingDoc & " instances of this  
document displayed.", MB_OK, "Example Script"
```

```
'Example: NumWordsInDoc property
'This example prints the number of pages, words, and characters to the LotusScript
output panel.
NPages = .ActiveDocument.Docinfo.NumPagesInDoc
NWords = .ActiveDocument.Docinfo.NumWordsInDoc
NChars = .ActiveDocument.Docinfo.NumCharsInDoc
Print "The current document contains " & NPages & " pages, with " & NWords & " words,
and " & NChars & " characters."
```

Word Pro: MoveDivision method

{button ,AL('H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL('H_MOVEDIVISION_METHOD_EXSCRIPT',1)} [See example](#)

Moves the specified division to the place specified by the Parent, DivisionLocation, or NeighborName parameters. If you do not specify where the division should be moved, it is moved inside the current division.

Syntax

[objectreference].MoveDivision(DivisionName,[Parent,] [DivisionLocation,] [NeighborName])

Parameters

DivisionName

A String expression which specifies the internal name of the division you want to move.

Parent

A String expression representing the internal name of the division which you want to become the parent of the division you are moving. Optional parameter.

DivisionLocation

A String or Integer value which indicates where you want the division to be moved in relation to the currently active division. Data type is Variant, which allows the value of this parameter to be one of the three division locations listed below or its numeric equivalent (in parentheses). Default is \$LwpDivLocInsertAtInsertionPt.

<u>Value</u>	<u>Effect</u>
\$LwpDivLocInsertBeforeCurrentdiv (184)	Moves the division to a position before the currently active division.
\$LwpDivLocInsertAfterCurrentdiv (185)	Moves the division to a position after the currently active division.
\$LwpDivLocInsertAtInsertionPt (186)	Moves the division to the insertion point. All items that fall before the insertion point remain part of the active division. All items after the insertion point become part of the moved division.

Note If the insertion point is in a table cell or a frame, Word Pro splits the contents of the cell or frame, leaving the items before the insertion point intact and moving the items after the insertion point into the moved division. Items outside the cell or frame are not affected and remain in the original division.

NeighborName

A String expression representing the name of the division which you want to become the neighbor of the moved division. Optional parameter.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

For more information on division names in LotusScript, see [Overview: Division names in LotusScript](#)

Word Pro: MoveDown method

{button ,AL('H_CLICKHERE_CLASS;H_TEXTMARKER_CLASS;H_TEXT_CLASS',0)} [See list of classes](#)

{button ,AL('H_MOVEDOWN_METHOD_EXSCRIPT',1)} [See example](#)

Moves the insertion point down the specified number of times. Each down movement is equivalent to pressing the Down arrow.

Syntax

[objectreference].MoveDown(Count)

Parameters

Count

An Integer expression which specifies the number of times the insertion point is moved down.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Word Pro: MoveItem method

{button ,AL('H_MENUITEM_CLASS',0)} [See list of classes](#)

{button ,AL('H_MOVEITEM_METHOD_EXSCRIPT',1)} [See example](#)

Moves a menu item to another location within the same menu or to a different parent menu item.

Syntax

[objectreference].MoveItem(FromItem, [After,][TargetText,][Caption])

Parameters

FromItem

Specifies the menu item you want to move.

After

Default of True places the moved item after the item specified by the TargetText parameter. False places the moved item before the item specified by the TargetText parameter. Optional Boolean parameter.

TargetText

An optional String expression that allows you to specify any MenuItem object and position the moved item before or after it. If this parameter is omitted, the menu item will be moved to the end of the destination menu.

Caption

The name of the relocated menu item. You can use this optional String parameter to change the caption of a moved menu item.

Return value

The return value for this method will always be -1 or 0. When testing the return value, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Call this method to move menu items from one menu to another, or to reorder menu items within the same menu.

Word Pro: MoveParagraph method

{button ,AL('H_CLICKHERE_CLASS;H_TEXTMARKER_CLASS;H_TEXT_CLASS',0)} [See list of classes](#)

{button ,AL('H_MOVEPARAGRAPH_METHOD_EXSCRIPT',1)} [See example](#)

Moves the paragraph up or down by one paragraph.

Syntax

[objectreference].MoveParagraph(MoveDirection)

Parameters

MoveDirection

Specifies whether you want to move the paragraph up or down. Data type is Variant, which allows the value of this parameter to be one of the constants listed below or its numeric equivalent (in parentheses). There is no default value.

\$LwpMoveDirectionDown (1513)

\$LwpMoveDirectionUp (1512)

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Word Pro: MoveToBack method

```
{button ,AL(^H_LAYOUT_CLASS;H_CELLGROUPLAYOUT_CLASS;H_CELLLAYOUT_CLASS;H_CONNECTEDLAY  
OUT_CLASS;H_COLUMNGROUPLAYOUT_CLASS;H_FOOTERLAYOUT_CLASS;H_FRAMELAYOUT_CLASS;H  
_FRAMEGROUPLAYOUT_CLASS;H_DROP_CAPLAYOUT_CLASS;H_GROUPLAYOUT_CLASS;H_HEADERLAY  
OUT_CLASS;H_NOTELAYOUT_CLASS;H_PAGELAYOUT_CLASS;H_ROWGROUPLAYOUT_CLASS;H_ROWLA  
YOUT_CLASS;H_RUBYLAYOUT_CLASS;H_SUPERTABLEGROUPLAYOUT_CLASS;H_SUPERTABLELAYOUT  
_CLASS;H_TABLELAYOUT_CLASS;H_ENDNOTELAYOUT_CLASS;H_FOOTNOTELAYOUT_CLASS;H_TABLE  
HEADINGLAYOUT_CLASS;H_TOCSUPERTABLELAYOUT_CLASS',0)} See list of classes
```

Moves a layout to the back of all its siblings.

Syntax

```
[objectreference].MoveToBack()
```

Parameters

Return value

The return value for this method will always be -1 or 0. When testing the return value, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Use this method to reposition a layout object in relation to its sibling layout objects. For example, if you have a number of overlapping frames on a page, you can use this method on one of the FrameLayout objects to position it behind its sibling frames.

Equivalent to changing a frame's priority setting in Word Pro. In this situation, the placement of the frames must be set to "On current page" in order for the priority option to be enabled.

Word Pro: MoveToEnd method

{button ,AL('H_CLICKHERE_CLASS;H_TEXTMARKER_CLASS;H_TEXT_CLASS',0)} [See list of classes](#)

{button ,AL('H_MOVETOEND_METHOD_EXSCRIPT',1)} [See example](#)

Moves the caret or insertion point to the end of the nearest specified type of location.

Syntax

[objectreference].MoveToEnd(LocationType)

Parameters

LocationType

Specifies a type of location. See the descriptions below for the effect of each value. Data type is Variant, which allows the value of this parameter to be one of the constants listed below or its numeric equivalent (in parentheses). There is no default value.

<u>Value</u>	<u>Effect</u>
\$LwpLocationTypeDivision	
\$LwpLocationTypeDocument (573)	Moves the insertion point to the end of the current document.
\$LwpLocationTypeLine (571)	Moves the insertion point to the end of the current line.
\$LwpLocationTypeObject (2523)	
\$LwpLocationTypePage (2524)	
\$LwpLocationTypeParagraph (572)	Moves the insertion point to the end of the current paragraph.
\$LwpLocationTypeSelection (568)	Moves the caret to the end of the current selection without deselecting. If you select text by dragging from right to left and press ESC, the selection is deselected with the insertion point at the left. If you make the same selection and call this method with \$LwpLocationTypeSelection as the parameter, pressing ESC leaves the insertion point at the right.
\$LwpLocationTypeSentence (570)	Moves the insertion point to the end of the current sentence.
\$LwpLocationTypeStream (574)	Moves the insertion point to the end of the current text stream.
\$LwpLocationTypeWord (569)	Moves the insertion point to the end of the current word.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Word Pro: MoveToFront method

```
{button ,AL(^H_LAYOUT_CLASS;H_CELLGROUPLAYOUT_CLASS;H_CELLLAYOUT_CLASS;H_CONNECTEDLAY  
OUT_CLASS;H_COLUMNGROUPLAYOUT_CLASS;H_FOOTERLAYOUT_CLASS;H_FRAMELAYOUT_CLASS;H  
_FRAMEGROUPLAYOUT_CLASS;H_DROP_CAPLAYOUT_CLASS;H_GROUPLAYOUT_CLASS;H_HEADERLAY  
OUT_CLASS;H_NOTELAYOUT_CLASS;H_PAGELAYOUT_CLASS;H_ROWGROUPLAYOUT_CLASS;H_ROWLA  
YOUT_CLASS;H_RUBYLAYOUT_CLASS;H_SUPERTABLEGROUPLAYOUT_CLASS;H_SUPERTABLELAYOUT  
_CLASS;H_TABLELAYOUT_CLASS;H_ENDNOTELAYOUT_CLASS;H_FOOTNOTELAYOUT_CLASS;H_TABLE  
HEADINGLAYOUT_CLASS;H_TOCSUPERTABLELAYOUT_CLASS',0)} See list of classes
```

Moves a layout to the front of all its siblings.

Syntax

```
[objectreference].MoveToFront()
```

Parameters

Return value

The return value for this method will always be -1 or 0. When testing the return value, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Use this method to reposition a layout object in relation to its sibling layout objects. For example, if you have a number of overlapping frames on a page, you can use this method on one of the FrameLayout objects to position it in front of its sibling frames.

Equivalent to changing a frame's priority setting in Word Pro. In this situation, the placement of the frames must be set to "On current page" in order for the priority option to be enabled.

Word Pro: MoveToStart method

{button ,AL('H_CLICKHERE_CLASS;H_TEXTMARKER_CLASS;H_TEXT_CLASS',0)} [See list of classes](#)

{button ,AL('H_MOVETOSTART_METHOD_EXSCRIPT',1)} [See example](#)

Moves the caret or insertion point to the start of the nearest specified type of location.

Syntax

[objectreference].MoveToStart(LocationType)

Parameters

LocationType

Specifies a type of location. See the descriptions below for the effect of each value. Data type is Variant, which allows the value of this parameter to be one of the constants listed below or its numeric equivalent (in parentheses). There is no default value.

<u>Value</u>	<u>Effect</u>
\$LwpLocationTypeSelection (568)	Moves the caret to the start of the current selection without deselecting. If you select text by dragging from left to right and press ESC, the selection is deselected with the insertion point at the right. If you make the same selection, but call this method with \$LwpLocationTypeSelection as the parameter, pressing ESC leaves the insertion point at the left.
\$LwpLocationTypeWord (569)	Moves the insertion point to the start of the current word.
\$LwpLocationTypeLine (571)	Moves the insertion point to the start of the current line.
\$LwpLocationTypeSentence (570)	Moves the insertion point to the start of the current sentence.
\$LwpLocationTypeParagraph (572)	Moves the insertion point to the start of the current paragraph.
\$LwpLocationTypeStream (574)	Moves the insertion point to the start of the current text stream.
\$LwpLocationTypeDocument (573)	Moves the insertion point to the start of the current document.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Word Pro: MoveUp method

{button ,AL('H_CLICKHERE_CLASS;H_TEXTMARKER_CLASS;H_TEXT_CLASS',0)} [See list of classes](#)

{button ,AL('H_MOVEUP_METHOD_EXSCRIPT',1)} [See example](#)

Moves the insertion point up the specified number of times. Each up movement is equivalent to pressing the Up arrow.

Syntax

[objectreference].MoveUp(Count)

Parameters

Count

An Integer expression which specifies the number of times the insertion point is moved up.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Word Pro: Move method

{button ,AL('H_WINDOW_CLASS;H_APPLICATIONWINDOW_CLASS;H_STATUSBAR_CLASS;H_DOCWINDOW_CLASS;H_DIVISION_CLASS;H_TEXTDOCUMENT_CLASS',0)} [See list of classes](#)

{button ,AL('H_MOVE_METHOD_EXSCRIPT',1)} [See example](#)

This method is defined in the following classes:

[Division]

Allows you to move a division object.

[StatusBar]

Allows you to move the status bar.

[TextDocument]

Allows you to move a TextDocument object.

[Window]

Allows you to move a window object to the coordinates specified by the XPosition and YPosition parameters.

[Application Window]

Allows you to move the application window to the coordinates specified by the XPosition and YPosition parameters.

Syntax

[objectreference].ActiveDocWindow.Move(Name,[ParentName,] [BeforeNeighbor,] [NeighborName]

[objectreference].Division.Move(XPosition,YPosition)

[objectreference].ActiveDocument.Move(XPosition,YPosition)

[objectreference].IconBarManager.Move(XPosition,YPosition)

[objectreference].StatusBar.Move(XPosition,YPosition)

[objectreference].ApplicationWindow.Move(XPosition,YPosition)

Parameters

XPosition

Data type is Long but the unit of measurement used for this property is [Twips](#). There are 1440 Twips per inch.

YPosition

Data type is Long but the unit of measurement used for this property is [Twips](#). There are 1440 Twips per inch.

MoveName

Data type is String.

ParentName

Data type is String. Optional parameter.

BeforeNeighbor

Data type is Integer. The legal values for this parameter are -1 and 0 but you may use the LotusScript constants True (-1) and False (0). Optional parameter. Default is False (0).

NeighborName

Data type is String. Optional parameter.

Return value

The return value for this method will always be -1 or 0. When testing the return value, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Word Pro: NewDivision method

{button ,AL('H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL('H_NEWDIVISION_METHOD_EXSCRIPT',1)} [See example](#)

Creates a new division in the active document.

Syntax

[objectreference].NewDivision([ParentName,] [BeforeNeighbor,] [NeighborName,] [Initialize,][Split])

Parameters

ParentName

A String expression which allows you to specify the internal name of the parent for the new division. Optional parameter.

BeforeNeighbor

An Integer expression which allows you to specify whether you want the new division to be placed before the neighbor or after the neighbor. The legal values for this parameter are -1 and 0, but you may use the LotusScript constants True (-1) and False (0). Optional parameter. Default is False.

NeighborName

A String expression which allows you to specify the internal name of the neighbor for the new division. Optional parameter.

Initialize

An Integer expression which allows you to copy all the styles from the neighboring division into the new division. The legal values for this parameter are -1 and 0 but you may use the LotusScript constants True (-1) and False (0). Optional parameter. Default is True, which copies all the styles from the neighboring division.

Split

An Integer expression which allows you to split the current division at the insertion point and place the new division at the split. The legal values for this parameter are -1 and 0, but you may use the LotusScript constants True (-1) and False (0). Optional parameter. Default is False, which does not split the current division.

Return value

A String expression which represents the internal name of the external division.

For more information on division names in LotusScript, see [Overview: Division names in LotusScript](#)

Usage

Word Pro: NewDocument method

{button ,AL(^H_APPLICATION_CLASS;H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL(^H_NEWDOCUMENT_METHOD_EXSCRIPT',1)} [See example](#)

Creates a new Word Pro document, including the DocWindow and TextDocument objects.

Syntax

[objectreference].NewDocument([DocFileName,][DocFilePath,] [SmartMasterName,][SmartMasterDir,] [Password,] [Kind])

Parameters

DocFileName

An optional String expression representing the name of the new file. You can leave this parameter blank if you want to open an untitled document.

DocFilePath

An optional String expression representing the path where you want to store the new file.

SmartMasterName

An optional String expression representing the name of the SmartMaster from which you want to create the new document. To create a plain document, leave this parameter empty.

SmartMasterDir

An optional String expression representing the path where the SmartMaster for the new document is stored. If the SmartMaster you want to use is stored in the default SmartMaster directory, you do not have to include this parameter.

Password

An optional String expression representing the password you want to assign the new document. Providing a value for this parameter activates the password feature for the new document. If you lose or forget the password, the document cannot be opened.

Kind

An optional String expression representing the file format for the new document.

Return value

Long.

Usage

Use this method to create a new document.

Word Pro: NewFrame method

{button ,AL('H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL('H_NEWFRAME_METHOD_EXSCRIPT',1)} [See example](#)

Inserts a new frame into the active Word Pro document.

Syntax

[objectreference].NewFrame([Width], [Height], [X], [Y], [FrameStyle], [Columns], [Rows])

Parameters

Width

The width of the new frame expressed in Twips. Use a null string ("") if you want Word Pro to use the width specified in the frame style.

Height

The height of the new frame expressed in Twips. Use a null string ("") if you want Word Pro to use the height specified in the frame style.

X

The position, expressed in Twips, of the upper left corner of the frame on the X axis. Use a null string ("") if you want Word Pro to use the X coordinate specified in the frame style.

Y

The position, expressed in Twips, of the upper left corner of the frame on the Y axis. Use a null string ("") if you want Word Pro to use the Y coordinate specified in the frame style.

FrameStyle

A String, expressing the name of the style from which the new frame should be created. If you do not specify a frame style, Word Pro uses the default frame style.

Columns

Data type is Long.

Rows

Data type is Long.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

If you use the recorder to get the code for creating a new frame, the recorder uses NewFrame when you use the new frame icon, and CreateFrame when you choose Create - Frame.

Word Pro: NewItem method

{button ,AL('H_MENUITEM_CLASS',0)} [See list of classes](#)

{button ,AL('H_NEWITEM_METHOD_EXSCRIPT',1)} [See example](#)

Adds a new child MenuItem object to a parent menu.

Syntax

[objectreference].NewItem(Caption, [p2,] [After,] [TargetText])

Parameters

Caption

A String parameter representing the name of the new menu item.

p2

An optional Variant parameter that initializes the Action property. Default is 0.

prexAfter

The default of True places the new item after the item specified by the TargetText parameter. A value of False places the new item before the item specified by the TargetText parameter. Optional Boolean parameter.

prexTargetText

An optional String expression that allows you to specify a menu item object so you can place the new item before or after it. If you are trying to place your new item before or after an existing Word Pro item that displays an accelerator keystroke on the menu, you must build the TargetText string. Use the LotusScript function, Chr\$(8), with the text on the menu to attach the accelerator keystroke to an existing item. For example, to attach to the Save menu item accelerator keystroke (in the File menu) you can use:

```
MyTargetText$ = "&Save" & Chr$(8) & "Ctrl+S"
```

If this parameter is omitted, the menu item is moved to the end of the destination menu.

Return value

This method returns the new MenuItem object.

Usage

This method is used to add new menu items to Word Pro menus and to create new custom menus. Each time the NewItem method is executed, a new menu item object is created and added to the Items property of the parent menu item.

Word Pro: NewWindow method

{button ,AL('H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL('H_NEWWINDOW_METHOD_EXSCRIPT',1)} [See example](#)

Opens another window for the currently active document. The new window is not a copy or separate version of the document. It is the same document displayed in a different window. Changes made to the document in one window are immediately reflected in the other window(s).

Syntax

[objectreference].NewWindow()

Parameters

None.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Depending on the property settings, may be equivalent to Window - Tile Left-Right (horizontal), or Window - Tile Top-Bottom (vertical).

Word Pro: NextToObject method

{button ,AL(^H_CLICKHERE_CLASS;H_TEXTMARKER_CLASS;H_TEXT_CLASS',0)} [See list of classes](#)

Determines whether or not the insertion point is next to a Text subobject of the specified type.

Note For details on Text subobjects, see [Overview: Word Pro Text Subobjects](#).

Syntax

[objectreference].NextToObject(P1, StayThere)

Parameters

P1

A String expression which specifies the type of Text subobject for which you want to check. If the insertion point is located at the end of the specified type of subobject, Word Pro may move the insertion point to the beginning of the subobject to be sure the subobject is of the type you specify. Use the StayThere parameter to tell Word Pro to leave the insertion point at the front of the subobject, if it has to move there.

<u>Value</u>	<u>Effect</u>
AnchoredFrame	A frame placed "In Text," "In Text - Vertical," or "With Paragraph Above."
Bookmark	The marker for either the start or end of a bookmark range.
ClickHereBlock	The marker for either the start or end of a ClickHereBlock.
ColumnBreak	A hard column break within a multi-column page layout, created by choosing Text - Insert Other - Column Break.
DDE	The marker for either the start or end of a DDE link.
DocVariable	A DocInfo field
EOP	The marker at the end of a paragraph created by pressing ENTER.
FootnoteMark	The marker for either the start or end of a footnote.
HardSpace	A non-breaking space created by pressing CTRL+Spacebar.
HKatakana	A special Asian language character. Found only in Asian language versions of Word Pro.
Kanji	A special Asian language character. Found only in Asian language versions of Word Pro.
LineBreak	A soft line break created by pressing SHIFT+ENTER.
Note	A comment note created by choosing Create - Comment Note.
PageBreak	A hard page break created by pressing CTRL+ENTER.
PageNumber	A page number created by choosing Text - Insert Page Number.
ParaNumber	Any number, bullet, or other text assigned to a paragraph as part of a numbering/bullet style. Note that any outline number sequence which uses decimal notation (such as 1.1.1, 1.1.2, 1.1.3, and so on) contains one ParaNumber subobject for each digit in the sequence and one Text subobject for each decimal in the sequence. These subobjects are separate from the subobject(s) which comprise the text of the paragraph itself.
PowerField	The marker for either the start or end of a PowerField.
RubyFrame	Found only in Asian language versions of Word Pro.
RubyMarker	Found only in Asian language versions of Word Pro.
Section	A section marker which defines the separation of two

	sections in a Word Pro document.
SoftHyphen	A hyphen which does not appear unless the hyphenated word flows to the end of a line where it would break naturally.
SpecialTab	Special tabs contain their own tab stop and alignment information. Normal tab markers derive their tab stop and alignment information from the ruler. SpecialTab markers are created in two ways: the InsertSpecialTab method or by importing a document from a file format which uses SpecialTabs (such as WordPerfect).
Tab	A normal tab marker created by pressing TAB.
Table	A table which is placed "In Text," "In Text - Vertical," or "With Paragraph Above."
Text	Text characters as seen in the user's code page. Users running the US English version of Word Pro under Windows 95 usually use the ANSI 1252 code page.
Tombstone	A revision marker which displays the initials of the author of a particular revision.
Unicode	A special foreign-language type of character used in multi-language documents.

StayThere

An integer which specifies whether you want Word Pro to leave the insertion point at the front of the subobject. Use a value of -1 (True) to leave the insertion point in front of the subobject.

Return value

This method returns a value of 1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Word Pro: NextVersion method

{button ,AL(^H_VERSIONMANAGER_CLASS',0)} [See list of classes](#)

Changes the active version to the next version.

Syntax

[objectreference].NextVersion()

Parameters**Return value****Usage**

Word Pro: Next method

{button ,AL(^H_CLICKHERE_CLASS;H_TEXTMARKER_CLASS;H_TEXT_CLASS',0)} [See list of classes](#)

Selects the next specified object type in a Text , TextMarker, or ClickHere object.

Note The Next method as seen on any of the Layout classes is for internal use only. This holds true for the Next method in any of the derived Layout classes, such as TableLayout or FrameLayout. There is no documentation for the Next method on Layout or any of Layout's derived classes.

Syntax

When called from a Text, TextMarker, or ClickHere object.

[objectreference.]Next(ObjectType, [SearchObjectType],[p3],[MarkerName],[ClassName],[SubClass,])

Parameters

ObjectType

Specifies the type of object you want to find. Data type is Variant, which allows the value of this parameter to be one of the constants listed below or its numeric equivalent (in parentheses).

<u>Value</u>	<u>Effect</u>
\$LwpNextObjectTypeRevision (1527)	Selects the next revision. If you use this value you must also use 18 for the value of the SearchObjectType parameter. If the insertion point is located within a revision, this method selects that revision. Equivalent to choosing the Next Revision button on the Revision bar.
\$LwpNextObjectTypeSearch (1526)	Selects the next object of the type specified in the SearchObjectType and ClassName parameters.
\$LwpNextObjectTypeTombstone (1528)	Ignores all other parameters. Searches for TeamConsolidate tags.
\$LwpNextObjectTypeFormatcheck (1529)	Selects the next item flagged by Format Check.

SearchObjectType

Specifies what type of object you want to search for. When ObjectType is \$LwpNextObjectTypeRevision, the value of this parameter must be 18. Data type is Variant, which allows the value of this parameter to be one of the constants listed below or its numeric equivalent (in parentheses). There is no default value.

<u>Value</u>	<u>Effect</u>
\$LwpSelectObjectTypeObject (1736)	Searches for the next object of the type specified in the ClassName and SubClass parameters.
\$LwpSelectObjectTypeAlignment (1739)	Searches for the next instance of a local alignment setting. Local alignment settings are those which differ from the alignment defined in the paragraph style. No values are needed for ClassName or SubClass.
\$LwpSelectObjectTypeIndent (1740)	Searches for the next instance of a local indent setting. Local indent settings are those which differ from the indent setting defined in the paragraph style. No values are needed for ClassName or SubClass.
\$LwpSelectObjectTypeSpacing (1741)	Searches for the next instance of a local space setting. Local space settings are those which differ

from the space setting defined in the paragraph style. No values are needed for ClassName or SubClass.

\$LwpSelectObjectTypeTab (1742)

Searches for the next instance of a local tab ruler setting. Local tab ruler settings are those which differ from the tab ruler setting defined in the paragraph style. No values are needed for ClassName or SubClass.

p3

If the ObjectType parameter is \$LwpNextObjectTypeRevision and you want to restrict the Next method's operation to a specific range in your document, you can use this parameter to provide the name of the range in which you want to find the next revision. Always an optional parameter.

If the ObjectType parameter is \$LwpNextObjectTypeSearch and the insertion point is within the type of object you are searching for, you can use p3 to tell Word Pro to skip the object your insertion point is in and go to the next one. You must use an integer value of 1(True) to skip the current object or 0 (False) to select the current object. You must provide a value for this parameter. Default is 1 (True).

MarkerName

If the ObjectType parameter is \$LwpNextObjectTypeSearch and you want to restrict the Next method's search to a specific range in your document, you can use the MarkerName parameter to provide the name of the range in which you want to search for the next object. Always an optional parameter.

ClassName

If the ObjectType parameter is \$LwpNextObjectTypeSearch, you must provide a value for the ClassName parameter to specify the type of object for which you are searching. Data type for ClassName is String and the value must be one of the values listed in the table below.

Value	Effect
PageBreak	Finds and selects the next hard page break. To find the next page layout change, use PageBreak for ClassName and Layout for SubClass.
PowerField	Finds and selects the next Power Field. Index and TOC entries are types of powerfields. To find only the next Index or TOC entry, use PowerField for ClassName and Index or TOC for SubClass.
BookMark	Finds and selects the next bookmark.
FootnoteMark	Finds and selects the next footnote.
ColumnBreak	Finds and selects the next column break.
AnchoredFrame	Finds and selects the next frame.
LineBreak	Finds and selects the next line break.
HardSpace	Finds and selects the next non-breaking space (CTRL + [space]).
SoftHyphen	Finds and selects the next soft hyphen (CTRL + -).
Note	Finds and selects the next Comment Note.
DDE	Finds and selects the next DDE marker.
PageNumber	Finds and selects the next page number.
Table	Finds and selects the next table.
DocVariable	Finds and selects the next document field. To see the document fields, choose Text - Insert Other - Power/Doc. Field. Select "Document Field."

Note These values are case-sensitive.

SubClass

Allows you to search for the next change in page layout or the next instance of a TOC or Index field. Data type is

String.

Value	Effect
Layout	Page layout changes are seen as a type of page break. To search for the next change in page layout, specify PageBreak as the value for the ClassName parameter, and then specify Layout as the value for SubClass.
Index	Index entries are a type of Power Field. To search for the next instance of an Index entry, specify PowerField as the value for the ClassName parameter, and Index as the value for SubClass.
TOC	TOC entries are a type of Power Field. To search for the next instance of an TOC entry, specify PowerField as the value for the ClassName parameter, and TOC as the value for SubClass.

Note These values are case-sensitive.

Return value

If ObjectType is \$LwpNextObjectTypeRevision, the return value of this method is always 1.

If ObjectType is \$LwpNextObjectTypeSearch, this method returns a value of -1 (True) or 0 (False), indicating that the method succeeded or failed respectively.

Usage

Word Pro: NormalText method

{button ,AL('H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL('H_NORMALTEXT_METHOD_EXSCRIPT',1)} [See example](#)

Removes local attributes from text in a document. Reverts the text attributes back to the attributes defined in the paragraph style. Equivalent to choosing Text - Normal.

Syntax

[objectreference].NormalText()

Parameters

None.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Word Pro: NumberOfVersions method

{button ,AL('H_VERSIONMANAGER_CLASS',0)} [See list of classes](#)

{button ,AL('H_NUMBEROFVERSIONS_METHOD_EXSCRIPT',1)} [See example](#)

Displays the number of versions of a document. Equivalent to choosing File - Versions.

Syntax

[objectreference].NumberOfVersions(VersionObjectType)

Parameters

VersionObjectType

Data type is Variant. The value of this parameter must be \$LwpVersionObjectType or its code equivalent (1961).

Return value

Usage

Word Pro: OpenDataFile method

{button ,AL('H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL('H_OPENDATAFILE_METHOD_EXSCRIPT',1)} [See example](#)

Opens the specified Merge data file.

Syntax

[objectreference].OpenDataFile(DataFilePath, DataFileType, [Password])

Parameters

DataFilePath

A String expression specifying the name and file path of the data file to open.

DataFileType

A String expression specifying file type of the data file to open.

Password

A String expression specifying the password for the data file. Use this parameter only if the data file is password-protected. Optional parameter.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Word Pro: OpenDocument method

{button ,AL('H_APPLICATION_CLASS;H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL('H_OPENDOCUMENT_METHOD_EXSCRIPT',1)} [See example](#)

Opens a document in Word Pro.

Syntax

[objectreference].OpenDocument(DocName, [Location,] [Kind,] [Password,] [OpenRO,] [MakeVisible,] [OpenFlags,] [Range,] [Styles])

Parameters

DocName

A String expression representing the name of the document to open.

Location

An optional String expression representing the path of the document you are opening. If the document is not in the current working directory, you must use this parameter.

Kind

An optional String expression representing the file format for the document you are opening. You must provide the file format if the document is not a Word Pro document.

Password

An optional String expression which provides the password for files that are password-protected.

OpenRO

An optional Boolean expression indicating whether you want to open the file as Read-write (False) or Read-only (True). Default is False.

MakeVisible

An optional Boolean expression indicating whether or not the document is visible when opened. Optional parameter. Default is True.

OpenFlags

This bitmask parameter has a default value of 0. The data type for this parameter is Variant which allows the value of this parameter to be one of the constants listed below or its numeric equivalent (in parentheses). You can combine these values when you want Word Pro to combine the features listed below. Use the OR operator to combine values.

<u>Value</u>	<u>Effect</u>
LwpDocOpenNoUi (&H1)	
LwpDocOpenOleFileOpenMode (&H2)	
LwpDocConverted (&H4)	
LwpDocRefresh (&H8)	

Range

No default. String

Styles

No default. String

Return value

Long.

Usage

Use this method to open an existing document in Word Pro.

Word Pro: OpenFromStorage method

{button ,AL(^H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

Opens an embedded OLE Word Pro document from a specified IStorage.

Note OLE is not supported under OS/2.

Syntax

[objectreference].OpenFromStorage(plStorage, FileType, [Password])

Parameters

plStorage

A Numeric expression which specifies the IStorage space in which the embedded Word Pro object is stored. Data type is Long.

File Type

A String expression indicating the type of Word Pro or Ami Pro object being opened. The file types include:

Lotus Ami Pro

Lotus Ami Pro 3.x Macro

Lotus Ami Pro 3.x Styles

Lotus Word Pro

Lotus Word Pro SmartMaster

Password

A String expression which specifies the password for the embedded Word Pro document. Required only if the embedded Word Pro document is password protected.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Word Pro: OpenObject method

{button ,AL('H_CLICKHERE_CLASS;H_TEXTMARKER_CLASS;H_TEXT_CLASS',0)} [See list of classes](#)

Opens the comment note at the insertion point.

Syntax

[objectreference].OpenObject()

Parameters

None.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

A comment note is represented in LostusScript by a NoteContainer object, which contains several class members and objects, including a NoteLayout object, a DivisionInfo object, and a Presentation object.

If there is no comment note at the insertion point, Word Pro does nothing.

Word Pro: Open method

{button ,AL(^H_WINDOW_CLASS;H_APPLICATIONWINDOW_CLASS;H_STATUSBAR_CLASS;H_DOCWINDOW_CLASS;H_NOTELAYOUT_CLASS;H_OLEOBJECT_CLASS',0)} [See list of classes](#)

[StatusBar]

Opens/displays the status bar.

[ApplicationWindow]

Opens the application window and brings it to the top.

[OleObject]

Opens the OLE object.

Note This method is not implemented for OleObject within OS/2.

[NoteLayout]

Opens a specific note layout object.

Syntax

[objectreference].NoteLayout.Open()

[objectreference].OleObject.Open(Verb)

[objectreference].Window.Open()

[objectreference].StatusBar.Open()

[objectreference].ApplicationWindow.Open()

Parameters

Verb

Data type is Long. Optional parameter. Default is 0.

Return value

The return value for this method will always be -1 or 0. When testing the return value, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Word Pro: OutlineMoveTextDown method

{button ,AL(^H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL(^H_OUTLINEMOVETEXTDOWN_METHOD_EXSCRIPT',1)} [See example](#)

Moves the currently active paragraph down one paragraph. The paragraph remains at the same outline level.

Syntax

[objectreference].OutlineMoveTextDown()

Parameters

None.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Word Pro: OutlineMoveTextUp method

{button ,AL(^H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL(^H_OUTLINEMOVETEXTUP_METHOD_EXSCRIPT',1)} [See example](#)

Moves the currently active paragraph up one paragraph. The paragraph remains at the same outline level.

Syntax

[objectreference].OutlineMoveTextUp()

Parameters

None.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Word Pro: OutlineStyleSequence method

{button ,AL('H_PARAGRAPHSTYLE_CLASS',0)} [See list of classes](#)

Applies paragraph styles to outline sequences.

Syntax

[objectreference].OutlineStyleSequence(ParagraphGetType)

Parameters

ParagraphGetType

Data type is Variant. The value of this parameter must be \$LwpParagraphGetType or its code equivalent (1628).

Return value

Usage

Word Pro: PageDown method

{button ,AL(`H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL(`H_PAGEDOWN_METHOD_EXSCRIPT',1)} [See example](#)

Moves the insertion point down one page.

Syntax

[objectreference].PageDown()

Parameters

None.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Word Pro: PageUp method

{button ,AL(`H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL(`H_PAGEUP_METHOD_EXSCRIPT',1)} [See example](#)

Moves the insertion point up one page.

Syntax

[objectreference].PageUp()

Parameters

None.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Word Pro: PasteLink method

{button ,AL('H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL('H_PASTELINK_METHOD_EXSCRIPT',1)} [See example](#)

Pastes the contents of the Clipboard into the active Word Pro document as a linked OLE object.

Note OLE is not supported under OS/2.

Syntax

[objectreference].PasteLink(PresentationFormat, [UpdateDataOnly,] [IconMetaFilePictHandle])

Parameters

PresentationFormat

A String expression that indicates the Clipboard format to use for presenting the linked object in your Word Pro document. In most circumstances, you should use LWPOLEFormat. However, there may be times when you want to use one of the other formats to ensure that certain features will be available to you. For example, if you are pasting spreadsheet data into a Word Pro table and you want to update data without changing the cell formatting, you must use Rich Text Format for this parameter. Presentation formats include the following:

<u>Value</u>	<u>Effect</u>
LWPOLEFormat	The normal setting for pasted links.
CF_METAFILEPICT	Windows Metafile
CF_BITMAP	Windows Bitmap
CF_DIB	Device Independent Bitmap
CF_TEXT	Text
Rich Text Format	Rich Text Format

UpdateDataOnly

Indicates whether or not you want the cell format of a linked Word Pro table to be updated along with the data in the cells. Only available for spreadsheet data which is pasted into a Word Pro table and linked through OLE to the original spreadsheet. A value of True (-1) causes OLE to update only the data in the table cells. Any local formatting done to the Word Pro table remains intact. Data type is Integer. The legal values for this parameter are -1 and 0 but you may use the LotusScript constants True (-1) and False (0) instead of the integer values. Optional parameter. Default is False (0).

Note If you use a value of True for this parameter, you must use Rich Text Format as the value of the PresentationFormat parameter.

IconMetaFilePictHandle

An optional Numeric expression that allows you to specify which icon to use in representing the linked object in the Word Pro file. Using any value other than 0 automatically tells Word Pro to display the linked object as an icon. This value serves as a numeric handle (known as the HGLOBAL) to the metafile pict for that icon. You can get the HGLOBAL for a specific metafile pict by using the appropriate Windows API calls. Data type must be Long. Default is 0, which indicates that you want the contents of the linked OLE object to display in the Word Pro document.

Caution If you record the process of choosing Edit - Paste Special and linking a file to be displayed as an icon, Word Pro records a value for IconMetaFilePict that is valid only during the recording. When you play back the recorded script, the IconMetaFilePict value is invalid and Word Pro treats the value as if you passed 0.

Return value

Usage

Word Pro: PasteSpecial method

{button ,AL('H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL('H_PASTESPECIAL_METHOD_EXSCRIPT',1)} [See example](#)

Pastes an OLE object into a Word Pro document. Equivalent to choosing Edit - Paste Special from the object's source application to the Word Pro application.

Note OLE is not supported under OS/2.

Syntax

[objectreference].PasteSpecial(Format, IconMetaFilePictHandle)

Parameters

Format

A String expression indicating the format of the OLE object.

<u>Value</u>	<u>Source of OLE object</u>
CF_TEXT	ASCII text
Rich Text Format	Rich text format
Biff	MS Excel versions earlier than 3
Biff3	MS Excel 3
Biff4	MS Excel 4
LotusChart	LotusChart
CF_METAFILEPICT	Windows Metafile
CF_DIB	Device independent bitmap
CF_BITMAP	Bitmap
WordProDraw	WordProDraw
WordProEquation	WordProEquation
WordProNative	Internal paste from AppFoundry
FileName	OLE file
Embed Source	Creates an OLE embedded object from any object that can become an OLE embedded object.
Embedded Object	Creates an OLE embedded object from an existing OLE embedded object.
Link Source	Creates a linked OLE object.
Link	Creates a DDE link.

IconMetaFilePictHandle

A Numeric expression which allows you to specify which icon to use in representing the new OLE object. Optional parameter. Using any value other than 0 automatically tells Word Pro to display the new object as an icon. This value serves as a numeric handle (known as the HGLOBAL) to the metafile pict for that icon. You can get the HGLOBAL for a specific metafile pict by using the appropriate Windows API calls. Data type must be Long. Default is 0, which indicates that you want Word Pro to display the contents of the new OLE object.

Caution If you record the process of creating a new OLE object to be displayed as an icon, Word Pro records a value for IconMetaFilePict, which is valid only during the recording. When you play back the recorded script, the IconMetaFilePict value is invalid and Word Pro treats the value as if you passed 0.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Word Pro: Paste method

{button ,AL(^H_WPAPPLICATION_CLASS;H_DOCUMENT_CLASS;H_TEXTDOCUMENT_CLASS;H_FOUNDRY_CLASS',0)} [See list of classes](#)

{button ,AL(^H_PASTE_METHOD_EXSCRIPT',1)} [See example](#)

This method is defined in the following classes:

[WPAApplication]

Pastes the contents of the Clipboard into the currently active document at the insertion point.

[Foundry]

Pastes all the object(s) contained in the specified Foundry object into the Foundry object from which the Paste method is called.

Syntax

[objectreference].WPAApplication.Paste([MakeVisible])

[objectreference].Document.Paste([MakeVisible])

[objectreference].Foundry.Paste([FoundryType,][Clear])

Parameters

[Document, WPAApplication]

MakeVisible

Do not use this parameter. This parameter is not implemented in this release of Word Pro.

FoundryType

Used when calling this method from a Foundry object, this parameter specifies which Foundry object you want to paste from. Data type is Variant, which allows the value of this parameter to be one of the string values listed below or its numeric equivalent (in parentheses). Default is \$LwpFoundryTypeDocument.

<u>Value</u>	<u>Effect</u>
\$LwpFoundryTypeApplication (346)	Indicates the Foundry object found in the AppFoundry property on WPAApplication. Pastes all objects currently in WordPro.AppFoundry into the Foundry object from which the script calls the Paste method. For example, if you use this value to call this method from a division called Summary, all objects in AppFoundry are pasted into the Summary division's Foundry.
\$LwpFoundryTypeDocument (345)	Indicates the same Foundry object from which the script calls the Paste method. For example, if you use this value to call this method from a division called Summary, all objects in the Summary division's Foundry are pasted into the Summary division's Foundry a second time, thus creating duplicates of each object in the Summary division's Foundry object.
\$LwpFoundryTypeTemporary (347)	Indicates the Foundry object found in the TempFoundry property on WPAApplication. Pastes all objects currently in WordPro.TempFoundry into the Foundry object from which the script calls the Paste method. For example, if you use this value to call this method from a division called Summary, all objects in TempFoundry are pasted into the Summary division's Foundry.

Clear

Used when calling this method from a Foundry object, this parameter clears the source Foundry of the pasted

object(s). For example, if FoundryType has a value of \$LwpFoundryTypeApplication (or 346), setting this parameter to False (0) leaves all the AppFoundry objects in place after the Paste method is executed. A value of True (-1) causes all objects in AppFoundry to be cleared after the Paste method is executed. Data type is Integer, but the legal values are True (-1) and False (0). Default is True (-1).

Return value

When used with WPAApplication and Document, this method returns a value of -1 (True) or 0 (False), indicating that the method succeeded or failed respectively.

Foundry objects return a value of type Variant.

Usage

Word Pro: PColConnectCells method

{button ,AL(^H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL(^H_PCOLCONNECTCELLS_METHOD_EXSCRIPT',1)} [See example](#)

Connects the selected parallel column table cells.

Syntax

[objectreference].PColConnectCells()

Parameters

None.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Word Pro: PColConnectRows method

{button ,AL('H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL('H_PCOLCONNECTROWS_METHOD_EXSCRIPT',1)} [See example](#)

Connects the cells in the parallel column row that has the focus. If cells from more than one row are selected, Word Pro connects the cells in the row that has the focus, and then moves the contents of the selected rows into the connected row. Word Pro leaves the rows which did not have the focus as disconnected rows of empty cells.

Equivalent to choosing Columns - Connect Across Row.

Syntax

[objectreference].PColConnectRows()

Parameters

None.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Word Pro: PColDisconnectCells method

{button ,AL(^H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL(^H_PCOLDISCONNECTCELLS_METHOD_EXSCRIPT',1)} [See example](#)

Disconnects the selected parallel column table cells. Only works if there are connected cells or rows within the selection. Equivalent to choosing Columns -Disconnect Column Block.

Syntax

[objectreference].PColDisconnectCells()

Parameters

None.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

If you disconnect parallel column cells, the contents of the connected cell remain in a single cell rather than returning to their original separate cells.

Word Pro: PColSelectColumn method

{button ,AL('H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL('H_PCOLSELECTCOLUMN_METHOD_EXSCRIPT',1)} [See example](#)

Selects the parallel column which contains the cell that currently has the focus. If cells from more than one parallel column are in the selection, Word Pro selects all the parallel columns with cells in the selection. The insertion point must be in a parallel column table. Equivalent to choosing Columns - Select - Column Contents.

Syntax

[objectreference].PColSelectColumn()

Parameters

None.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Word Pro: PColSelectRow method

{button ,AL('H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL('H_PCOLSELECTROW_METHOD_EXSCRIPT',1)} [See example](#)

Selects the parallel column row which contains the cell that currently has the focus. If cells from more than one row are in the selection, Word Pro selects all the rows with cells in the selection. The insertion point must be in a parallel column table. Equivalent to choosing Columns - Select - Row Contents.

Syntax

[objectreference].PColSelectRow()

Parameters

None.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Word Pro: PColSelectTable method

{button ,AL(^H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL(^H_PCOLSELECTTABLE_METHOD_EXSCRIPT',1)} [See example](#)

Selects the parallel column table which currently has the focus. The focus must be include a parallel column table cell. Equivalent to choosing Table - Select - All Column Contents.

Syntax

[objectreference].PColSelectTable()

Parameters

None.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Word Pro: Play method

{button ,AL(^H_APPLICATIONWINDOW_CLASS',0)} [See list of classes](#)

Determines what script to play and when to play it..

Syntax

[objectreference].Play(p1)

Parameters

p1

Data type is Variant. Optional parameter. This parameter has two parts:

1. String - Indicates the script filename (.LSS or .LWP).
2. Enumerated list - Indicates when the script will play.

<u>Value</u>	<u>Effect</u>
\$LwpPlayWhenDosIsClosed (1633)	Plays when DOS is closed.
\$LwpPlayWhenDosIsOpened (1632)	Plays when DOS is open.
\$LwpPlayWhenNewdoc (1634)	Plays when you create a new document.
\$LwpPlayWhenWordproIsExecuted (1631)	Plays when you open Word Pro.
\$LwpPlayWhenWordproIsExited (1635)	Plays when you exit Word Pro.

If this method contains no parameters, the current script will play when the method is called.

Return value

Long integer.

Usage

Use this method to play a script.

Word Pro: Present method

{button ,AL('H_TEXTDOCUMENT_CLASS',0)} [See list of classes](#)

Syntax

[objectreference].Present()

Parameters

Return value

Usage

Word Pro: PreviousVersion method

{button ,AL(^H_VERSIONMANAGER_CLASS',0)} [See list of classes](#)

Syntax

[objectreference].PreviousVersion()

Parameters

Return value

Usage

Word Pro: Previous method

{button ,AL('H_CLICKHERE_CLASS;H_TEXTMARKER_CLASS;H_TEXT_CLASS',0)} [See list of classes](#)

Selects the previous specified object type in a Text , TextMarker, or ClickHere object.

Syntax

[objectreference.]Previous(ObjectType, [SearchObjectType,] [p3,] [MarkerName,] [ClassName,] [SubClass,])

Parameters

ObjectType

Specifies the type of object you want to find. Data type is Variant, which allows the value of this parameter to be one of the constants listed below or its numeric equivalent (in parentheses).

<u>Value</u>	<u>Effect</u>
\$LwpNextObjectTypeRevision (1527)	Selects the previous revision. If you use this value, you must also use 18 for the value of the SearchObjectType parameter. If the insertion point is located within a revision, this method selects that revision. Using this value with the Previous method is similar to choosing the Previous Revision button on the Revision bar.
\$LwpNextObjectTypeSearch (1526)	Selects the previous object of the type specified in the SearchObjectType and ClassName parameters.
\$LwpNextObjectTypeTombstone (1528)	Ignores all other parameters. Searches for TeamConsolidate tags.
\$LwpNextObjectTypeFormatcheck (1529)	Selects the next item flagged by Format Check.

SearchObjectType

Specifies the type of object for which you want to search. When ObjectType is \$LwpNextObjectTypeRevision, the value of this parameter must be 18. Data type is Variant, which allows the value of this parameter to be one of the constants listed below or its numeric equivalent (in parentheses). There is no default value.

<u>Value</u>	<u>Effect</u>
\$LwpSelectObjectTypeObject (1736)	Searches for the previous object of the type specified in the ClassName and SubClass parameters.
\$LwpSelectObjectTypeAlignment (1739)	Searches for the previous instance of a local alignment setting. Local alignment settings are those which differ from the alignment defined in the paragraph style. No values are needed for ClassName or SubClass.
\$LwpSelectObjectTypeIndent (1740)	Searches for the previous instance of a local indent setting. Local indent settings are those which differ from the indent setting defined in the paragraph style. No values are needed for ClassName or SubClass.
\$LwpSelectObjectTypeSpacing (1741)	Searches for the previous instance of a local space setting. Local space settings are those which differ from the space setting defined in the paragraph style. No values are needed for

\$LwpSelectObjectTypeTab (1742)

ClassName or SubClass.

Searches for the previous instance of a local tab ruler setting. Local tab ruler settings are those which differ from the tab ruler setting defined in the paragraph style. No values are needed for ClassName or SubClass.

p3

If the ObjectType parameter is \$LwpNextObjectTypeRevision and you want to restrict the Previous method's operation to a specific range in your document, you can use the p3 parameter to provide the name of the range in which you want to find the previous revision. Always an optional parameter.

If the ObjectType parameter is \$LwpNextObjectTypeSearch and the insertion point is within the type of object you are searching for, you can use p3 to tell Word Pro to skip the object your insertion point is in and go to the previous one. You must use an integer value of 1(True) to skip the current object or 0 (False) to select the current object. You must provide a value for this parameter. Default is 1 (True).

MarkerName

If the ObjectType parameter is \$LwpNextObjectTypeSearch and you want to restrict the Previous method's search to a specific range in your document, you can use the MarkerName parameter to provide the name of the range in which you want to search for the previous object. Always an optional parameter.

ClassName

If the ObjectType parameter is \$LwpNextObjectTypeSearch, you must provide a value for the ClassName parameter to specify the type of object for which you are searching. Data type for ClassName is String and the value must be one of the values listed in the table below.

<u>Value</u>	<u>Effect</u>
PageBreak	Finds and selects the previous hard page break. To find the previous page layout change, use "PageBreak" for ClassName and "Layout" for SubClass.
PowerField	Finds and selects the previous Power Field. Index and TOC entries are types of powerfields. To find only the previous Index or TOC entry, use "PowerField" for ClassName and "Index" or "TOC" for SubClass.
BookMark	Finds and selects the previous bookmark.
FootnoteMark	Finds and selects the previous footnote.
ColumnBreak	Finds and selects the previous column break.
AnchoredFrame	Finds and selects the previous frame.
LineBreak	Finds and selects the previous line break.
HardSpace	Finds and selects the previous non-breaking space (CTRL + [space]).
SoftHyphen	Finds and selects the previous soft hyphen (CTRL + -).
Note	Finds and selects the previous Comment Note.
DDE	Finds and selects the previous DDE marker.
PageNumber	Finds and selects the previous page number.
Table	Finds and selects the previous table.
DocVariable	Finds and selects the previous Document Field. To see the document fields, choose Text - Insert Other - Power/Doc. Field. Specify the Document Field option.

Note These values are case-sensitive.

SubClass

Allows you to search for the previous change in page layout or the previous instance of a TOC or Index field. Data type is String.

<u>Value</u>	<u>Effect</u>
Layout	Page layout changes are seen as a type of page break. To search for the previous change in page layout, specify "PageBreak" as the value for the ClassName parameter and then specify "Layout" as the value for SubClass.
Index	Index entries are a type of Power Field. To search for the previous instance of an Index entry, specify "PowerField" as the value for the ClassName parameter and "Index" as the value for SubClass.
TOC	TOC entries are a type of Power Field. To search for the previous instance of an TOC entry, specify "PowerField" as the value for the ClassName parameter and "TOC" as the value for SubClass.

Note These values are case-sensitive.

Return value

If ObjectType is "\$LwpNextObjectTypeRevision", the return value of this method is always 1.

If ObjectType is "\$LwpNextObjectTypeSearch", this method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Word Pro: PrintOut method

{button ,AL('H_WPAPPLICATION_CLASS;H_DOCUMENT_CLASS;H_TEXTDOCUMENT_CLASS',0)} [See list of classes](#)

{button ,AL('H_PRINTOUT_METHOD_EXSCRIPT',1)} [See example](#)

Prints a Word Pro document. This method is defined in several classes. See Usage for details on how this method behaves in each class.

Note This method is the same as the [Print](#) method defined in WPAplication and TextDocument. We provide this alternative because the word "Print" is a reserved word in OLE automation. This reserved status prevents the Print method from working. If you are developing a script for use in OLE automation, you must use the PrintOut method instead of the Print

Note OLE is not supported under OS/2. method.

Syntax

[objectreference].Print([From,][To,][Copies,][nodialog])

Parameters

From

An Integer specifying the first page to be printed. Optional parameter. Default is taken from the document's print options.

To

An Integer specifying the last page to be printed. Optional parameter. Default is taken from the document's print options.

Copies

An Integer specifying the number of copies to be printed. Optional parameter. Default is taken from the document's print options..

nodialog

Allows you to suppress or display the Print dialog box, which normally appears when you choose File - Print. Data type is Integer. The legal values for this parameter are -1 and 0 but you may use the LotusScript constants True (-1) and False (0). Optional parameter. Default is taken from the document's print options..

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Default is taken from the document's print options.

Word Pro: Print method

{button ,AL(^H_WPAPPLICATION_CLASS;H_DOCUMENT_CLASS;H_TEXTDOCUMENT_CLASS;H_PRINTMANAGER_CLASS,0)} [See list of classes](#)

{button ,AL(^H_PRINT_METHOD_EXSCRIPT,1)} [See example](#)

Prints a Word Pro document. This method is defined in several classes. See Usage for details on how this method behaves in each class.

Note The word "Print" is a reserved word in OLE automation. This reserved status prevents the Print method from working. If you are developing a script for use in OLE automation, you must use the [PrintOut](#) method instead of the Print method.

Note OLE is not supported under OS/2.

Syntax

[objectreference].WPAApplication.Print([From,][To,][Copies,][nodialog])

[objectreference].Document.Print([From,][To,][Copies,][nodialog])

[objectreference].PrintManager.Print(DocName)

Parameters

From

An Integer specifying the first page to be printed. Optional parameter. Default is taken from the document's print options. Used only when using Print from WPAApplication or Document.

To

An Integer specifying the last page to be printed. Optional parameter. Default is taken from the document's print options. Used only when using Print from WPAApplication or Document.

Copies

An Integer specifying the number of copies to be printed. Optional parameter. Default is taken from the document's print options. Used only when using Print from WPAApplication or Document.

nodialog

Allows you to suppress or display the Print dialog box, which normally appears when you choose File - Print. Data type is Integer. The legal values for this parameter are -1 and 0 but you may use the LotusScript constants True (-1) and False (0). Optional parameter. Default is taken from the document's print options. Used only when using Print from WPAApplication or Document.

DocName

A String expression specifying the name of the document you want to print. Use this parameter when using Print from PrintManager.

Return value

When called from WPAApplication and TextDocument objects, this method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed, respectively.

PrintManager returns an Integer.

Usage

When called from WPAApplication or a TextDocument, you can take the default values for the parameters that will print one copy of the entire document without displaying the Print dialog box.

When called from the PrintManager object, the Print method requires only the DocName parameter.

Word Pro: ProcessAccelKey method

{button ,AL(^H_GRAPHIC_CLASS',0)} [See list of classes](#)

Syntax

[objectreference].ProcessAccelKey(WParam)

Parameters

WParam

Data type is Integer.

Return value

Integer

Usage

Word Pro: PromoteOutlineLevel method

{button ,AL(^H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

Promotes the current paragraph to the next higher outline level. Equivalent to choosing Text - Outline - Promote.

Syntax

[objectreference].PromoteOutlineLevel()

Parameters

None.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Word Pro: Promote method

{button ,AL(^H_CLICKHERE_CLASS;H_TEXTMARKER_CLASS;H_TEXT_CLASS',0)} [See list of classes](#)

Promotes a paragraph to the next higher outline level. Equivalent to choosing Text - Outline - Promote.

Syntax

[objectreference].Promote()

Parameters

None

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Assigns the next higher paragraph style in the outline style sequence. Affects the paragraph or paragraphs in the object from which you call the method. You can call this method from a Text, TextMarker or ClickHere object.

Word Pro: Purge method

{button ,AL(^H_DIVISION_CLASS;H_FOUNDRY_CLASS;H_TEXTDOCUMENT_CLASS',0)} [See list of classes](#)

Gets rid of any objects in a Foundry object which were derived from a SmartMaster. Use this when you want to switch from one SmartMaster to another and get rid of the old objects created by the original SmartMaster.

Syntax

[objectreference].Purge()

Parameters**Return value**

The return value for this method will always be -1 or 0. When testing the return value, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Word Pro: QuickAlignFrame method

{button ,AL('H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL('H_QUICKALIGNFRAME_METHOD_EXSCRIPT',1)} [See example](#)

Aligns the currently active frame relative to the center or edges of the container in which the frame is located.

Syntax

[objectreference].QuickAlignFrame(QuickLayoutAlign)

Parameters

QuickLayoutAlign

Specifies how you want to align this frame in relation to the container. Data type is Variant, which allows the value of this parameter to be one of the string values listed below or its numeric equivalent (in parentheses). There is no default value.

<u>Value</u>	<u>Effect</u>
\$LwpQuickLayoutAlignLeft (1657)	Aligns the left edge of the frame with the left edge of the container; does not affect the vertical position of the frame.
\$LwpQuickLayoutAlignRight (1658)	Aligns the right edge of the frame with the right edge of the container; does not affect the vertical position of the frame.
\$LwpQuickLayoutAlignVertcenter (1659)	Aligns the center point of the frame with the vertical center of the container; does not affect the horizontal position of the frame.
\$LwpQuickLayoutAlignHorzcenter (1660)	Aligns the center point of the frame with the horizontal center of the container; does not affect the vertical position of the frame.
\$LwpQuickLayoutAlignTop (1661)	Aligns the top of the frame with the bottom edge of the container; does not affect the horizontal position of the frame.
\$LwpQuickLayoutAlignBottom (1662)	Aligns the bottom of the frame with the bottom edge of the container; does not affect the horizontal position of the frame.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

To center a frame on a page container, call this method twice, once with the value, \$LwpQuickLayoutAlignVertcenter, and a second time with the value, \$LwpQuickLayoutAlignHorzcenter.

Word Pro: QuickAlignTable method

{button ,AL(^H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL(^H_QUICKALIGNTABLE_METHOD_EXSCRIPT',1)} [See example](#)

Aligns the currently active table relative to the center or edges of the container in which the table is located.

Syntax

[objectreference].QuickAlignTable(QuickLayoutAlign)

Parameters

QuickLayoutAlign

Specifies how you want to align this table in relation to the container. Data type is Variant, which allows the value of this parameter to be one of the string values listed below or its numeric equivalent (in parentheses). There is no default value.

<u>Value</u>	<u>Effect</u>
\$LwpQuickLayoutAlignLeft (1657)	Aligns the left edge of the table with the left edge of the container; does not affect the vertical position of the table.
\$LwpQuickLayoutAlignRight (1658)	Aligns the right edge of the table with the right edge of the container; does not affect the vertical position of the table.
\$LwpQuickLayoutAlignVertcenter (1659)	Aligns the center point of the table with the vertical center of the container; does not affect the horizontal position of the table.
\$LwpQuickLayoutAlignHorzcenter (1660)	Aligns the center point of the table with the horizontal center of the container; does not affect the vertical position of the table.
\$LwpQuickLayoutAlignTop (1661)	Aligns the top of the table with the bottom edge of the container; does not affect the horizontal position of the table.
\$LwpQuickLayoutAlignBottom (1662)	Aligns the bottom of the table with the bottom edge of the container; does not affect the horizontal position of the table.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

To center a table on a page container, call this method twice, once with the value, \$LwpQuickLayoutAlignVertcenter, and a second time with the value, \$LwpQuickLayoutAlignHorzcenter.

Word Pro: Quit method

{button ,AL(`H_APPLICATION_CLASS;H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL(`H_QUIT_METHOD_EXSCRIPT',1)} [See example](#)

Closes the Word Pro application. Equivalent to choosing File - Exit.

Syntax

[objectreference].Quit()

Parameters

None.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Use this method to exit the Word Pro application.

If Word Pro is acting as an OLE server when this method is executed, Word Pro disappears from the Windows workspace, but remains active in the background until its services are no longer required.

Note OLE is not supported under OS/2.

Word Pro: Read method

{button ,AL('H_BAG_CLASS',0)} [See list of classes](#)

{button ,AL('H_READ_METHOD_EXSCRIPT',1)} [See example](#)

Syntax

[objectreference].Read(Length)

Parameters

Length

Data type is Long.

Return value

Usage

Word Pro: Redo method

{button ,AL('H_WPAPPLICATION_CLASS;H_TEXTDOCUMENT_CLASS',0)} [See list of classes](#)

{button ,AL('H_REDO_METHOD_EXSCRIPT',1)} [See example](#)

Reverses an action which has been undone using the Undo command. Equivalent to choosing Edit - Undo/Redo Special and using the Redo feature.

Syntax

[Objectreference].WPAApplication.Redo()

[Objectreference].TextDocument.Redo(Count)

Parameters

Count

Use only when calling this method from a TextDocument object. The value of this parameter must be an Integer, which specifies the number of undone actions you want to redo. Word Pro can only redo as many actions as have been undone. If this number is greater than the number of undone actions, Word Pro redoes all undone actions and then stops. Optional parameter. Default is 1.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Use this method any time you want to redo an action listed in the "Edits you can redo" list in the Undo/Redo dialog box.

Word Pro: Refresh method

{button ,AL(^H_WINVIEWPREFS_CLASS',0)} [See list of classes](#)

Causes Word Pro to verify the current view preference settings and refresh the view of your document.

Syntax

[objectreference].Refresh()

Parameters

None.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Word Pro: RegisterWPDataSet method

```
{button ,AL('H_WPAPPLICATION_CLASS;H_CHARACTERSTYLE_CLASS;H_DIVISION_CLASS;H_LAYOUT_CLASSES;H_CELLGROUPLAYOUT_CLASS;H_CELLLAYOUT_CLASS;H_CONNECTEDLAYOUT_CLASS;H_COLUMNGROUPPLAYOUT_CLASS;H_FOOTERLAYOUT_CLASS;H_FRAMELAYOUT_CLASS;H_FRAMEGROUPLAYOUT_CLASS;H_DROPCAPLAYOUT_CLASS;H_GROUPLAYOUT_CLASS;H_HEADERLAYOUT_CLASS;H_NOTELAYOUT_CLASS;H_PAGELAYOUT_CLASS;H_ROWGROUPLAYOUT_CLASS;H_ROWLAYOUT_CLASS;H_RUBYLAYOUT_CLASS;H_SUPERTABLEGROUPLAYOUT_CLASS;H_SUPERTABLELAYOUT_CLASS;H_TABLELAYOUT_CLASS;H_ENDNOTELAYOUT_CLASS;H_FOOTNOTELAYOUT_CLASS;H_TABLEHEADINGLAYOUT_CLASS;H_TOCSUPERTABLELAYOUT_CLASS;H_MARKER_CLASS;H_POWERFIELD_CLASS;H_CLICKHERE_CLASS;H_TEXTMARKER_CLASS;H_RUBYMARKER_CLASS;H_TABLEMARKER_CLASS;H_PARAGRAPHSTYLE_CLASS;H_TEXT_CLASS;H_TEXTDOCUMENT_CLASS',0)} See list of classes
```

```
{button ,AL('H_REGISTERWPDATASET_METHOD_EXSCRIPT',1)} See example
```

Creates a WPDataSet and attaches it to the object from which you call this method.

Syntax

```
[objectreference].RegisterWPDataset(GroupName)
```

Parameters

GroupName

A String expression representing the name given to the WPDataSet.

Return value

Returns the newly created WPDataSet object. Assign this return value to a variable to gain easy access to the WPDataSet object.

Usage

Creates and gives a name to a WPDataSet object. The new data set becomes attached to the object from which you call this method. This method returns the new object so you can assign it to a variable.

WPDataSet objects are useful tools that store data with an object. When you close a document that has one or more data sets attached to it, Word Pro saves the data set(s) with the document. Any time the document is open, you have access to the data sets created for that document.

When you register a WPDataSet on a Text, TextMarker, or ClickHere object, the WPDataSet is assigned to the currently active object. If more than one object is selected, the WPDataSet is assigned to the object on which the caret is located.

When you register a WPDataSet on a WPApplication object, the WPDataSet is available in memory for the current session of Word Pro.

Word Pro: Release method

{button ,AL('H_FOUNDRY_CLASS',0)} [See list of classes](#)

{button ,AL('H_RELEASE_METHOD_EXSCRIPT',1)} [See example](#)

Gets rid of a SilverBullet or Style object in the Foundry object from which you call the Release method.

Syntax

[objectreference].Release(FoundryReleaseType, p2, [ObjectName])

Parameters

Type

Indicates whether you want to get rid of a SilverBullet or an object created from one of the following classes: CellLayout, CharacterStyle, FrameLayout, PageLayout, ParagraphStyle, TableLayout. Data type is Variant. The value of this parameter must be one of the strings below or its numeric equivalent (indicated in parentheses):

\$LwpFoundryReleaseTypeSilverbullet (344)

\$LwpFoundryReleaseTypeStyle (343)

p2

The value of this parameter depends on the value of the Type parameter. Data type is Variant. If the value of the Type parameter is \$LwpFoundryReleaseTypeSilverBullet (or 344), this parameter must be a string indicating the name of the SilverBullet object you want to release. If the value of the Type parameter is \$LwpFoundryReleaseTypeStyle (or 343), this parameter must be one of the following integers:

<u>Value</u>	<u>Style Object Deleted</u>
7	Page
8	Frame
23	Table
25	Cell
35	Paragraph
39	Character
107	Header
108	Footer
676	DropCap

ObjectName

If the value of the Type parameter is \$LwpFoundryReleaseTypeStyle (or 343), this parameter must be a string indicating the name of the object you want to release. Data type is String.

Return value

The return value for this method will always be -1 or 0. When testing the return value, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Word Pro: RemoveBookmark method

{button ,AL('H_BOOKMARKMANAGER_CLASS',0)} [See list of classes](#)

{button ,AL('H_REMOVEBOOKMARK_METHOD_EXSCRIPT',1)} [See example](#)

Removes a bookmark from a BookmarkManager object. You must use the internal (hexadecimal) name of the bookmark in order to remove it.

Syntax

[objectreference].RemoveBookmark(MarkerName)

Parameters

MarkerName

A String parameter which represents the internal (hexadecimal) name of the Bookmark object to be removed.

Return value

The return value for this method will always be -1 or 0. When testing the return value, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

To obtain an internal (hexadecimal) name, you can enumerate through the BookmarksByMarkerName collection, or you can select a bookmark by its user-defined name and retrieve its internal name from the MarkerName property.

Word Pro: RemoveChildFromLayout method

```
{button ,AL(^H_LAYOUT_CLASS;H_CELLGROUPLAYOUT_CLASS;H_CELLLAYOUT_CLASS;H_CONNECTEDLAY  
OUT_CLASS;H_COLUMNGROUPLAYOUT_CLASS;H_FOOTERLAYOUT_CLASS;H_FRAMELAYOUT_CLASS;H  
_FRAMEGROUPLAYOUT_CLASS;H_DROP_CAPLAYOUT_CLASS;H_GROUPLAYOUT_CLASS;H_HEADERLAY  
OUT_CLASS;H_NOTELAYOUT_CLASS;H_PAGELAYOUT_CLASS;H_ROWGROUPLAYOUT_CLASS;H_ROWLA  
YOUT_CLASS;H_RUBYLAYOUT_CLASS;H_SUPERTABLEGROUPLAYOUT_CLASS;H_SUPERTABLELAYOUT  
_CLASS;H_TABLELAYOUT_CLASS;H_ENDNOTELAYOUT_CLASS;H_FOOTNOTELAYOUT_CLASS;H_TABLE  
HEADINGLAYOUT_CLASS;H_TOCSUPERTABLELAYOUT_CLASS',0)} See list of classes
```

Removes a specific layout object from its parent layout.

Syntax

```
[objectreference].RemoveChildFromLayout()
```

Parameters

Return value

The return value for this method will always be -1 or 0. When testing the return value, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Call this method from the child layout that you want to remove from its parent.

Word Pro: RemoveDataFile method

{button ,AL(`H_MERGEOPTIONS_CLASS',0)} [See list of classes](#)

{button ,AL(`H_REMOVEDATAFILE_METHOD_EXSCRIPT',1)} [See example](#)

Removes a Merge data file from the current document.

Syntax

[objectreference].RemoveDataFile()

Parameters

None

Return value

Long.

Usage

Word Pro: RemoveDdeLink method

{button ,AL(^H_DDELINKMANAGER_CLASS',0)} [See list of classes](#)

Removes a DDE link from a document.

Syntax

[objectreference].RemoveDdeLink(LinkInfo, [DdeNameType])

Parameters

LinkInfo

Data type is String. Required parameter indicating the name of the DDE link.

DdeNameType

Data type is Variant. Optional parameter. The value of this parameter must be one of the strings below or its code equivalent.

<u>Value</u>	<u>Effect</u>
\$LwpDdeNameTypeLinkinfo (176)	Default. Indicates that the name provided in the LinkInfo parameter is the DDE link info name.
\$LwpDdeNameTypeMarkername (175)	Indicates that the name provided in the LinkInfo parameter is the marker name of the DDE link.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Word Pro: RemoveDepOnDocFile method

{button ,AL(^H_WPAPPLICATION_CLASS;H_TEXTDOCUMENT_CLASS',0)} [See list of classes](#)

Syntax

[objectreference].RemoveDepOnDocFile()

Parameters

Return value

Usage

Word Pro: RemoveDivision method

{button ,AL('H_DIVISION_CLASS;H_TEXTDOCUMENT_CLASS',0)} [See list of classes](#)

{button ,AL('H_REMOVEDIVISION_METHOD_EXSCRIPT',1)} [See example](#)

Removes a division in a text document.

Syntax

[objectreference].RemoveDivision(Name)

Parameters

Name

Data type is String.

Return value

Usage

Word Pro: RemoveEditor method

{button ,AL('H_EDITORMANAGER_CLASS',0)} [See list of classes](#)

{button ,AL('H_REMOVEEDITOR_METHOD_EXSCRIPT',1)} [See example](#)

Removes an editor's name from a document.

Syntax

[objectreference].RemoveEditor(EditorName)

Parameters

EditorName

The name of the editor you want to delete. Data type is String.

Return value

The return value for this method is always -1 or 0. When testing the return value, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

This method returns True if Word Pro removes the editor's name from the document. It returns False under the following conditions: the editor field is blank, an editor does not exist, the editor is the "All Others" editor, or the editor owns any edits in the document.

Usage

Do not use this method to delete "All Others," "Current," or "SmartMaster" editors.

Word Pro: RemoveNamedProperty method

```
{button ,AL(^H_CHARACTERSTYLE_CLASS;H_DIVISION_CLASS;H_LAYOUT_CLASS;H_CELLGROUPLAYOUT_CLASS;H_CELLLAYOUT_CLASS;H_CONNECTEDLAYOUT_CLASS;H_COLUMNGROUPLAYOUT_CLASS;H_FOOTERLAYOUT_CLASS;H_FRAMELAYOUT_CLASS;H_FRAMEGROUPLAYOUT_CLASS;H_DROPCAPLAYOUT_CLASS;H_GROUPLAYOUT_CLASS;H_HEADERLAYOUT_CLASS;H_NOTELAYOUT_CLASS;H_PAGELAYOUT_CLASS;H_ROWGROUPLAYOUT_CLASS;H_ROWLAYOUT_CLASS;H_RUBYLAYOUT_CLASS;H_SUPERTABLEGROUPLAYOUT_CLASS;H_SUPERTABLELAYOUT_CLASS;H_TABLELAYOUT_CLASS;H_ENDNOTELAYOUT_CLASS;H_FOOTNOTELAYOUT_CLASS;H_TABLEHEADINGLAYOUT_CLASS;H_TOCSUPERTABLELAYOUT_CLASS;H_MARKER_CLASS;H_POWERFIELD_CLASS;H_CLICKHERE_CLASS;H_TEXTMARKER_CLASS;H_RBYMARKER_CLASS;H_TABLEMARKER_CLASS;H_TEXTDOCUMENT_CLASS',0)} See list of classes
```

```
{button ,AL(^H_REMOVENAMEDPROPERTY_METHOD_EXSCRIPT',1)} See example
```

Removes the specified user-defined property from a division, layout, marker, or text document. Only works on a WPDataSet property.

Syntax

```
[objectreference].RemoveNamedProperty(PropertyName)
```

Parameters

PropertyName

A String expression representing the name of the property you want to delete in a layout object.

Return value

The return value for this method will always be -1.

Usage

A named property is a user-defined property assigned to an object. Unlike variables, named properties are persistent. They continue to exist when a script stops executing, and when a document is closed and reopened.

Since the RemoveNamedProperty method always returns -1, you must use the HasNamedProperty method to determine whether or not a specified named property actually exists. You do not receive a run-time error if you attempt to remove a named property that does not exist.

Word Pro: RemovePersistentAccelerators method
{button ,AL(^H_ACCELERATORS_CLASS',0)} [See list of classes](#)

Syntax

[objectreference].RemovePersistentAccelerators()

Parameters

Return value

Usage

Word Pro: Remove method

{button ,AL('H_CLICKHERE_CLASS;H_TEXTMARKER_CLASS;H_TEXT_CLASS',0)} [See list of classes](#)

Performs one of the following actions on the current paragraph:

- Removes the local collapse/expand outline settings for all the paragraphs in the text stream.
- Removes the Team Consolidate tag for the current paragraph.

Syntax

[objectreference].Remove(RemoveType, [OthersInSet,] [DeleteParaData])

Parameters

RemoveType

Specifies the type of removal action to perform. Data type is Variant, which allows the value of this parameter to be one of the constants listed below or its numeric equivalent (in parentheses). There is no default value.

<u>Value</u>	<u>Effect</u>
\$LwpRemoveTypeHideshowprops (1672)	Removes the local collapse/expand outline settings for all paragraphs in the text stream. No values required for OthersInSet or DeleteParaData.
\$LwpRemoveTypeMultiParaTag (1673)	Removes the Team Consolidate tag for the current paragraph. Uses default value of False for both OthersInSet and DeleteParaData. If you change the values of OthersInSet or DeleteParaData, you can remove the tags from the other paragraphs in the set, and/or remove the text of the paragraphs along with the tags.

OthersInSet

Allows you to invert the effect of the method by removing the tags from the other paragraphs in the same set as the current paragraph. For example, if you consolidate a document in which the same paragraph contains edits from three people, Word Pro displays three versions of that paragraph. The three versions are seen as a single set. When you use True (-1) as the value for the OthersInSet parameter, Word Pro removes the TeamConsolidate tags from all the paragraphs in the set, except the paragraph from which you call the method. Data type is Integer. The legal values for this parameter are -1 and 0 but you may use the LotusScript constants True (-1) and False (0) instead of the integer values. Optional parameter. Default is False.

DeleteParaData

Specifies whether or not you want to delete the paragraph text, as well as the TeamConsolidate tag(s). Data type is Integer. The legal values for this parameter are -1 and 0 but you may use the LotusScript constants True (-1) and False (0) instead of the integer values. Optional parameter. Default is False.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Word Pro: RenderClipBitmap method

{button ,AL(^H_DOCWINDOW_CLASS',0)} [See list of classes](#)

Syntax

[objectreference].RenderClipBitmap()

Parameters

Return value

Usage

Word Pro: RenderClipDIB method

{button ,AL(^H_DOCWINDOW_CLASS',0)} [See list of classes](#)

Syntax

[objectreference].RenderClipDIB()

Parameters

Return value

Usage

Word Pro: RenderClipMetafile method

{button ,AL(^H_DOCWINDOW_CLASS',0)} [See list of classes](#)

Syntax

[objectreference].RenderClipMetafile()

Parameters

Return value

Usage

Word Pro: RenderClipPalette method

{button ,AL(^H_DOCWINDOW_CLASS',0)} [See list of classes](#)

Syntax

[objectreference].RenderClipPalette()

Parameters

Return value

Usage

Word Pro: Render method

{button ,AL(^H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

Syntax

[objectreference].Render(Format)

Parameters

Format)

Data type is String.

Return value

Usage

Word Pro: ReplaceAll method

{button ,AL(^H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL(^H_REPLACEALL_METHOD_EXSCRIPT',1)} [See example](#)

Executes a replace all, based on the settings found in the FindAndReplace object.

Syntax

[objectreference].ReplaceAll()

Parameters

None.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Word Pro: ReplaceCmd method

{button ,AL('H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL('H_REPLACECMD_METHOD_EXSCRIPT',1)} [See example](#)

Executes a replace, based on the settings found in the FindAndReplace object.

Syntax

[objectreference].ReplaceCmd()

Parameters

None.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Word Pro: Replace method

```
{button ,AL('H_WPAPPLICATION_CLASS;H_CLICKHERE_CLASS;H_TEXTMARKER_CLASS;H_TEXT_CLASS',0)}
```

[See list of classes](#)

```
{button ,AL('H_REPLACE_METHOD_EXSCRIPT',1)} See example
```

Replaces the current text of the specified type with the text specified in the ReplaceText parameter.

Syntax

```
[objectreference].Replace(ReplaceObjectType, ReplaceText)
```

Parameters

ReplaceObjectType

Specifies the type of object. Data type is Variant. The value of this parameter must be one of the strings below or its code equivalent.

<u>Value</u>	<u>Effect</u>
\$LwpReplaceObjectTypeCharacter (1674)	Replaces the current character with Replace text. You must have a character selected for this to work. If more than one character is selected, the entire selection is replaced with the Replace text.
\$LwpReplaceObjectTypeWord (1675)	Replaces the current word with Replacetext.
\$LwpReplaceObjectTypeChunk (1676)	Replaces the current chunk with Replacetext. A chunk is comprised of a single word (a group of characters with no spaces) and all the contiguous spaces following that word. If the insertion point is at the beginning, the end, or anywhere within a word, the chunk is comprised of that word and the spaces which follow it. If the insertion point is between two spaces, the chunk is seen as all the spaces following the insertion point to the beginning of the next word. If there is no word between the spaces and the end of the paragraph, the chunk is comprised of all the spaces up to the end of the paragraph.
\$LwpReplaceObjectTypeSentence (1677)	Replaces the current sentence with Replace text.

ReplaceText

Data type is String. On a ClickHere object, this method replaces the existing Content object with the content object (usually a Text object) named in the ReplaceText parameter.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

If you select text across more than one word, chunk, or sentence and you specify a word, chunk or sentence in ReplaceObjectType, Word Pro replaces the word, chunk, or sentence located at the caret. The caret is always located at the end of the selection. Therefore, if you select from top to bottom, the caret is at the bottom of the selection. If you select from bottom to top, the caret is at the top of the selection.

Word Pro: RequestAndProcessData method

{button ,AL(^H_DDELINKMANAGER_CLASS',0)} [See list of classes](#)

Updates a Dde link in Word Pro.

Syntax

[objectreference].RequestAndProcessData(LinkInfo, [DdeSelection])

Parameters

LinkInfo.

Data type is String. Required parameter indicating the name of the DDE link.

DdeSelection

Data type is Variant. Optional parameter. The value of this parameter must be one of the strings below or its code equivalent.

<u>Value</u>	<u>Effect</u>
\$LwpDdeSelectionDdelinkName (177)	Indicates that the name provided in the LinkInfo parameter is the DDE link info name.
\$LwpDdeSelectionMarkerName (178)	Default value. Indicates that the name provided in the LinkInfo parameter is the marker name of the DDE link.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Word Pro: ResetFindAndReplace method

{button ,AL(^H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL(^H_RESETFINDANDREPLACE_METHOD_EXSCRIPT',1)} [See example](#)

Resets the Find & Replace marks from the insertion point forward and allows you to continue the Find & Replace operation.

Syntax

[objectreference].ResetFindAndReplace()

Parameters

None.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

If you begin a Find & Replace operation and interrupt the operation to make changes inside your document, Word Pro dismisses the Find & Replace marks and allows you to make the changes. Use this method to reset the Find & Replace marks and continue the operation. The operation begins at the insertion point and continues to the end of the document, then loops back to the beginning and continues to the location of the insertion point.

Word Pro: Reset method

{button ,AL(`H_BAG_CLASS;H_NUMERICFORMAT_CLASS;H_FINDANDREPLACE_CLASS',0)} [See list of classes](#)

{button ,AL(`H_RESET_METHOD_EXSCRIPT',1)} [See example](#)

[FindAndReplace]

Sets the options back to the default setting (clear) for Find & Replace text.

[NumericFormat]

Removes conditional number format attributes.

Syntax

[objectreference].Reset()

Parameters

Return value

The return value for this method will always be -1 or 0. When testing the return value, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values. Default is True.

Usage

[FindAndReplace]

Equivalent to choosing Edit - Find & Replace Text, clicking Options and the Font button in either the "Find options" or "Replace options" section. If you choose a font name, attributes, style, or color, and then want to start again, click Clear to reset.

[NumericFormat]

This method removes conditional number format attributes, such as leading text for negative numbers. Equivalent to clicking the Reset button in the Edit Format dialog box. You can access this dialog box by clicking the Format Options button on the Number Format panel of the InfoBox.

Word Pro: Resize method

{button ,AL(^H_WINDOW_CLASS;H_APPLICATIONWINDOW_CLASS;H_STATUSBAR_CLASS;H_DOCWINDOW_CLASS',0)} [See list of classes](#)

{button ,AL(^H_RESIZE_METHOD_EXSCRIPT',1)} [See example](#)

[ApplicationWindow]

Resizes the application window at the current position to the width and height specified by the parameters.

Syntax

[objectreference].Resize(Width, Height)

Parameters*Width*

Data type is Long but the unit of measurement used for this property is Twips. There are 1440 Twips per inch.

Height

Data type is Long but the unit of measurement used for this property is Twips. There are 1440 Twips per inch.

Return value

The return value for this method will always be -1 or 0. When testing the return value, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

This method is not valid for StatusBar.

[ApplicationWindow]

Use this method to change the size of the application window.

Word Pro: Restore method

{button ,AL('H_APPLICATIONWINDOW_CLASS;H_DOCWINDOW_CLASS',0)} [See list of classes](#)

{button ,AL('H_RESTORE_METHOD_EXSCRIPT',1)} [See example](#)

Restores the Word Pro application or document window to its previous size.

Syntax

[objectreference].Restore()

Parameters

None.

Return value

The return value for this method will always be -1 or 0. When testing the return value, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Equivalent to the Restore button in the upper right-hand corner of the window from which this method is called.

Word Pro: RetrieveInternetFileAndOpen method

{button ,AL('H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL('H_RETRIEVEINTERNETFILEANDOPEN_METHOD_EXSCRIPT',1)} [See example](#)

Gets a file via WWW or FTP, stores it in your temporary directory, and opens the file for viewing in Word Pro. Equivalent to choosing File - Open and clicking Internet. The parameters for this method correspond to the options available in the Open from Internet dialog box.

Syntax

[objectreference].RetrieveInternetFileAndOpen(URL, UserID, Password, Passive, Proxy, ProxyPort)

Parameters

URL

A String expression specifying the URL for the document you want to open.

UserID

Used when retrieving a file from an FTP site, this parameter takes a String value representing the name of the user who has an account with the FTP server. If you are retrieving a WWW document, the value of this parameter should be a null string ("").

Password

Used when retrieving a file from an FTP site, this optional parameter takes a String value representing the password for the user named in UserID. If you are retrieving a WWW document, the value of this parameter should be a null string ("").

Passive

Set this optional parameter to True when you want to initiate the file transfer. Set it to False to allow the server to respond to your request when it is ready. Some FTP servers do not support this feature. The value of this parameter is usually False. If you are retrieving a WWW document, the value of this parameter should be a null string (""). Data type is Integer, but the value is always 0 (False) or -1 (True). You can use the LotusScript constants of True and False.

Proxy

An optional String expression specifying the DNS (for example, screen.companyname.com) or IP address (for example, 123.456.78.912) of the Proxy server. Do not include the "http:\\\" in front of the the proxy value.

ProxyPort

An optional Integer which specifies the port number for the Proxy server. The value of this parameter is usually 8080 for the WWW and 21 for FTP, but you should check with your Internet Service Provider for your settings.

Return value

A String representing the name of the file you retrieved and opened.

Usage

You must install the Word Pro HTML filter to access most WWW documents with this method. This method does not work unless your machine is configured for Internet access. A standard Internet access configuration includes a WINSOCK compliant DLL.

The retrieved file is stored in the directory you marked as your temporary directory. The temporary directory is identified in an environment variable, but most temporary directories are named "TEMP."

Word Pro: RetrievalInternetFile method

{button ,AL('H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL('H_RETRIEVEINTERNETFILE_METHOD_EXSCRIPT',1)} [See example](#)

Gets a file via WWW or FTP and stores it in your temporary directory. Equivalent to choosing File - Open and clicking Internet. The parameters for this method correspond to the options available in the Open from Internet dialog box.

Syntax

[objectreference].RetrievalInternetFile(URL, [UserID], [Password], [Passive], [Proxy], [ProxyPort])

Parameters

URL

A String expression specifying the URL for the document you want to open.

UserID

Used when retrieving a file from an FTP site, this parameter takes a String value representing the name of the user who has an account with the FTP server. If you are retrieving a WWW document, the value of this parameter should be a null string ("").

Password

Used when retrieving a file from an FTP site, this optional parameter takes a String value representing the password for the user named in UserID. If you are retrieving a WWW document, the value of this parameter should be a null string ("").

Passive

Set this optional parameter to True when you want to initiate the file transfer. Set it to False to allow the server to respond to your request when it is ready. Some FTP servers do not support this feature. The value of this parameter is usually False. If you are retrieving a WWW document, the value of this parameter should be a null string (""). Data type is Integer, but the value is always 0 (False) or -1 (True). You can use the LotusScript constants of True and False.

Proxy

An optional String expression specifying the DNS (for example, screen.companyname.com) or IP address (for example, 123.456.78.912) of the Proxy server. Do not include the "http:\\\" in front of the the proxy value.

ProxyPort

An optional Integer parameter which specifies the port number for the Proxy server. The value of this parameter is usually 8080 for the WWW and 21 for FTP, but you should check with your Internet Service Provider for your settings.

Return value

A String representing the name of the file you retrieved.

Usage

This method does not work unless your machine is configured for Internet access. A standard Internet access configuration includes a WINSOCK compliant DLL.

The retrieved file is stored in the directory you marked as your temporary directory. The temporary directory is identified in an environment variable, but most temporary directories are named "TEMP."

Word Pro: RevertToSaved method

{button ,AL('H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL('H_REVERTTOSAVED_METHOD_EXSCRIPT',1)} [See example](#)

Cancels changes made to the document since it was last saved, and displays the previously saved version of the document. Equivalent to closing a file without saving the changes and then reopening the file.

Syntax

[objectreference].RevertToSaved()

Parameters

None.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Word Pro: RevertToStyle method

```
{button ,AL('H_ALIGNMENT_CLASS;H_WPAPPLICATION_CLASS;H_AMIKAKE_CLASS;H_ATTRIBUTES_CLASS;H_COLOR_CLASS;H_BREAKS_CLASS;H_BULLET_CLASS;H_CELLCONTAINER_CLASS;H_CHARACTERBORDER_CLASS;H_FONT_CLASS;H_FRAMECONTAINER_CLASS;H_NOTECONTAINER_CLASS;H_RUBYCONTAINER_CLASS;H_DROPDOWNCONTAINER_CLASS;H_INDENT_CLASS;H_RELATIVEINDENT_CLASS;H_KINSOKU_CLASS;H_LANGUAGE_CLASS;H_PARAGRAPHBORDER_CLASS;H_CLICKHERE_CLASS;H_TEXTMARKER_CLASS;H_NUMBERING_CLASS;H_PAGECONTAINER_CLASS;H_SUBPAGECONTAINER_CLASS;H_SUPERPAGECONTAINER_CLASS;H_PARAGRAPHBACKGROUND_CLASS;H_SPACING_CLASS;H_SUPERTABLECONTAINER_CLASS;H_TABRACK_CLASS;H_TEXT_CLASS',0)} See list of classes
```

```
{button ,AL('H_REVERTTOSTYLE_METHOD_EXSCRIPT',1)} See example
```

Reverts an object to a style contained within the document's SmartMaster.

Syntax

When called from a Color or a Language object: [Objectreference].RevertToStyle()

When called from a ParagraphBackground object: [Objectreference].RevertToStyle(Integer)

When called from any other object: [Objectreference].RevertToStyle(RevertType)

Parameters

RevertType

Indicates the type of object that is being reverted to the original style. Data type of Variant. The legal values for this parameter vary according to the class from which this method is called. The legal values for each type of object are listed under Usage.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

You can call this method from a wide variety of objects, but the method acts differently with each type of object. For example, when you call this method from an Alignment object, your only choice is to remove local alignment settings. However, if you call this method from an Attributes object, it can remove any of nine different types of attributes; you can use the RevertType parameter to choose which type of attribute to remove.

Listed below are the objects that contain the RevertToStyle method, along with the possible values for the RevertType parameter, and what those values do when this method is called from the specific object.

When called from the WPAApplication object

Affects the object which has the focus at the time the method is called. The method will accept any value but will only work correctly if the value matches one accepted by an object in the focus. For example, when the insertion point is in a table cell, there are several objects in the focus which can use the RevertToStyle method. You specify the desired object by passing its corresponding RevertType value.

A value of \$LwpStyleTypeParagraph or \$LwpStyleTypeCharacter would affect the Text object in the table cell. \$LwpStyleTypeCell would affect the Cell object. \$LwpStyleTypeTable would affect the Table object. All of these objects are within the focus when the insertion point is located in a table cell. You specify which object should be reverted by using a RevertType value for the desired object.

When called from an Alignment object:

<u>RevertType Value</u>	<u>Effect</u>
\$LwpAlignStyleType (2095)	Removes any local alignment settings.

When called from an Attributes object:

<u>RevertType Value</u>	<u>Effect</u>
\$LwpAttributePropAll (19)	Removes all local attribute settings.
\$LwpAttributePropBaseline (2200)	Removes any local baseline attribute setting.
\$LwpAttributePropHidden (20)	Removes any local hidden attribute setting.

\$LwpAttributePropHighlight (23)	Removes any local highlight attribute setting.
\$LwpAttributePropHyphen (21)	Removes any local hyphenation attribute setting.
\$LwpAttributePropLevel (26)	Removes any local level attribute setting.
\$LwpAttributePropMisspelled (24)	Removes any local misspelling attribute settings
\$LwpAttributePropProtected (22)	Removes any local protected attribute setting.
\$LwpAttributePropSkipped (25)	Removes any local skipped attribute setting.

When called from a Breaks object:

RevertType Value	Effect
\$LwpBreakPropAllBreaks (59)	Removes all local break settings.
\$LwpBreakPropColumnBreakAfter (64)	Removes any local column break after settings.
\$LwpBreakPropColumnBreakBefor (63)	Removes any local column break before settings.
\$LwpBreakPropKeepWithNext (65)	Removes any local keep-with next setting.
\$LwpBreakPropKeepWithPrev (66)	Removes any local keep-with previous setting.
\$LwpBreakPropNextStyleName (68)	Removes any local paragraph style setting from the "Style to use for next paragraph" box; does not enable or disable the option completely.
\$LwpBreakPropPageBreakAfter (61)	Removes any local break page after settings.
\$LwpBreakPropPageBreakBefore (60)	Removes any local break page before settings.
\$LwpBreakPropPageBreakWithin (62)	Removes any local "Keep entire paragraph on same page" setting.
\$LwpBreakPropUseNextStyle (67)	Removes any local setting from the "Style to use for next paragraph" box. You must also use \$LwpBreakPropNextStyleName in order to remove local paragraph style name settings.

When called from a Bullet object:

RevertType Value	Effect
\$LwpBulletPropertyAll (82)	Removes all local bullet property settings.
\$LwpBulletPropertyBullet (83)	Removes any local "Bullet style" bullet setting.
\$LwpBulletPropertyEditable (86)	Removes any local "Edit on page" bullet setting.
\$LwpBulletPropertyRightalign (85)	Removes any local "Right align" bullet setting.
\$LwpBulletPropertyValidprop (84)	

When called from any of the following objects:

Text, TextMarker, ClickHere

Container objects, including CellContainer, DropCapContainer, FrameContainer, NoteContainer, PageContainer, RubyContainer, SubPageContainer, SuperTableContainer:

RevertType Value	Effect
\$LwpStyleTypeCell (1834)	Reverts a cell to the settings received from its cell style.
\$LwpStyleTypeCharacter (1830)	Removes any local character attribute setting.
\$LwpStyleTypeFrame (1832)	Reverts a frame to the settings received from its frame style.
\$LwpStyleTypePage (1831)	Reverts a page to the settings received from its page style.
\$LwpStyleTypeParagraph (1829)	Reverts a paragraph to the settings received from its paragraph style.

\$LwpStyleTypeTable (1833) Reverts a table to the settings received from its table style.

When called from a CharacterBorder or ParagraphBorder object:

RevertType Value	Effect
\$LwpLineRevertAboveType (565)	Removes any local line above attribute. Since individual line settings may vary, it is recommended that you use \$LwpLineRevertAll in order to remove all local line settings.
\$LwpLineRevertAboveWidth (563)	Removes any local line above width attribute. Since individual line settings may vary, it is recommended that you use \$LwpLineRevertAll in order to remove all local line settings.
\$LwpLineRevertAll (562)	Removes all local line settings.
\$LwpLineRevertBelowType (566)	Removes any local line below attribute. Since individual line settings may vary, it is recommended that you use \$LwpLineRevertAll in order to remove all local line settings.
\$LwpLineRevertBelowWidth (564)	Removes any local line below width attribute. Since individual line settings may vary, it is recommended that you use \$LwpLineRevertAll in order to remove all local line settings.
\$LwpLineRevertRightType (567)	Removes any local right line attributes. Since individual line settings may vary, it is recommended that you use \$LwpLineRevertAll in order to remove all local line settings.

When called from a Color object:

Does not use the RevertType parameter. Always removes the current local color attribute and reverts to the original color attribute set for the associated style.

When called from a Font object:

RevertType Value	Effect
\$LwpFontPropertyAll (297)	
\$LwpFontPropertyAlllower (315)	
\$LwpFontPropertyAllupper (314)	
\$LwpFontPropertyBold (303)	
\$LwpFontPropertyCase (316)	
\$LwpFontPropertyDoubleunderline (308)	
\$LwpFontPropertyItalic (304)	
\$LwpFontPropertyName (298)	
\$LwpFontPropertyNextname (300)	
\$LwpFontPropertyOverstrike (312)	
\$LwpFontPropertySize (301)	
\$LwpFontPropertySmallcaps (313)	
\$LwpFontPropertySpacing (302)	
\$LwpFontPropertyStrikethru (311)	
\$LwpFontPropertySubscript (306)	
\$LwpFontPropertySuperscript (305)	
\$LwpFontPropertyUnderline (307)	

\$LwpFontPropertyWindowsname (299)
\$LwpFontPropertyWorddoubleunderline
(310)
\$LwpFontPropertyWordunderline (309)

When called from an Indent or RelativeIndent object:

RevertType Value	Effect
\$LwpIndentPropertyAll (403)	
\$LwpIndentPropertyBodyonly (409)	
\$LwpIndentPropertyEnabled (410)	
\$LwpIndentPropertyEvery (402)	
\$LwpIndentPropertyFirst (404)	
\$LwpIndentPropertyHang (407)	
\$LwpIndentPropertyRelative (411)	
\$LwpIndentPropertyRest (405)	
\$LwpIndentPropertyRight (406)	
\$LwpIndentPropertySidesequal (408)	

When called from a Language object:

Does not use the RevertType parameter. Always removes any local language attributes.

When called from a Numbering object:

RevertType Value	Effect
\$LwpNumberingPropsRevertAll (1583)	
\$LwpNumberingPropsRevertHeading (1586)	
\$LwpNumberingPropsRevertLevel (1584)	
\$LwpNumberingPropsRevertPosition (1585)	

When called from a ParagraphBackground object:

This method is not in service for ParagraphBackground objects, but may be implemented in the future.

When called from a Spacing object:

RevertType Value	Effect
\$LwpSpacingPropertyAboveamount (1784)	
\$LwpSpacingPropertyAbovenumber (1785)	
\$LwpSpacingPropertyAbovetype (1783)	
\$LwpSpacingPropertyAll (1778)	
\$LwpSpacingPropertyAlways (1779)	
\$LwpSpacingPropertyAmount (1787)	
\$LwpSpacingPropertyBelowamount (1781)	
\$LwpSpacingPropertyBelownumber (1782)	
\$LwpSpacingPropertyBelowtype (1780)	
\$LwpSpacingPropertyNumber (1788)	

When called from a TabRack object:

RevertType Value

Effect

\$LwpTabRackPropertyAll (1859)

```
'Example: OpenDataFile method
' This example opens the merge data file for the current merge document.
' RUNTIME DEPENDENCIES: You must have a document open that has been assigned a
' merge data file for this script to work.

DataFile = .ActiveDocument.MergeOptions.DataFileName
If DataFile <> "" Then
    .OpenDataFile(DataFile,"Lotus Word Pro")
End If
```

```
'Example: OpenDocsVisible method
' This example enumerates all the entries in the glossary named GLOSSARY.GLS.
' RUNTIME DEPENDENCIES: You must have a glossary named GLOSSARY.GLS and a
' document open for this script to work.

.ApplicationWindow.UserInterfacePrefs.OpenDocsVisible = False
.ApplicationWindow.UserInterfacePrefs.OpenReadOnly = True
.GlossaryOpen "GLOSSARY.GLS", "Lotus Word Pro"
.ApplicationWindow.UserInterfacePrefs.OpenDocsVisible = True
.ApplicationWindow.UserInterfacePrefs.OpenReadOnly = False
Forall Gloss In .Division.Foundry.Glossarys
    Count% = Gloss.NumRows
    For Item% = 1 To (Count% - 1)
        Print "Item" & Format$(Item%) & "= " & Gloss.EnumerateTerm(Item%)
    Next
End Forall
.Close
```

```
'Example: OpenDocument method
' This example uses an input box to get a file name from you and then opens
' the document with that file name. The default file name is "README95.LWP".
' RUNTIME DEPENDENCIES: You must have a document open for this script to work.

Dim FileName as String
FileName = InputBox$ ("Enter a filename to open, e.g., README95.LWP:", "Example
Script", "")

.OpenDocument FileName, "", "", "", False, True
```

```
'Example: OpenReadOnly method
' This example enumerates all the entries in the glossary named GLOSSARY.GLS.
' RUNTIME DEPENDENCIES: You must have a glossary named GLOSSARY.GLS and a
' document open for this script to work.

.ApplicationWindow.UserInterfacePrefs.OpenDocsVisible = False
.ApplicationWindow.UserInterfacePrefs.OpenReadOnly = True
.GlossaryOpen "GLOSSARY.GLS", "Lotus Word Pro"
.ApplicationWindow.UserInterfacePrefs.OpenDocsVisible = True
.ApplicationWindow.UserInterfacePrefs.OpenReadOnly = False
Forall Gloss In .Division.Foundry.Glossarys
    Count% = Gloss.NumRows
    For Item% = 1 To (Count% - 1)
        Print "Item" & Format$(Item%) & "= " & Gloss.EnumerateTerm(Item%)
    Next
End Forall
.Close
```


'Example: OrigHeight property
'This example imports a graphic, then changes the frame size, so the
'graphic's size changes as well. The graphic's original height and width,
'along with the new height and width, are reported.

```
GraphicName = .ApplicationWindow.UserInterfacePrefs.GraphicPath & "\_wpdon.gif"  
.ImportGraphic GraphicName, ".gif", False, False, "Default Graphic/OLE"  
.Layout.Width = 4320  
.Layout.Height = 2880  
OHeight = .Graphic.OrigHeight  
OWidth = .Graphic.OrigWidth  
CHeight = .Graphic.Height  
CWidth = .Graphic.Width  
MessageBox "This graphic was originally " & OHeight & " by " & OWidth & _  
    " twips. It is currently " & CHeight & " by " & CWidth & " twips."
```

'Example: OrigWidth property

'This example imports a graphic, then changes the frame size, so the
'graphic's size changes as well. The graphic's original height and width,
'along with the new height and width, are reported.

```
GraphicName = .ApplicationWindow.UserInterfacePrefs.GraphicPath & "\_wpdon.gif"  
.ImportGraphic GraphicName, ".gif", False, False, "Default Graphic/OLE"  
.Layout.Width = 4320  
.Layout.Height = 2880  
OHeight = .Graphic.OrigHeight  
OWidth = .Graphic.OrigWidth  
CHeight = .Graphic.Height  
CWidth = .Graphic.Width  
MessageBox "This graphic was originally " & OHeight & " by " & OWidth & _  
    " twips. It is currently " & CHeight & " by " & CWidth & " twips."
```

```
'Example: OutlineButtons property
'This example asks the user whether to display outline buttons in outline view,
'and then sets the appropriate option.

stat = MessageBox ("Do you want to view outline buttons?", 36, "Example Script")
If stat = 6 Then ' user said yes
    .ActiveDocWindow.WinViewPrefs.OutlineButtons = 1
Else
    .ActiveDocWindow.WinViewPrefs.OutlineButtons = 0
End If
```

```
'Example: OutlineHeadingButtonsOnly property
'This example asks the user whether to display buttons for outline headings
'in Outline view and then sets the appropriate option.

stat = MessageBox ("Do you want to view heading buttons?", 36, "Example Script")
If stat = 6 Then ' user said yes
    .ActiveDocWindow.WinViewPrefs.OutlineHeadingButtonsOnly = 1
Else
    .ActiveDocWindow.WinViewPrefs.OutlineHeadingButtonsOnly = 0
End If
```

'Example: OutlineLevel property

'This example inserts text at differing outline levels, then sets the

'OutlineLevel property to show only level 2 text and higher.

```
.Text.InsertText "This is outline level 1.", True
```

```
.Text.Numbering.Level = 2
```

```
.Text.InsertText "This is outline level 2.", True
```

```
.Text.Numbering.Level = 3
```

```
.Text.InsertText "This is outline level 3.", True
```

```
.Text.Numbering.Level = 4
```

```
.Text.InsertText "This is outline level 4.", True
```

```
.Text.Numbering.Level = 2
```

```
.Text.InsertText "This is outline level 2.", True
```

```
.Page.Presentation.OutlineLevel = 2
```

'Example: OutlineMoveTextDown method

'This example types four paragraphs of text at different outline levels.

'The insertion point is then moved up two lines, and the second paragraph is
'moved below the third.

.Type "This is the first outline level.[Enter]"

.DemoteOutlineLevel

.Type "This is the second level of outline[Enter]"

.DemoteOutlineLevel

.Type "This is the third level of outline[Enter]"

.DemoteOutlineLevel

.Type "This is the fourth level.[Up][Up]"

.OutlineMoveTextDown

'Example: OutlineMoveTextUp method

'This example types four paragraphs of text into the current document.

'The OutlineMoveTextUp Method is then called to move the fourth line
'above the third.

.Type "This is the first level of outline.[Enter]"

.DemoteOutlineLevel

.Type "This is the second level of outline[Enter]"

.DemoteOutlineLevel

.Type "This is the third level of outline[Enter]"

.DemoteOutlineLevel

.Type "This is the fourth level."

.OutlineMoveTextUp

```
'Example: OutlineOnlyHeadingsWhenCollapsed property
'This example asks the user whether to display outline headings in a collapsed outline
view,
'and then sets the appropriate option.

stat = MessageBox ("Do you want to view outline headings only?", 36, "Example Script")
If stat = 6 Then  ' user said yes
    .ActiveDocWindow.WinViewPrefs.OutlineOnlyHeadingsWhenCollapsed = 1
Else
    .ActiveDocWindow.WinViewPrefs.OutlineOnlyHeadingsWhenCollapsed = 0
End If
```


'Example: OutlineSeqItems property

'This example creates an outline style sequence, and adds three items to it.

'First, create the sequence, name it, and make it empty

```
NewSequence = .Division.Foundry.Create($LwpFoundryCreateTypeOutlineseq)
.Division.Foundry.OutlineStyleSequences(NewSequence).Name = "Typical Outline"
.Division.Foundry.OutlineStyleSequences("Typical Outline").Clear
.Division.Foundry.OutlineStyleSequences("Typical Outline").Heading = 1
```

' Now, add a sequence item for the Heading 1 paragraph style

```
.Division.Foundry.OutlineStyleSequences("Typical Outline").AddOutlineSequenceItem
"Heading 1"
```

' And set it's position

```
.Division.Foundry.OutlineStyleSequences("Typical
Outline").OutlineSeqItems("Heading 1").Position = 1
```

'Add Heading 2 and Heading 3 paragraph styles

```
.Division.Foundry.OutlineStyleSequences("Typical Outline").AddOutlineSequenceItem
"Heading 2"
```

```
.Division.Foundry.OutlineStyleSequences("Typical
Outline").OutlineSeqItems("Heading 2").Position = 2
```

```
.Division.Foundry.OutlineStyleSequences("Typical Outline").AddOutlineSequenceItem
"Heading 3"
```

```
.Division.Foundry.OutlineStyleSequences("Typical
Outline").OutlineSeqItems("Heading 3").Position = 3
```

'Example: OutlineStyleSequences property

'This example creates a new outline style sequence and adds an item to it.

'Create the new sequence

```
NewSequence = .Division.Foundry.Create($LwpFoundryCreateTypeOutlineseq)
```

'Give the new sequence a name

```
.Division.Foundry.OutlineStyleSequences(NewSequence).Name = "Typical Outline"
```

'Clear any existing data, and set the Heading type

```
.Division.Foundry.OutlineStyleSequences("Typical Outline").Clear
```

```
.Division.Foundry.OutlineStyleSequences("Typical Outline").Heading = 1
```

'Now add an item to the sequence

```
.Division.Foundry.OutlineStyleSequences("Typical Outline").AddOutlineSequenceItem  
"Heading 1"
```

'And set the new item's position.

```
.Division.Foundry.OutlineStyleSequences("Typical Outline").OutlineSeqItems("Heading  
1").Position = 1
```

'Example: OutputToFile property
'This example asks the user whether to send print output to a file, then sets
'the appropriate option, and prints the document.

```
stat = MessageBox ("Do you want to print to file?", 36, "Example Script")
If stat = 6 Then ' user said yes
    .ActiveDocument.PrintSettings.OutputToFile = True
Else
    .ActiveDocument.PrintSettings.OutputToFile = False
End If
.Print
```

'Example: Override property
'This example sets the text color to an RGB value, and to black.

```
.Text.Font.FontColor.Red = 65  
.Text.Font.FontColor.Blue = 255  
.Text.Font.FontColor.Green = 129  
.Text.Font.FontColor.Override = $LwpColorOverrideRgb  
.Type "This text is light blue."  
.Text.Font.FontColor.Override = $LwpColorOverrideBlack  
.Type "This text is black. "
```

```
'Example: OverstrikeCharacter property  
'This example sets the overstrike character for text to the hyphen.  
'Some text is typed, and then the overstrike attribute is removed.
```

```
Dim HyphChar as Long  
HyphChar = Asc("-")  
.Text.Font.OverstrikeCharacter = HyphChar  
.Type "This text is overstruck."  
.Text.Font.RevertToStyle $LwpFontPropertyOverstrike
```

'Example: Overstrike property

'This example asks the user whether to strike through the text, and then sets the appropriate option.

```
.Type "This is some example text[ShiftHome]"
```

```
stat = MessageBox ("Do you want to strike through the text?", 36, "Example Script")
```

```
If stat = 6 Then ' user said yes
```

```
    .Text.Font.Overstrike = True
```

```
Else
```

```
    .Text.Font.Overstrike = False
```

```
End If
```

```
'Example: PageDown method
' This example simulates pressing the page down and page up keys in the
' current document;
' RUNTIME DEPENDENCIES: You must have a document open for this script to work.

.PageDown
MessageBox "Click OK to pageup.",MB_OK,"Example Script"
.PageUp
```

'Example: PageNumberAsText property
'This example inserts two references to the text on different pages, and
'marks them with bookmarks. At the end of the document, a table of
'references is built, using the PageNumberAsText property to display
'the page number on which the references occur.
'Because page numbering with Roman Numeral style is used in the document,
'that style is reflected in the references.

```
.GoToObject "Header", True ' insert page numbering in the header
.InsertPageNumber $LwpNumberingStyleUppercaseroman, "", ".", 0, 1, &H0
.Header.GoToContainer $LwpGoToLocationExit ' return to the document
.Text.InsertText "This is Reference 1." ' insert first reference
.Text.Backward $LwpNavigateObjectTypeWord, 3
.Text.Shade $LwpLocationTypeWord, $LwpNavigateDirectionRight, 1
Ref1Mark = .Text.Mark ($LwpMarkerTypeBookmark) ' bookmark the reference
.Division.BookmarkManager.AddBookmark "Ref1", Ref1Mark
.Deselect
.Text.Forward $LwpNavigateObjectTypeWord, 1
.InsertPageBreak
.Text.InsertText "This is Reference 2." ' insert second reference
.Text.Backward $LwpNavigateObjectTypeWord, 3
.Text.Shade $LwpLocationTypeWord, $LwpNavigateDirectionRight, 1
Ref2Mark = .Text.Mark ($LwpMarkerTypeBookmark)
.Division.BookmarkManager.AddBookmark "Ref2", Ref2Mark
.Text.Forward $LwpNavigateObjectTypeWord, 1
.Deselect
.InsertPageBreak
.Text.InsertText "References:", True ' build table of references
MarkText = .Division.Foundry.Markers(Ref1Mark).GetMarkedText
.Text.InsertText MarkText
.Text.InsertTab
.Text.InsertText .Division.Foundry.Markers(Ref1Mark).PageNumberAsText, True
MarkText = .Division.Foundry.Markers(Ref2Mark).GetMarkedText
.Text.InsertText MarkText
.Text.InsertTab
.Text.InsertText .Division.Foundry.Markers(Ref2Mark).PageNumberAsText, True
```


'Example: PageNumber property

'This example inserts references in the document on two separate pages.

'These references are marked by bookmarks. At the end of the document, a

'table of references is built, using the PageNumber property of the

'bookmarks to display the page number on which the references appear.

```
.Text.InsertText "This is Reference 1." 'insert the first reference
.Text.Backward $LwpNavigateObjectTypeWord, 3 ' select text for the bookmark
.Text.Shade $LwpLocationTypeWord, $LwpNavigateDirectionRight, 1
Ref1Mark = .Text.Mark ($LwpMarkerTypeBookmark ) ' add the bookmark
.Division.BookmarkManager.AddBookmark "Ref1", Ref1Mark
.Deselect
.Text.Forward $LwpNavigateObjectTypeWord, 1 ' move to the end, and
.InsertPageBreak ' insert page break
.Text.InsertText "This is Reference 2."
.Text.Backward $LwpNavigateObjectTypeWord, 3
.Text.Shade $LwpLocationTypeWord, $LwpNavigateDirectionRight, 1
Ref2Mark = .Text.Mark ($LwpMarkerTypeBookmark)
.Division.BookmarkManager.AddBookmark "Ref2", Ref2Mark
.Text.Forward $LwpNavigateObjectTypeWord, 1
.Deselect
.InsertPageBreak
.Text.InsertText "References:", True ' create our table of reference
MarkText = .Division.Foundry.Markers(Ref1Mark).GetMarkedText
.Text.InsertText MarkText
.Text.InsertTab
PNum = .Division.Foundry.Markers(Ref1Mark).PageNumber
.Text.InsertText Pnum, True
MarkText = .Division.Foundry.Markers(Ref2Mark).GetMarkedText
.Text.InsertText MarkText
.Text.InsertTab
PNum = .Division.Foundry.Markers(Ref2Mark).PageNumber
.Text.InsertText Pnum, True
```

'Example: PageNum property

'This example inserts text on two pages, and reports the page number of
'each page in a message box.

```
.Text.InsertText "This is some sample text", True
```

```
MessageBox "The current page number is " & .Page.PageNum, MB_OK, "Example Script"
```

```
.InsertPageBreak
```

```
.Text.InsertText "This is some sample text", True
```

```
MessageBox "The current page number is " & .Page.PageNum, MB_OK, "Example Script"
```

'Example: PageOrder property

'This example asks the user whether to print the document in reverse order,

'sets the appropriate option, and prints the document.

```
stat = MessageBox ("Do you want to print the document in reverse order?", 36, "Example  
Script")
```

```
If stat = 6 Then ' user said yes
```

```
    .ActiveDocument.PrintSettings.PageOrder = $LtsPageOrderBackToFront
```

```
Else
```

```
    .ActiveDocument.PrintSettings.PageOrder = $LtsPageOrderFrontToBack
```

```
End If
```

```
.Print
```

'Example: PageToUseLayoutOn property

'This example sets the header to print beginning on page 2.

```
.Page.Layout.RightPage.Header.ConditionType = $LwpConditionTypeStartatpage
```

```
.Page.Layout.RightPage.Header.PageToUseLayoutOn = 2
```

```
'Example: PageUp method
' This example simulates pressing the page down and page up keys in the
' current document;
' RUNTIME DEPENDENCIES: You must have a document open for this script to work.

.PageDown
MessageBox "Click OK to pageup.",MB_OK,"Example Script"
.PageUp
```

'Example: Page property

'This example displays a message box with the width and height of the
'current page.

PgHeight = **.Page**.Height / 1440

PgWide = **.Page**.Width / 1440

MessageBox "The page height is " & PgHeight & " inches, and the width is " & PgWide &
" inches.", MB_OK, "Example Script"

```
'Example: PaperNames property
'This example prints the available page sizes to the LotusScript output panel.

For i = 0 To .ActiveDocument.PrintManager.PaperNames.Count - 1
    Print .ActiveDocument.PrintManager.PaperNames(i)
Next i
```

'Example: PaperName property

'This example displays an input box with the current paper name. If the
'name is changed, the new value is stored.

```
OldName = .Layout.RightPage.PaperName
```

```
NewPaperName = Inputbox("Type the new desired paper type:", "Example Script", OldName)
```

```
If NewPaperName <> "" Then ' user did not cancel
```

```
    .Layout.RightPage.PaperName = NewPaperName
```

```
End If
```


'Example: ParagraphBorder property

'This example places lines around the current paragraph.

```
.Text.Paragraphborder.RevertToStyle $LwpLineRevertAll ' reset everything
.Text.Paragraphborder.MarginLeft = 144 ' distance between lines and text
.Text.Paragraphborder.MarginRight = 144
.Text.Paragraphborder.MarginBottom = 144
.Text.Paragraphborder.MarginTop = 144
' set the length of the border to the length of the text
.Text.Paragraphborder.TypeAbove = $LwpParaBorderWidthText
.Text.Paragraphborder.TypeBelow = $LwpParaBorderWidthText
.Text.Paragraphborder.TypeRight = $LwpParaBorderWidthText
.Text.Paragraphborder.BorderLines.LinePlacement = &Hf ' put border on all sides
.Text.Paragraphborder.BorderLines.AllBorders.Pattern = $LtsBorderPatternSolid
.Text.Paragraphborder.BorderLines.AllBorders.WidthInTwips = 20 ' 1 point line
```

```
'Example: ParagraphStyleName property  
'This example inserts an example paragraph using each available  
'paragraph style in the current division.
```

```
Forall Styles In .Division.Foundry.ParagraphStyles  
    .Text.ParagraphStyleName = Styles.Name  
    .Text.InsertText "This is an example of the " & .Text.ParagraphStyleName & "  
style.", True  
End Forall
```

```
'Example: ParagraphStyles property
'This example script copies some text to the application foundry.
' It then uses the ParagraphStyles property to list the name of the style
' used for the text on the LotusScript browser output panel.

.Type "This is some text to be copied"
.SelectParagraph
.CopySelection
Forall ParaStyles In .AppFoundry.ParagraphStyles
    Print ParaStyles.Name
End Forall
```

'Example: PasteLink method

' This example pastes an Ole object from its source application and creates
' a linked Ole object.

' RUNTIME DEPENDENCIES: You must have a document open for this script to work.

.PasteLink "LWPOLEFormat", False, 0

Note OLE is not supported under OS/2.

```
'Example: PasteSpecial method
' This example pastes any CF_TEXT data from the clipboard.
' RUNTIME DEPENDENCIES: You must have a document open for this script to work.
```

```
FormatCategories = .Text.GetPasteFormatCategories
If (FormatCategories And &H1) Then
    .PasteSpecial "CF_TEXT"
End If
```

Note OLE is not supported under OS/2.

'Example: Paste method

' This example inserts some text into the current document. The text is then
' selected and cut to the clipboard. After the message box is closed, the
' text is then pasted back into the document.
' RUNTIME DEPENDENCIES: You must have a document open for this script to work.

.Text.InsertText "This is some sample text."

.Text.Select \$LwpSelectObjectTypeParagraph

.CutSelection

Messagebox "Click OK paste removed text.",MB_OK,"Example Script"

.Paste

'Example: Pattern property

'This uses a wavy pattern to create lines around the page.

```
%include "WPBITMSK.LSS" ' lets us use bitmask constants
```

```
.Page.Layout.RightPage.BorderLines.LinePlacement = LwpLinePlacementAllsides
```

```
.Page.Layout.RightPage.BorderLines.AllBorders.Pattern = $LwpBorderPatternWavy
```

```
.Page.Layout.RightPage.BorderLines.AllBorders.WidthInTwips = 80
```

'This creates a frame on the current page, and sets the background pattern

'of the frame to a right-to-left fill. The pattern color is set to yellow.

```
.NewFrame 1851, 1746, 2247, 2502, "Default Frame"
```

```
.Frame.Layout.Background.Pattern = $LtsFillRtLeftGrad
```

```
.Frame.Layout.Background.BackColor.Red = 255
```

```
.Frame.Layout.Background.BackColor.Blue = 0
```

```
.Frame.Layout.Background.BackColor.Green = 255
```

```
.Frame.Layout.Background.BackColor.Override = $LwpColorOverrideRgb
```

```
'Example: PColConnectCells method
' This example creates a parallel column table with 3 columns. The first row
' is then selected and connected.
' RUNTIME DEPENDENCIES: You must have a document open for this script to work.

.CreateParallelColumns 3, $LtsAlignmentHorizCenter
.PColSelectRow
.PColConnectCells
```


'Example: PColConnectRows method

' This example creates a parallel column table with 3 columns and then
' connects the first row.

```
.CreateParallelColumns 3, $LtsAlignmentHorizCenter  
.PColConnectRows
```

```
'Example: PColDisconnectCells method
' This example creates a parallel column table with 3 columns. The first row
' is then selected and connected. After the message box is closed the cells
' are then disconnected.
' RUNTIME DEPENDENCIES: You must have a document open for this script to work.

.CreateParallelColumns 3, $LtsAlignmentHorizCenter
.PColSelectRow
.PColConnectCells
MessageBox "Click OK to disconnect cells ",MB_OK,"Example Script"
.PColDisconnectCells
```

'Example: PColSelectColumn method

' This example creates a parallel column table with 3 columns and then selects
' the first column.

' RUNTIME DEPENDENCIES: You must have a document open for this script to work.

.CreateParallelColumns 3, \$LtsAlignmentHorizCenter

.PColSelectColumn

'Example: PColSelectRow method

' This example creates a parallel column table with 3 columns and then selects
' the first row.

' RUNTIME DEPENDENCIES: You must have a document open for this script to work.

.CreateParallelColumns 3, \$LtsAlignmentHorizCenter

.PColSelectRow

```
'Example: PColSelectTable method
' This example creates a parallel column block with 3 columns. The first row
' is then selected and connected. Text is typed into the selected row. A new
' row is inserted and then the entire parallel column area is selected.
' RUNTIME DEPENDENCIES: You must have a document open for this script to work.

.CreateParallelColumns 3, $LtsAlignmentHorizCenter
.PColSelectRow
.PColConnectCells
.Type "This is a parallel column."
.BaseTable.InsertRowOrColumn $LwpTableInsTypeRow, True, 1
.PColSelectTable
```

'Example: PersonalData1 property

'This example retrieves the current value for personal information,

'lets the user change it, then stores the new value.

```
CurrentValue = .Preferences.PersonalData1
```

```
NewValue = Inputbox ("What would you like to use for personal information?", "Example  
Script", CurrentValue)
```

```
.Preferences.PersonalData1 = NewValue
```

'Example: **PersonalData2** property

'This example retrieves the current value for personal information,

'lets the user change it, then stores the new value.

```
CurrentValue = .Preferences.PersonalData2
```

```
NewValue = Inputbox ("What would you like to use for personal information?", "Example  
Script", CurrentValue)
```

```
.Preferences.PersonalData2 = NewValue
```

'Example: PersonalData3 property

'This example retrieves the current value for personal information,

'lets the user change it, then stores the new value.

```
CurrentValue = .Preferences.PersonalData3
```

```
NewValue = Inputbox ("What would you like to use for personal information?", "Example  
Script", CurrentValue)
```

```
.Preferences.PersonalData3 = NewValue
```


'Example: `PersonalData4` property

'This example retrieves the current value for personal information,

'lets the user change it, then stores the new value.

```
CurrentValue = .Preferences.PersonalData4
```

```
NewValue = Inputbox ("What would you like to use for personal information?", "Example  
Script", CurrentValue)
```

```
.Preferences.PersonalData4 = NewValue
```

'Example: PhoneNumber property

'This example retrieves the current value for the user's phone number,
'lets the user change it, then stores the new value.

```
CurrentValue = .Preferences.PhoneNumber
```

```
NewValue = Inputbox ("What is the correct phone number?", "Example Script",  
CurrentValue)
```

```
.Preferences.PhoneNumber = NewValue
```

```
'Example: PositionType property
Dim CR As String*1
Dim IcnPallet As String
Dim MsgStr As String
Dim IcnMgr As IconBarManager

IcnPallet = "Comment Tools"
CR = Chr(10)
Set IcnMgr = .ApplicationWindow.IconBarManager

With IcnMgr.IconBars(IcnPallet)
    MsgStr = "Height = " & .Height & CR
    MsgStr = MsgStr & "IconBarPositionState = " & .IconBarPositionState & CR
    MsgStr = MsgStr & "PositionType = " & .PositionType & CR
    MsgStr = MsgStr & "ScreenPositionX = " & .ScreenPositionX & CR
    MsgStr = MsgStr & "ScreenPositionY = " & .ScreenPositionY
    MessageBox MsgStr, 64, "Script Example - " & .Name
End With
```

'Example: PositionXInContainer property

'This example creates a frame on the current page, and displays a message
'box with the position of the insertion point relative to the page and
'the frame container.

```
Dim px As Single, py As Single, cx As Single, cy As Single
```

```
.NewFrame 3039, 2656, 3389, 6555, "Default Frame"
```

'since these properties return twips, divide by 1440 for inches, and round
'to two decimal places

```
px = Round(.Text.PositionXOnPage / 1440, 2)
```

```
py = Round(.Text.PositionYOnPage / 1440, 2)
```

```
cx = Round(.Text.PositionXInContainer / 1440, 2)
```

```
cy = Round(.Text.PositionYInContainer / 1440, 2)
```

```
MessageBox "The insertion point is " & px & " inches from the left margin of the  
page," & Chr(10) & _
```

```
py & " inches from the top margin of the page," & Chr(10) & _
```

```
cx & " inches from the left margin of the frame," & Chr(10) & _
```

```
"and " & cy & " inches from the top margin of the frame." , MB_OK, "Example Script"
```

'Example: PositionXOnPage property

'This example creates a frame on the current page, and displays a message
'box with the position of the insertion point relative to the page and
'the frame container.

```
Dim px As Single, py As Single, cx As Single, cy As Single
```

```
.NewFrame 3039, 2656, 3389, 6555, "Default Frame"
```

'since these properties return twips, divide by 1440 for inches, and round
'to two decimal places

```
px = Round(.Text.PositionXOnPage / 1440, 2)
```

```
py = Round(.Text.PositionYOnPage / 1440, 2)
```

```
cx = Round(.Text.PositionXInContainer / 1440, 2)
```

```
cy = Round(.Text.PositionYInContainer / 1440, 2)
```

```
MessageBox "The insertion point is " & px & " inches from the left margin of the  
page," & Chr(10) & _
```

```
py & " inches from the top margin of the page," & Chr(10) & _
```

```
cx & " inches from the left margin of the frame," & Chr(10) & _
```

```
"and " & cy & " inches from the top margin of the frame." , MB_OK, "Example Script"
```

'Example: PositionYInContainer property

'This example creates a frame on the current page, and displays a message
'box with the position of the insertion point relative to the page and
'the frame container.

```
Dim px As Single, py As Single, cx As Single, cy As Single
```

```
.NewFrame 3039, 2656, 3389, 6555, "Default Frame"
```

'since these properties return twips, divide by 1440 for inches, and round
'to two decimal places

```
px = Round(.Text.PositionXOnPage / 1440, 2)
```

```
py = Round(.Text.PositionYOnPage / 1440, 2)
```

```
cx = Round(.Text.PositionXInContainer / 1440, 2)
```

```
cy = Round(.Text.PositionYInContainer / 1440, 2)
```

```
MessageBox "The insertion point is " & px & " inches from the left margin of the  
page," & Chr(10) & _
```

```
py & " inches from the top margin of the page," & Chr(10) & _
```

```
cx & " inches from the left margin of the frame," & Chr(10) & _
```

```
"and " & cy & " inches from the top margin of the frame." , MB_OK, "Example Script"
```

'Example: PositionYOnPage property

'This example creates a frame on the current page, and displays a message
'box with the position of the insertion point relative to the page and
'the frame container.

```
Dim px As Single, py As Single, cx As Single, cy As Single
```

```
.NewFrame 3039, 2656, 3389, 6555, "Default Frame"
```

'since these properties return twips, divide by 1440 for inches, and round
'to two decimal places

```
px = Round(.Text.PositionXOnPage / 1440, 2)
```

```
py = Round(.Text.PositionYOnPage / 1440, 2)
```

```
cx = Round(.Text.PositionXInContainer / 1440, 2)
```

```
cy = Round(.Text.PositionYInContainer / 1440, 2)
```

```
MessageBox "The insertion point is " & px & " inches from the left margin of the  
page," & Chr(10) & _
```

```
py & " inches from the top margin of the page," & Chr(10) & _
```

```
cx & " inches from the left margin of the frame," & Chr(10) & _
```

```
"and " & cy & " inches from the top margin of the frame." , MB_OK, "Example Script"
```

'Example: Position property

[Alignment]

' This example sets up numeric alignment, with the decimal point 1/2 inch from the right margin.

```
.Text.Alignment.AlignmentType = $LwpAlignmentTypeNumericright
```

```
.Text.Alignment.Position = 720 ' 720 twips = 1/2 inch
```

[OutSeqItem]

'This example creates an outline style sequence, adds three items to it, and sets their sequence positions.

'First, create the sequence, name it, and make it empty

```
NewSequence = .Division.Foundry.Create($LwpFoundryCreateTypeOutlineseq)
```

```
.Division.Foundry.OutlineStyleSequences(NewSequence).Name = "Typical Outline"
```

```
.Division.Foundry.OutlineStyleSequences("Typical Outline").Clear
```

```
.Division.Foundry.OutlineStyleSequences("Typical Outline").Heading = 1
```

' Now, add a sequence item for the Heading 1 paragraph style

```
.Division.Foundry.OutlineStyleSequences("Typical Outline").AddOutlineSequenceItem  
"Heading 1"
```

' And set it's position

```
.Division.Foundry.OutlineStyleSequences("Typical  
Outline").OutlineSeqItems("Heading 1").Position = 1
```

'Add Heading 2 and Heading 3 paragraph styles

```
.Division.Foundry.OutlineStyleSequences("Typical Outline").AddOutlineSequenceItem  
"Heading 2"
```

```
.Division.Foundry.OutlineStyleSequences("Typical  
Outline").OutlineSeqItems("Heading 2").Position = 2
```

```
.Division.Foundry.OutlineStyleSequences("Typical Outline").AddOutlineSequenceItem  
"Heading 3"
```

```
.Division.Foundry.OutlineStyleSequences("Typical  
Outline").OutlineSeqItems("Heading 3").Position = 3
```


'Example: PowerField property

'This example asks the user whether to display power field formulas

'in the current document, and then sets the appropriate option.

```
stat = MessageBox ("Do you want to view power field formulas?", 36, "Example Script")
```

```
If stat = 6 Then ' user said yes
```

```
    .ActiveDocument.DocOptions.PowerField = 1
```

```
Else
```

```
    .ActiveDocument.DocOptions.PowerField = 0
```

```
End If
```

'Example: Prefix property

'This example creates a table, types a number in the first cell, then
'formats the cell for US currency, inserting a dollar sign as the prefix.

```
.CreateTable False, "Default Table", 4, 4  
.Type "123.45678"  
.Table.CurrentCell.NumericFormat.NumDecimalPlaces = 2  
.Table.CurrentCell.NumericFormat.AnyNumber.Prefix = "$"  
.Table.CurrentCell.NumericFormat.AnyNumber.Suffix = ""  
.type "[Down]"
```

'Example: Presentation property

'This example inserts outline text into a document, then shows only level 2

'headings and higher. After clicking OK, all text is shown again.

```
.Text.InsertText "This is outline level 1.", True
```

```
.Text.Numbering.Level = 2
```

```
.Text.InsertText "This is outline level 2.", True
```

```
.Text.Numbering.Level = 3
```

```
.Text.InsertText "This is outline level 3.", True
```

```
.Text.Numbering.Level = 4
```

```
.Text.InsertText "This is outline level 4.", True
```

```
.Text.Numbering.Level = 2
```

```
.Text.InsertText "This is outline level 2.", True
```

```
.Page.Presentation.OutlineLevel = 2
```

```
MessageBox "Click OK to show all outline text.", MB_OK, "Example Script"
```

```
.Page.Presentation.OutlineLevel = 255
```

'Example: PrintDestination property

'This example script sets the current document's printer to the HP LaserJet
'4L, printing to LPT1:.

```
.ActiveDocument.PrintManager.PrintDestination = "LPT1:"
```

```
.ActiveDocument.PrintManager.PrinterName = "HP LaserJet 4L"
```

```
.ActiveDocument.PrintManager.UseDefaultPrinter = True
```

```
.ActiveDocument.PrintManager.UpdatePrinterBins
```

```
.ActiveDocument.PrintManager.GetDefaultPageSize
```

'Example: PrinterName property

'This example script sets the current document's printer to the HP LaserJet
'4L, printing to LPT1:.

```
.ActiveDocument.PrintManager.PrintDestination = "LPT1:"  
.ActiveDocument.PrintManager.PrinterName = "HP LaserJet 4L"  
.ActiveDocument.PrintManager.UseDefaultPrinter = True  
.ActiveDocument.PrintManager.UpdatePrinterBins  
.ActiveDocument.PrintManager.GetDefaultPageSize
```

```
'Example: PrintGraphics property
'This example asks the user whether to print graphics in the document,
'sets the appropriate option, then prints the document.

stat = MessageBox ("Do you want to print pictures?", 36, "Example Script")
If stat = 6 Then ' user said yes
    .ActiveDocument.PrintSettings.PrintGraphics = 1
Else
    .ActiveDocument.PrintSettings.PrintGraphics = 0
End If
.Print
```

'Example: PrintManager property

'This example sets the current document to use the default printer.

```
.ActiveDocument.PrintManager.PrintDestination = "LPT1:"
```

```
.ActiveDocument.PrintManager.PrinterName = "HP LaserJet 4L"
```

```
.ActiveDocument.PrintManager.UseDefaultPrinter = True
```

```
.ActiveDocument.PrintManager.UpdatePrinterBins
```

```
.ActiveDocument.PrintManager.GetDefaultPageSize
```

'Example: PrintOut method

' This example prints the current page of the current document.

' RUNTIME DEPENDENCIES: You must have an open document configured

' to an available printer and for this script to work.

.ActiveDocument.PrintSettings.PrintRange = \$LtsPrintRangeCurrentPage

.PrintOut


```
'Example: PrintPagesFrom property
'This example prints the first two pages of a three page document.

'Create some pages to work with.
.NewDocument
.Text.InsertText "This is page 1"
.InsertPageBreak
.Text.InsertText "This is page 2"
.InsertPageBreak
.Text.InsertText "This is page 3"

.ActiveDocument.PrintSettings.PrintPagesFrom = 1
.ActiveDocument.PrintSettings.PrintPagesTo = 2
.ActiveDocument.PrintSettings.PrintRange = $LwpPrintRangeRangeOfPages
.Print
```

'Example: PrintPagesTo property

'This example prints the first two pages of a three page document.

'Create some pages to work with.

.NewDocument

.Text.InsertText "This is page 1"

.InsertPageBreak

.Text.InsertText "This is page 2"

.InsertPageBreak

.Text.InsertText "This is page 3"

.ActiveDocument.PrintSettings.PrintPagesFrom = 1

.ActiveDocument.PrintSettings.**PrintPagesTo** = 2

.ActiveDocument.PrintSettings.PrintRange = \$LwpPrintRangeRangeOfPages

.Print

```
'Example: PrintPageType property
'This example creates a 3 page document, then prints the odd pages
'of the document.

' set up some pages to work with
.NewDocument
.Text.InsertText "This is page 1"
.InsertPageBreak
.Text.InsertText "This is page 2"
.InsertPageBreak
.Text.InsertText "This is page 3"

.ActiveDocument.PrintSettings.PrintRange = $LtsPrintRangeAllPages
.ActiveDocument.PrintSettings.PrintPageType = $LwpPrintPageOddPages
.Print
```

'Example: PrintRange property

'This example prints the first two pages of a three page document.

'Create some pages to work with.

.NewDocument

.Text.InsertText "This is page 1"

.InsertPageBreak

.Text.InsertText "This is page 2"

.InsertPageBreak

.Text.InsertText "This is page 3"

.ActiveDocument.PrintSettings.PrintPagesFrom = 1

.ActiveDocument.PrintSettings.PrintPagesTo = 2

.ActiveDocument.PrintSettings.**PrintRange** = \$LwpPrintRangeRangeOfPages

.Print

'Example: PrintSettings property
'This example sets the document's print settings to print page 3, then
'prints the document.

```
.NewDocument  
.Text.InsertText "This is page 1"  
.InsertPageBreak  
.Text.InsertText "This is page 2"  
.InsertPageBreak  
.Text.InsertText "This is page 3"  
.ActiveDocument.Printsettings.Copies = 1  
.ActiveDocument.Printsettings.PrintPagesFrom = 3  
.ActiveDocument.Printsettings.PrintPagesTo = 3  
.ActiveDocument.Printsettings.PrintRange = $LtsPrintRangeCurrentPage  
.ActiveDocument.Printsettings.PrintPageType = $LwpPrintPageEvenAndOddPages  
.ActiveDocument.Printsettings.Collate = False  
.ActiveDocument.Printsettings.OutputToFile = False  
.Print
```

'Example: Print method

' This example prints the current page of the current document.

' RUNTIME DEPENDENCIES: You must have an open document configured

' to an available printer and for this script to work.

.ActiveDocument.PrintSettings.PrintRange = \$LtsPrintRangeCurrentPage

.Print

'Example: PromptHidden property

'This example creates a click here block in the current document, then

'hides the prompt text.

```
MyClickHere = .InsertClickHere()  
.Division.Foundry.ClickHeres(MyClickHere).Prompt.Clear  
.Division.Foundry.ClickHeres(MyClickHere).Prompt.InsertText "Click here to type Text",  
False  
.Division.Foundry.ClickHeres(MyClickHere).Action = 1  
.Division.Foundry.ClickHeres(MyClickHere).Name = MyClickHere  
  
MessageBox "Click OK to hide the Click Here prompt text", MB_OK, "Example Script"  
.Division.Foundry.ClickHeres(MyClickHere).PromptHidden = True
```

```
'Example: Prompt property
' This example inserts a ClickHere block in the current document and assigns
' the return value (the ClickHere ID) to a variable (NewClickHereId).
' The script then stores the new ClickHere in a variable (NewClickHere) and
' inserts some text in it.
' RUNTIME DEPENDENCIES: You must have a document open for this script to work.
```

```
Dim NewClickHereId as String
Dim NewClickHere as ClickHere
```

```
NewClickHereId = .InsertClickHere()
Set NewClickHere = .Division.Foundry.ClickHeres(NewClickHereId)
NewClickHere.Prompt.Clear ' first delete the old prompt
NewClickHere.Prompt.InsertText "Click Here to type text "
```



```
'Example: ProtectedMode property
'This example protects some text, then enables protection for the text.
.Text.Attributes.ProtectedMode = True ' turn on protect attribute
.Division.DivisionOptions.IsTextLocked = False ' disable protection for text
.Type "Now is the time for all good men to come to the aid of their party."
.Text.Attributes.ProtectedMode = False
.Division.DivisionOptions.IsTextLocked = True ' now honor protection
```

```
'Example: QuickAlignFrame method
' This example creates a frame. After the message box is closed, the
' frame is then horizontally centered.
' RUNTIME DEPENDENCIES: You must have a document open for this script to work.
```

```
Dim FrameWidth as Integer
Dim FrameHeight as Integer
Dim X as Integer
Dim Y as Integer
```

```
FrameWidth = 1440
FrameHeight = 1440
X = 1440
Y = 1440
```

```
.InsertFrame FrameWidth,FrameHeight, X, Y
Messagebox "Click OK to center frame.",MB_OK,"Example Script"
```

```
.QuickAlignFrame $LwpQuickLayoutAlignHorzcenter
```

```
'Example: QuickAlignTable method
' This example creates a table with 4 rows and 5 columns based upon the
' 'Default Table' style and then changes the alignment attribute to
' left aligned.
' RUNTIME DEPENDENCIES: You must have a document open for this script to work.

.CreateTable False, "Default Table", 5, 4

.QuickAlignTable $LwpQuickLayoutAlignLeft
```

```
'Example: Quit method  
' This example closes WordPro.  
' RUNTIME DEPENDENCIES: None.
```

```
.Quit
```

```
'Example: ReadOnly property
'This example toggles the read-only status of the current document.

If .ActiveDocument.ReadOnly = True Then
    .ActiveDocument.ReadOnly = False
    MessageBox "This document is now read-write.", MB_OK, "Example Script"
Else
    .ActiveDocument.ReadOnly = True
    MessageBox "This document is now read-only.", MB_OK, "Example Script"
End If
```

```
'Example: Read method
' This example creates a bag in the active division and then writes some data
' to the bag. The data from the created bag is read and printed to the Lotus
' Script Output panel. Next, data from all bags in the Bag Collection is
' printed.
' RUNTIME DEPENDENCIES: You must have a document open for this script to work.
```

```
Dim BagName As String
Dim MyBag As Bag
Dim BagData As String
```

```
BagData = "This is data for the bag."
LenBagData = Len(BagData)
```

```
BagName = .Division.Foundry.Create($LwpFoundryCreateTypeBag)
Set MyBag = .Division.Foundry.Bags.Item(BagName)
```

```
Stat = MyBag.Write(BagData, LenBagData)
If Stat = True Then
    Print "BagData= " & MyBag.Read(LenBagData)
End If
```

```
Forall ThisBag In .Division.Foundry.Bags
    ThisBag.Reset
    Print "Name = " ThisBag.Name
    Print "Length = " ThisBag.Length
    Print ThisBag.Read(ThisBag.Length)
End Forall
```

'Example: Redo method

' This example types some text into the current document which is then

' undone and redone.

' RUNTIME DEPENDENCIES: You must have a document open for this script to work.

.Type "Typing this text will be be undone"

.Undo

.Redo

'Example: Red property

'This example changes the font color to red, and inserts some text into
'the document. The color then reverts to the style color.

```
.Text.Font.FontColor.Red = 255
```

```
.Text.Font.FontColor.Blue = 0
```

```
.Text.Font.FontColor.Green = 0
```

```
.Text.Font.FontColor.Override = $LwpColorOverrideRgb
```

```
.Text.InsertText "This is an example of red text. ", True
```

```
.Text.Font.FontColor.RevertToStyle
```



```
'Example: RegisterWPDataSet method
' This example creates a dataset named 'PhoneNumbers' off of the application
' object. Two dataset items are added and then printed to the Script Editor
' Output panel. The dataset is then removed.
' RUNTIME DEPENDENCIES: You must have a document open for this script to work.
```

```
Dim AppDataSet As WPDataSet
```

```
Set AppDataSet = .RegisterWPDataSet("PhoneNumbers")
```

```
AppDataSet.SetData "Mark", "555-1234"
```

```
AppDataSet.SetData "Peyton", "555-5678"
```

```
Print AppDataSet.GetData("Mark", " ")
```

```
Print AppDataSet.GetData("Peyton", " ")
```

```
.UnRegisterWPDataSet "PhoneNumbers"
```

```
'Example: RelativePageNum property
'This example creates some pages and inserts a division in the current
'document, then displays the page number of the division and the page
'number of the document.

.ActiveDocWindow.WinViewPrefs.IsViewSectionTabs = True
.InsertPageBreak
.InsertPageBreak
.ApplicationWindow.SectionTabs.AddNewSectionTabs
MessageBox "This is page " & .Page.RelativePageNum & " of the division, and page "
-
    & .Page.PageNum & " of the document.", MB_OK, "Example Script"
```

'Example: RelativeXDistance property

'This example creates a frame, then sets the frame's anchor offsets to

'one inch down from the anchor point, and two inches down from the anchor point.

```
.NewFrame 2039, 639, 3089, 1572, "Default Frame"
```

```
.Frame.Layout.RelativeXDistance = 2880
```

```
.Frame.Layout.RelativeYDistance = 1440
```

'Example: RelativeYDistance property

'This example creates a frame, then sets the frame's anchor offsets to

'one inch down from the anchor point, and two inches down from the anchor point.

```
.NewFrame 2039, 639, 3089, 1572, "Default Frame"
```

```
.Frame.Layout.RelativeXDistance = 2880
```

```
.Frame.Layout.RelativeYDistance = 1440
```

'Example: Relative property

'This example indents the current paragraph 1/2 inch relative to the
'indent level of the preceding paragraph's "All" indent value.

.Text.Indent.All = 720

.Text.Indent.UseRelative = True

.Text.Indent.**Relative** = \$LwpIndentPropertyAll

'Example: ReleaseNumber property

'This example displays the Word Pro release number in a message box.

```
MessageBox "This is Word Pro Version " & .ReleaseNumber, MB_OK, "Example Script"
```

```
'Example: Release method
' This example creates a new paragraph style and then deletes the style.
' RUNTIME DEPENDENCIES: You must have a document open for this script to work.
Dim StyleName As String
StyleName = "My New Char Style"
Style = .Division.Foundry.Create($LwpFoundryCreateTypeStyle, StyleName, 39)
With .Division.Foundry.CharacterStyles(Style)
    .Font.Underline = True
    .Font.FontColor.Blue = 128
    .Font.FontColor.Red = 128
    .Font.FontColor.Green = 0
End With
.Foundry.Release $LwpFoundryReleaseTypeStyle, 35, StyleName
```

```
'Example: RemoveBookmark method
' This example creates a new bookmark named "NewBookMark" and then removes it.
' RUNTIME DEPENDENCIES: You must have a document open for this script to work.

Dim MarkName As String

' Get the marker name for the currently selected text.
MarkName = .Mark($LwpMarkerTypeBookmark)

' Create the bookmark from the given marker name.
.Division.BookmarkManager.AddBookmark "NewBookMark", MarkName

MessageBox "Click OK to remove the bookmark. ",MB_OK,"Example Script"

' Get the marker name for this bookmark
MarkName = .Division.BookmarkManager.Bookmarks("NewBookMark").MarkerName

' The Bookmark Manager removes bookmarks by their marker name.
.Division.BookmarkManager.RemoveBookmark (MarkName)
```



```
'Example: RemoveDataFile method
' This example creates a data file for the current document.  Two records are
' added and the data file is then removed from the current document.
' RUNTIME DEPENDENCIES: You must have a document open for this script to work.

.CreateDataFile "~|", "Name~Address~City~State~Zip|", False, "C:\mergedat.lwp"

'Add some records to the virtual datafile
.MergeAddDataRecord "Jane Doe~100 Main St.~ Atlanta~ GA~30319|"
.MergeAddDataRecord "John Doe~100 Main St.~ Atlanta~ GA~30319|"

.ActiveDocument.MergeOptions.RemoveDataFile
```

'Example: RemoveDivision method

' This example creates and then removes a new division in the current
' document.

' RUNTIME DEPENDENCIES: You must have a document open for this script to work.

```
DivisionId = .ActiveDocument.AddDivision("NewDivision")  
.ActiveDocument.RemoveDivision DivisionId
```

```
'Example: RemoveEditor method  
' This example adds a new editor with read only rights to the current  
' document.  
' RUNTIME DEPENDENCIES: You must have a document open for this script to work.
```

```
Dim NewEditorName As String  
Dim NewEditorInitials As String
```

```
NewEditorName = "Lotus User"  
NewEditorInitials = "LU"  
.ActiveDocument.EditorManager.AddEditorManager NewEditorName, NewEditorInitials  
.ActiveDocument.EditorManager.RemoveEditor NewEditorName
```

```
'Example: RemoveNamedProperty method
' This example creates a named property, 'ExampleProp' on the active document
' and assigns it a value which is then printed to the Lotus Script Output
' panel. The named property is then removed from the active document.
' RUNTIME DEPENDENCIES: You must have a document open for this script to work.
```

```
.Division.SetNamedProperty "ExampleProp", "Here is some data."
Print .Division.GetNamedProperty("ExampleProp")
MessageBox "Click OK to remove the named property. ",MB_OK,"Example Script"
.Division.RemoveNamedProperty "ExampleProp"
```

'Example: RenderedPageNumber property

'This example inserts a page number using the Roman Numeral style. A message
'box then displays the current page number, and the page number as
'it prints.

```
.GoToObject "Header", True ' insert page numbering in the header  
.InsertPageNumber $LwpNumberingStyleUppercaseroman, "", ".", 0, 1, &H0  
.Header.GoToContainer $LwpGoToLocationExit ' return to the document
```

```
MessageBox "This is page number " & .Page.PageNum & ", which is displayed as " _  
& .Text.RenderedPageNumber, MB_OK, "Example Script"
```

```
'Example: ReplaceAll method
' This example inserts three identical sentences into the current document,
' clears the FindAndReplace settings, sets the FindString to "cat" and the
' ReplaceString to "dog", and then displays a message box.
' When you click OK, the script replaces all the cats with dogs.
' RUNTIME DEPENDENCIES: You must have a document open for this script to work.

Dim SentenceCount As Integer
For SentenceCount = 1 To 3
    .Text.InsertText "The sleep-deprived fox jumped over the gravity-challenged cat."
    .Text.SplitParagraph
Next
.Application.ResetFindAndReplace
.Application.FindAndReplace.FindString = "cat"
.Application.FindAndReplace.ReplaceString = "dog"
.InitFindAndReplace True
MessageBox "Click OK to replace all.",MB_OK,"Example Script"

.ReplaceAll
```

```
'Example: ReplaceCmd method
' This example inserts three identical sentences into the current document,
' clears the FindAndReplace settings, sets the FindString to "cat" and the
' ReplaceString to "dog", and then displays a message box.
' When you click OK, the script finds the first 'cat' then replaces it with 'dog'.
' RUNTIME DEPENDENCIES: You must have a document open for this script to work.

Dim SentenceCount As Integer
For SentenceCount = 1 To 3
    .Text.InsertText "The sleep-deprived fox jumped over the gravity-challenged cat."
    .Text.SplitParagraph
Next
.Application.ResetFindAndReplace
.Application.FindAndReplace.FindString = "cat"
.Application.FindAndReplace.ReplaceString = "dog"
.InitFindAndReplace True
For i = 1 to 2
    Messagebox "Click OK to find and then replace.",MB_OK,"Example Script"
    .Find

    .ReplaceCmd
Next I
```

```
'Example: ReplaceString property
' This example inserts three identical sentences into the current document,
' clears the FindAndReplace settings, sets the FindString to "cat" and the
' ReplaceString to "dog", and then displays a message box.
' When you click OK, the script finds the first 'cat' then replaces it with
' 'dog'.
' RUNTIME DEPENDENCIES: You must have a document open for this script to work.

Dim SentenceCount As Integer
For SentenceCount = 1 To 3
    .Text.InsertText "The sleep-deprived fox jumped over the gravity-challenged cat."
    .Text.SplitParagraph
Next
.Application.ResetFindAndReplace
.Application.FindAndReplace.FindString = "cat"

.Application.FindAndReplace.ReplaceString = "dog"

.InitFindAndReplace True
MessageBox "Click OK to find and then replace.",MB_OK,"Example Script"
.Find
.ReplaceCmd
```



```
'Example: Replace method
' This example inserts three identical sentences into the current document,
' clears the FindAndReplace settings, sets the FindString to "cat", and then displays
a message box.
' When you click OK, the script finds the first 'cat' then replaces it with 'dog'.
' RUNTIME DEPENDENCIES: You must have a document open for this script to work.

Dim SentenceCount As Integer
For SentenceCount = 1 To 3
    .Text.InsertText "The sleep-deprived fox jumped over the gravity-challenged cat."
    .Text.SplitParagraph
Next
.Application.ResetFindAndReplace
.Application.FindAndReplace.FindString = "cat"
.InitFindAndReplace True
MessageBox "Click OK to find and then replace.",MB_OK,"Example Script"
.Find

.Replace $LwpReplaceObjectTypeWord, "dog"
```

'Example: RequestRemarkOnClose property
'This example asks the user whether Word Pro should request an editor's
'remark when the current document is closed, and then sets the appropriate option.

```
stat = MessageBox ("Do you want to prompt for a remark?", 36, "Example Script")
If stat = 6 Then ' user said yes
    .ActiveDocument.DocControl.RequestRemarkOnClose = True
True
Else
    .ActiveDocument.DocControl.RequestRemarkOnClose = False
End If
```

```
'Example: RequireStartupScripts property
'This example asks the user whether to require running document startup scripts,
'and then sets the appropriate option.

stat = MessageBox ("Do you want to require running startup scripts?", 36, "Example
Script")
If stat = 6 Then ' user said yes
    .ActiveDocument.DocControl.RequireStartupScripts = True
Else
    .ActiveDocument.DocControl.RequireStartupScripts = False
End If
```

```
'Example: ResetFindAndReplace method
' This example inserts three identical sentences into the current document,
' clears the FindAndReplace settings, sets the FindString to "cat" and then
' displays a message box. When you click OK, the script finds the first 'cat'.
' RUNTIME DEPENDENCIES: You must have a document open for this script to work.

Dim SentenceCount As Integer
For SentenceCount = 1 To 3
    .Text.InsertText "The sleep-deprived fox jumped over the gravity-challenged cat."
    .Text.SplitParagraph
Next

.Application.ResetFindAndReplace
.Application.FindAndReplace.FindString = "cat"
.InitFindAndReplace True
MessageBox "Click OK to find the string.",MB_OK,"Example Script"
.Find
```

'Example: ResetOnEachPage property

'This example numbers lines in the current document. The line numbers
'are reset on each page, and only number every fifth line.

```
.Division.LineNumberOptions.ResetOnEachPage = True
```

```
.Division.LineNumberOptions.NumberWhichLines = $LwpLineNumberOptsSpecifiedLines
```

```
.Division.LineNumberOptions.NumberEveryNthLine = 5
```

'Example: ResetWhen property

'This example sets the option for restarting footnotes and endnotes.

.Division.FootnoteOptions.FootnoteNumbering.**ResetWhen** = \$LwpResetOptionEachPage

.Division.FootnoteOptions.EndnoteDivisionNum.**ResetWhen** = \$LwpResetOptionEachDivision

.Division.FootnoteOptions.EndnoteDivisionGroupNum.**ResetWhen** =
\$LwpResetOptionEachDivisiongroup

.Division.FootnoteOptions.EndnoteDocNum.**ResetWhen** = \$LwpResetOptionEachDoc

```
'Example: Reset method
' This example creates a bag in the active division and then writes some data
' to the bag. The data from the created bag is read and printed to the Lotus
' Script Output panel. Next, data from all bags in the Bag Collection is
' printed.
' RUNTIME DEPENDENCIES: You must have a document open for this script to work.
```

```
Dim BagName As String
Dim MyBag As Bag
Dim BagData As String
```

```
BagData = "This is data for the bag."
LenBagData = Len(BagData)
```

```
BagName = .Division.Foundry.Create($LwpFoundryCreateTypeBag)
Set MyBag = .Division.Foundry.Bags.Item(BagName)
```

```
Stat = MyBag.Write(BagData, LenBagData)
If Stat = True Then
    Print "BagData= " & MyBag.Read(LenBagData)
End If
```

```
Forall ThisBag In .Division.Foundry.Bags
    ThisBag.Reset
    Print "Name = " ThisBag.Name
    Print "Length = " ThisBag.Length
    Print ThisBag.Read(ThisBag.Length)
End Forall
```

```
'Example: Resize method
' This example resizes the main application window.
' RUNTIME DEPENDENCIES: None
Dim AppWidth as Integer
Dim AppHeight as Integer
AppWidth = 8640
AppHeight = 5760
.ApplicationWindow.Resize AppWidth, AppHeight
```



```
'Example: RestartStyleName property
'This example redefines the Default Text style so that it becomes a numbered
'list, with numbers resetting following the Heading 1 style
'Text is inserted into the document to illustrate the situation
'Finally, the restarting style name is set to Heading 2. This causes the
' numbering to become continuous, rather than resetting at the Heading 1 paragraph.

.SetCustomNumber $LwpNumberResetonspecificstyle, "Heading 1", $LwpNumberIncludenone,
"", $LwpNumberingStyleBasic, 0, 0, ""
.SetStyle $LwpStyleTypeParagraph, "Default Text", &H0
.Division.Foundry.ParagraphStyles("Default Text").Update

.Text.InsertText "This is a numbered list.", True
.Text.InsertText "Another item for the list", True
.Text.ParagraphStyleName = "Heading 1"
.Text.InsertText "This is heading 1.", True
.Text.ParagraphStyleName = "Default Text"
.Text.InsertText "The numbered list starts again.", True
.Text.InsertText "Another item for the list", True
MessageBox "Click OK to change the numbering reset style", MB_OK, "Example Script"
.Text.Bullet.SilverBullet.RestartStyleName = "Heading 2"
```

```
'Example: Restore method  
' This example restores the main application window.  
.ApplicationWindow.Restore
```

```
'Example: Rest property
'This example types a paragraph and indents all lines but the first line 1/2 inch.
.NewDocument
For i = 1 To 20
    .type "Indention test "
Next
.Text.Indent.Rest = 720
MessageBox "Click OK to revert to the style indents.", MB_OK, "Example Script"
.Text.Indent.RevertToStyle
```

```
'Example: RetrieveInternetFileAndOpen method  
' This example retrieves and opens Lotus' home page from the Internet.  
' RUNTIME DEPENDENCIES: You must have a document open for this script to work.
```

```
Dim Location As String  
Dim Proxy As String  
Dim Port As Integer
```

```
Location = "http://www.lotus.com"  
Proxy = "123.456.78.910"  
Port = 1234  
.RetrieveInternetFileAndOpen Location, "", "", 1, Proxy, Port
```

```
'Example: RetrieveInternetFile method
' This example retrieves a stock symbol from the Internet.
' RUNTIME DEPENDENCIES: You must have a document open for this script to work.
```

```
Dim Location As String
Dim Proxy As String
Dim Port As Integer
```

```
Symbol = Inputbox ("What stock symbol","Retrieve Quote","IBM")
Location = "http://www.spacecom.com:8001/cgi-bin/getquote?TICKER=" + Symbol
Proxy = "123.456.78.910"
Port = 1234
```

```
FileName = .RetrieveInternetFile(Location,"","",1,Proxy,Port)
.OpenDocument Filename, "", "HTML", "", False, True
```

'Example: RevertToSaved method

' This example returns to the last saved version of the current document.

' RUNTIME DEPENDENCIES: You must have a document open for this script to work.

.RevertToSaved

```
'Example: RevertToStyle method
' This example inserts some sample text into the current document, selects
' the inserted text, modifies the font and alignment properties of that text,
' and then displays a message box.
' When you close the message box, the script uses the RevertToStyle method to
' cause all the text properties to revert back to the paragraph style assigned
' to that text.
' RUNTIME DEPENDENCIES: You must have a document open for this script to work.
```

```
With .Text
    .InsertText "This is some sample text."
    .Select $LwpSelectObjectTypeParagraph
    .Font.Size = 15.00
    .Font.Bold = True
    .Alignment.AlignmentType = $LtsAlignmentHorizCenter
    MessageBox "Click OK to revert text to style.",MB_OK,"Example Script"
    .RevertToStyle $LwpStyleTypeParagraph
End With
```

'Example: ReviseAcceptAll method

' This example accepts the revisions in the entire document for the specified ' author name.

' RUNTIME DEPENDENCIES: You must have a document open for this script to work.

.ReviseAcceptAll False, "Peyton McManus", ""

'Example: ReviseCancelAll method

' This example accepts the revisions in the entire document.

' RUNTIME DEPENDENCIES: You must have a document open for this script to work.

.ReviseCancelAll 0, "", ""

Word Pro: RelativeIndent class members

Properties

All

Application AS WPAApplication class

BodyOnly

Description

First

Hang

IsBothSidesEqual

IsValid

Name

Parent AS BaseObject class

Relative

Rest

Right

UseRelative

VersionID

Methods

RevertToStyle

Events

None

Word Pro: ReviewVersions class members

Properties

Application AS WPAApplication class

Description

FilesToCompare

IsValid

Name

NewFile

NewVersion

Parent AS BaseObject class

VersionID

Methods

ReviewVersions

Events

None

Word Pro: Revisiondisplay class members

Properties

Application AS WPAApplication class

DeleteFont AS Font class

Description

InsertFont AS Font class

IsValid

Name

Parent AS BaseObject class

RevMarkCharacter

RevMarkPosition

RevMarkType

TextAttributes AS Attributes class

UndoLevels AS StringCollection class

VersionID

Methods

None

Events

None

Word Pro: Revision class members

Properties

Application AS WPAApplication class

Description

IsValid

Name

Parent AS BaseObject class

RevisionType

VersionID

Methods

None

Events

None

Word Pro: RowContainer class members

Properties

[AbsoluteTextOrientation](#)
[Application](#) AS [WPAApplication class](#)
[ClientHeight](#)
[ClientWidth](#)
[ContentHeight](#)
[ContentName](#)
[ContentWidth](#)
[Description](#)
[DisplayablePageNum](#)
[DivisionInfo](#) AS [DivisionInfo class](#)
[DivisionName](#)
[Height](#)
[IsInOleDivision](#)
[IsValid](#)
[Layout](#) AS [Layout class](#)
[MaxContentHeight](#)
[MaxContentWidth](#)
[Name](#)
[NumContainers](#)
[PageNum](#)
[Parent](#) AS [BaseObject class](#)
[PositionXOnPage](#)
[PositionYOnPage](#)
[Presentation](#) AS [Presentation class](#)
[RelativePageNum](#)
[SectionName](#)
[TextOrientation](#)
[VersionID](#)
[Width](#)

Methods

[Abandon](#)
[AddContainer](#)
[Adopt](#)
[Anchor](#)
[Backward](#)
[ConnectContainer](#)
[DeleteContainer](#)
[Disconnect](#)
[DoesMultiCellPaste](#)
[EditWrap](#)
[Ending](#)
[Forward](#)
[GetObjectList](#)
[GetPasteFormatCategories](#)

GoToContainer

Hide

IsPointWithin

LinkContainers

ShowContainers

Start

UnlinkContainers

Events

None

Word Pro: RowGroupLayout class members

Properties

[AbsoluteOn](#)

[AbsoluteXPos](#)

[AbsoluteYPos](#)

[Accelerator](#)

[AllowResizeWhenCrop](#)

[AmtTether](#)

[AmtToRotateContent](#)

[AmtToTetherFrom](#)

[Application](#) AS [WPAApplication class](#)

[Background](#) AS [Background class](#)

[BaseLineOffset](#)

[BinName](#)

[BorderLines](#) AS [BorderLines class](#)

[BorderOffset](#)

[BottomExternalMargin](#)

[Center](#)

[ChildLayouts](#) AS [LayoutCollection class](#)

[ClassName](#)

[ColumnBalance](#)

[ColumnGap](#)

[ConditionType](#)

[Content](#)

[ContentName](#)

[ContentStyleName](#)

[Definition](#)

[Description](#)

[DirectionDown](#)

[DirectionLeft](#)

[DirectionRight](#)

[DirectionUp](#)

[DropCapPosition](#)

[EditorName](#)

[Footer](#) AS [Layout class](#)

[GridDistance](#)

[GridType](#)

[Gutter](#) AS [Gutter class](#)

[Header](#) AS [Layout class](#)

[Height](#)

[IsBreakable](#)

[IsChildSpannable](#)

[IsCollapsed](#)

[IsCollapsible](#)

[IsColumnBreakable](#)

[IsComplex](#)

IsConnected
IsErrorChecking
IsExpandDown
IsExpandLeft
IsExpandRight
IsExpandUp
IsIndexMarkAble
IsLocal
IsLocked
IsMarginSameAsParent
IsNotCopyable
IsNotGroupable
IsNoUICommAllowed
IsOverridden
IsOverride
IsPageBreak
IsPartOfGroup
IsPrintable
IsProtected
IsScrollable
IsSingleClickEntry
IsSizable
IsSnapTo
IsStyle
IsTableHeading
IsTOC
IsTOCMarkAble
IsValid
Join AS Join class
Justifiable
Layer AS Layout class
LayerName
LeaderDotType
LeftExternalMargin
LeftPage AS Layout class
LeftTopCellId
LineLocation
LinkFrame
MaintainAspectRatio
MarginBottom
MarginLeft
MarginRight
MarginTop
MasterName
MinBottomMargin
MinHeight
MinLeftMargin

[MinRightMargin](#)
[MinTopMargin](#)
[Name](#)
[NameBasedOnStyle](#)
[NumberOfLines](#)
[NumCols](#)
[NumColsSpannedOneCell](#)
[NumericFormat](#) AS [NumericFormat class](#)
[NumRowsSpannedOneCell](#)
[PageToUseLayoutOn](#)
[PaperName](#)
[Parent](#) AS [BaseObject class](#)
[RelativeType](#)
[RelativeXDistance](#)
[RelativeYDistance](#)
[RevisionType](#)
[RightExternalMargin](#)
[RightPage](#) AS [Layout class](#)
[ScaleHeight](#)
[ScaleMode](#)
[ScalePercentage](#)
[ScaleWidth](#)
[SelectType](#)
[Shadow](#) AS [Shadow class](#)
[Span](#)
[Style](#) AS [Layout class](#)
[StyleExceptions](#)
[TabRack](#) AS [TabRack class](#)
[TextOrient](#)
[Tile](#)
[TopExternalMargin](#)
[TopLeftCellRowId](#)
[UseFooter](#)
[UseHeader](#)
[UsePrinterSettings](#)
[UseWhen](#) AS [UseWhen class](#)
[VersionID](#)
[VertAlign](#)
[WasDeletedInRevMarkMode](#)
[WasInsertedInRevMarkMode](#)
[Where](#)
[Width](#)
[WPDataSets](#) AS [WPDataSetCollection class](#)
[WrapType](#)
[XOffset](#)
[XPosition](#)
[YOffset](#)

YPosition

Methods

AddChildToLayout

Backward

CreateLayer

DeleteContents

DeleteLayout

DoesMarkerNameMatch

FindClass

Forward

GetMarkerName

GetNamedProperty

GoToLayout

HasNamedProperty

ImportWatermarkGraphic

Mark

MirrorPage

MoveToBack

MoveToFront

RegisterWPDataSet

RemoveChildFromLayout

RemoveNamedProperty

RevisionAcceptLayoutChange

RevisionCancelLayoutChange

SetAllMargins

SetMinimumOrigin

SetNamedProperty

SetPaperNameAndUpdateSize

SetupForCropping

UnregisterWPDataSet

Update

Events

EnterLayout

KeyStroke

MouseDown

MouseUp

Word Pro: RowLayoutCollection class members

Properties

Application AS WPAApplication class

Count

Description

IsValid

Name

Parent AS BaseObject class

VersionID

Methods

IsEmpty

Item

Events

None

Word Pro: RowLayout class members

Properties

[AbsoluteOn](#)

[AbsoluteXPos](#)

[AbsoluteYPos](#)

[Accelerator](#)

[AllowResizeWhenCrop](#)

[AmtTether](#)

[AmtToRotateContent](#)

[AmtToTetherFrom](#)

[Application](#) AS [WPAApplication class](#)

[Background](#) AS [Background class](#)

[BaseLineOffset](#)

[BinName](#)

[BorderLines](#) AS [BorderLines class](#)

[BorderOffset](#)

[BottomExternalMargin](#)

[Center](#)

[ChildLayouts](#) AS [LayoutCollection class](#)

[ClassName](#)

[ColumnBalance](#)

[ColumnGap](#)

[ConditionType](#)

[Content](#)

[ContentName](#)

[ContentStyleName](#)

[Definition](#)

[Description](#)

[DirectionDown](#)

[DirectionLeft](#)

[DirectionRight](#)

[DirectionUp](#)

[DropCapPosition](#)

[EditorName](#)

[Footer](#) AS [Layout class](#)

[GridDistance](#)

[GridType](#)

[Gutter](#) AS [Gutter class](#)

[Header](#) AS [Layout class](#)

[Height](#)

[IsBreakable](#)

[IsChildSpannable](#)

[IsCollapsed](#)

[IsCollapsible](#)

[IsColumnBreakable](#)

[IsComplex](#)

IsConnected
IsErrorChecking
IsExpandDown
IsExpandLeft
IsExpandRight
IsExpandUp
IsIndexMarkAble
IsLocal
IsLocked
IsMarginSameAsParent
IsNotCopyable
IsNotGroupable
IsNoUICommAllowed
IsOverridden
IsOverride
IsPageBreak
IsPartOfGroup
IsPrintable
IsProtected
IsScrollable
IsSingleClickEntry
IsSizable
IsSnapTo
IsStyle
IsTableHeading
IsTOC
IsTOCMarkAble
IsValid
Join AS Join class
Justifiable
Layer AS Layout class
LayerName
LeaderDotType
LeftExternalMargin
LeftPage AS Layout class
LeftTopCellId
LineLocation
LinkFrame
MaintainAspectRatio
MarginBottom
MarginLeft
MarginRight
MarginTop
MasterName
MinBottomMargin
MinHeight
MinLeftMargin

[MinRightMargin](#)
[MinTopMargin](#)
[Name](#)
[NameBasedOnStyle](#)
[NumberOfLines](#)
[NumCols](#)
[NumColsSpannedOneCell](#)
[NumericFormat](#) AS [NumericFormat class](#)
[NumRowsSpannedOneCell](#)
[PageToUseLayoutOn](#)
[PaperName](#)
[Parent](#) AS [BaseObject class](#)
[RelativeType](#)
[RelativeXDistance](#)
[RelativeYDistance](#)
[RevisionType](#)
[RightExternalMargin](#)
[RightPage](#) AS [Layout class](#)
[ScaleHeight](#)
[ScaleMode](#)
[ScalePercentage](#)
[ScaleWidth](#)
[SelectType](#)
[Shadow](#) AS [Shadow class](#)
[Span](#)
[Style](#) AS [Layout class](#)
[StyleExceptions](#)
[TabRack](#) AS [TabRack class](#)
[TextOrient](#)
[Tile](#)
[TopExternalMargin](#)
[TopLeftCellRowId](#)
[UseFooter](#)
[UseHeader](#)
[UsePrinterSettings](#)
[UseWhen](#) AS [UseWhen class](#)
[VersionID](#)
[VertAlign](#)
[WasDeletedInRevMarkMode](#)
[WasInsertedInRevMarkMode](#)
[Where](#)
[Width](#)
[WPDataSets](#) AS [WPDataSetCollection class](#)
[WrapType](#)
[XOffset](#)
[XPosition](#)
[YOffset](#)

YPosition

Methods

AddChildToLayout

Backward

CreateLayer

DeleteContents

DeleteLayout

DoesMarkerNameMatch

FindClass

Forward

GetMarkerName

GetNamedProperty

GoToLayout

HasNamedProperty

ImportWatermarkGraphic

Mark

MirrorPage

MoveToBack

MoveToFront

RegisterWPDataSet

RemoveChildFromLayout

RemoveNamedProperty

RevisionAcceptLayoutChange

RevisionCancelLayoutChange

SetAllMargins

SetMinimumOrigin

SetNamedProperty

SetPaperNameAndUpdateSize

SetupForCropping

UnregisterWPDataSet

Update

Events

EnterLayout

KeyStroke

MouseDown

MouseUp

Word Pro: RubyContainer class members

Properties

[AbsoluteTextOrientation](#)
[Application](#) AS [WPAApplication class](#)
[ClientHeight](#)
[ClientWidth](#)
[ContentHeight](#)
[ContentName](#)
[ContentWidth](#)
[Description](#)
[DisplayablePageNum](#)
[DivisionInfo](#) AS [DivisionInfo class](#)
[DivisionName](#)
[Height](#)
[IsInOleDivision](#)
[IsValid](#)
[Layout](#) AS [Layout class](#)
[MaxContentHeight](#)
[MaxContentWidth](#)
[Name](#)
[NumContainers](#)
[PageNum](#)
[Parent](#) AS [BaseObject class](#)
[PositionXOnPage](#)
[PositionYOnPage](#)
[Presentation](#) AS [Presentation class](#)
[RelativePageNum](#)
[SectionName](#)
[TextOrientation](#)
[VersionID](#)
[Width](#)

Methods

[Abandon](#)
[AddContainer](#)
[Adopt](#)
[Anchor](#)
[Backward](#)
[ConnectContainer](#)
[DeleteContainer](#)
[Disconnect](#)
[DoesMultiCellPaste](#)
[EditWrap](#)
[Ending](#)
[Forward](#)
[GetObjectList](#)
[GetPasteFormatCategories](#)

GoToContainer

Hide

IsPointWithin

linkcontainers

RevertToStyle

SetStyle

ShowContainers

Start

unlinkcontainers

Events

None

Word Pro: RubyLayoutCollection class members

Properties

Application AS WPAApplication class

Count

Description

IsValid

Name

Parent AS BaseObject class

VersionID

Methods

IsEmpty

Item

Events

None

Word Pro: RubyLayout class members

Properties

[AbsoluteOn](#)

[AbsoluteXPos](#)

[AbsoluteYPos](#)

[Accelerator](#)

[AllowResizeWhenCrop](#)

[AmtTether](#)

[AmtToRotateContent](#)

[AmtToTetherFrom](#)

[Application](#) AS [WPAApplication class](#)

[Background](#) AS [Background class](#)

[BaseLineOffset](#)

[BinName](#)

[BorderLines](#) AS [BorderLines class](#)

[BorderOffset](#)

[BottomExternalMargin](#)

[Center](#)

[ChildLayouts](#) AS [LayoutCollection class](#)

[ClassName](#)

[ColumnBalance](#)

[ColumnGap](#)

[ConditionType](#)

[content](#)

[ContentName](#)

[ContentStyleName](#)

[Definition](#)

[Description](#)

[DirectionDown](#)

[DirectionLeft](#)

[DirectionRight](#)

[DirectionUp](#)

[DropCapPosition](#)

[EditorName](#)

[Footer](#) AS [Layout class](#)

[GridDistance](#)

[GridType](#)

[Gutter](#) AS [Gutter class](#)

[Header](#) AS [Layout class](#)

[Height](#)

[IsBreakable](#)

[IsChildSpannable](#)

[IsCollapsed](#)

[IsCollapsible](#)

[IsColumnBreakable](#)

[IsComplex](#)

IsConnected
IsErrorChecking
IsExpandDown
IsExpandLeft
IsExpandRight
IsExpandUp
IsIndexMarkAble
IsLocal
IsLocked
IsMarginSameAsParent
IsNotCopyable
IsNotGroupable
IsNoUICommAllowed
IsOverridden
IsOverride
IsPageBreak
IsPartOfGroup
IsPrintable
IsProtected
IsScrollable
IsSingleClickEntry
IsSizable
IsSnapTo
IsStyle
IsTableHeading
IsTOC
IsTOCMarkAble
IsValid
Join AS Join class
Justifiable
Layer AS Layout class
LayerName
LeaderDotType
LeftExternalMargin
LeftPage AS Layout class
LeftTopCellId
LineLocation
LinkFrame
MaintainAspectRatio
MarginBottom
MarginLeft
MarginRight
MarginTop
MasterName
MinBottomMargin
MinHeight
MinLeftMargin

[MinRightMargin](#)
[MinTopMargin](#)
[Name](#)
[NameBasedOnStyle](#)
[NumberOfLines](#)
[NumCols](#)
[NumColsSpannedOneCell](#)
[NumericFormat](#) AS [NumericFormat class](#)
[NumRowsSpannedOneCell](#)
[PageToUseLayoutOn](#)
[PaperName](#)
[Parent](#) AS [BaseObject class](#)
[Placement](#)
[RelativeType](#)
[RelativeXDistance](#)
[RelativeYDistance](#)
[RevisionType](#)
[RightExternalMargin](#)
[RightPage](#) AS [Layout class](#)
[ScaleHeight](#)
[ScaleMode](#)
[ScalePercentage](#)
[ScaleWidth](#)
[SelectType](#)
[Shadow](#) AS [Shadow class](#)
[Span](#)
[Style](#) AS [Layout class](#)
[StyleExceptions](#)
[TabRack](#) AS [TabRack class](#)
[TextOrient](#)
[Tile](#)
[TopExternalMargin](#)
[TopLeftCellRowId](#)
[UseFooter](#)
[UseHeader](#)
[UsePrinterSettings](#)
[UseWhen](#) AS [UseWhen class](#)
[VersionID](#)
[VertAlign](#)
[WasDeletedInRevMarkMode](#)
[WasInsertedInRevMarkMode](#)
[Where](#)
[Width](#)
[WPDataSets](#) AS [WPDataSetCollection class](#)
[WrapType](#)
[XOffset](#)
[XPosition](#)

YOffset

YPosition

Methods

AddChildToLayout

Backward

createlayer

DeleteContents

DeleteLayout

DoesMarkerNameMatch

FindClass

Forward

GetMarkerName

GetNamedProperty

GoToLayout

HasNamedProperty

ImportWatermarkGraphic

Mark

MirrorPage

MoveToBack

MoveToFront

registerwpdataset

RemoveChildFromLayout

RemoveNamedProperty

RevisionAcceptLayoutChange

RevisionCancelLayoutChange

setallmargins

SetMinimumOrigin

SetNamedProperty

SetPaperNameAndUpdateSize

SetupForCropping

UnregisterWPDataSet

Update

Events

EnterLayout

KeyStroke

MouseDown

MouseUp

Word Pro: RubyMarker class members

Properties

[Application](#) AS [WPAApplication class](#)
[Description](#)
[DivisionInfo](#) AS [DivisionInfo class](#)
[IsChanged](#)
[IsMarkerValid](#)
[IsRegistered](#)
[IsValid](#)
[Layout](#) AS [Layout class](#)
[MarkerClass](#)
[Name](#)
[NumCols](#)
[NumRows](#)
[PageNumber](#)
[PageNumberAsText](#)
[Parent](#) AS [BaseObject class](#)
[StartColumns](#)
[StartRow](#)
[StateID](#)
[VersionID](#)
[WPDataSets](#) AS [WPDataSetCollection class](#)

Methods

[DeleteContents](#)
[DeleteMarker](#)
[GetContents](#)
[GetMarkedText](#)
[GetNamedProperty](#)
[GetParagraphNumber](#)
[GetParagraphNumberString](#)
[GoTo](#)
[HasNamedProperty](#)
[InsertMarker](#)
[RegisterWPDataSet](#)
[RemoveNamedProperty](#)
[SetNamedProperty](#)
[UnregisterWPDataSet](#)

Events

None

Word Pro: Ruler class members

Properties

Application AS WPAApplication class

Description

IsShowing

IsValid

Name

Parent AS BaseObject class

SelectTab

VersionID

Methods

None

Events

None

Word Pro: ScriptDataSet class members

Properties

Application AS WPAApplication class

DataNames AS StringCollection class

Description

IsPersistent

IsValid

Name

Parent AS BaseObject class

VersionID

Methods

GetData

IsDataNameUsed

SetData

Events

None

Word Pro: SectionCollection class members

Properties

Application AS WPAApplication class

Count

Description

IsValid

Name

Parent AS BaseObject class

VersionID

Methods

IsEmpty

Item

Events

None

Word Pro: SectionTabs class members

Properties

Application AS WPAApplication class

Description

ExternalName

IsShowing

IsValid

Name

Parent AS BaseObject class

VersionID

Methods

Activate

AddNewSectionTabs

AddSectionTabs

ConnectSectionTabs

Contract

Expand

InsertSection

UpdateTabs

Events

None

Word Pro: Section class members

Properties

Application AS WPAApplication class

Color AS Color class

Description

IsIndex

IsValid

LayoutName

Name

NextSection

Parent AS BaseObject class

ShowTabs

UserName

VersionID

Methods

DeleteSection

GoToSection

Events

None

Word Pro: SetTabsDialog class members

Properties

Application AS WPAApplication class

Description

IsValid

Name

Parent AS BaseObject class

SelectTab

VersionID

Methods

None

Events

None

Word Pro: Shadow class members

Properties

Application AS WPAApplication class

Color AS Color class

Description

IsValid

Name

Parent AS BaseObject class

VersionID

XPosition

YPosition

Methods

None

Events

None

Word Pro: SilverBulletCollection class members

Properties

Application AS WPAApplication class

Count

Description

IsValid

Name

Parent AS BaseObject class

VersionID

Methods

IsEmpty

Item

Events

None

Word Pro: SilverBullet class members

Properties

Application AS WPAApplication class

Description

IsPrivate

IsValid

Name

Parent AS BaseObject class

RestartStyleName

Text AS Text class

VersionID

Methods

GetArrayProp

SetArrayProp

Events

None

Word Pro: SmartCorrectCollection class members

Properties

Application AS WPAApplication class

Count

Description

IsValid

Name

Parent AS BaseObject class

VersionID

Methods

IsEmpty

Item

Events

None

Word Pro: SmartCorrect class members

Properties

Application AS WPAApplication class

Description

DoInitialCaps

DoSentenceInitialCaps

DoSmartLinks

DoSmartQuotes

IsValid

Name

Parent AS BaseObject class

VersionID

Methods

AddSmartCorrect

DeleteSmartCorrect

Get

Events

None

Word Pro: SmartFillCollection class members

Properties

Application AS WPAApplication class

Count

Description

IsValid

Name

Parent AS BaseObject class

VersionID

Methods

IsEmpty

Item

Events

None

Word Pro: SmartFill class members

Properties

Application AS WPAApplication class

Description

IsValid

ListCount

Name

Parent AS BaseObject class

VersionID

Methods

AddStringToList

CreateEmptyList

GetListName

GetString

IsCaseExact

RemoveList

Events

None

Word Pro: SortKey class members

Properties

Application AS WPAApplication class

Description

FieldNumber

IsSortFromEnd

IsValid

Name

Parent AS BaseObject class

SortOrder

SortType

SortWord

SortWordOption

VersionID

Methods

None

Events

None

Word Pro: SortOptions class members

Properties

Application AS WPAApplication class

Description

FieldDelimiter

FieldDelimiterText

IsValid

Name

NumParagraphs

Parent AS BaseObject class

SortLevel1 AS SortKey class

SortLevel2 AS SortKey class

SortLevel3 AS SortKey class

SortNumbers

VersionID

Methods

None

Events

None

Word Pro: Spacing class members

Properties

Always

Amount

AmountOfSpaceAbove

AmountOfSpaceAboveLine

AmountOfSpaceBelow

Application AS WPAApplication class

Description

IsValid

Name

NumLinesOfSpace

NumLinesOfSpaceAbove

NumLinesOfSpaceAboveLine

NumLinesOfSpaceBelow

Parent AS BaseObject class

Type

TypeAbove

TypeAboveLine

TypeBelow

VersionID

Methods

RevertToStyle

Events

None

Word Pro: StatusBarButtonCollection class members

Properties

Application AS WPAApplication class

Count

Description

IsValid

Name

Parent AS BaseObject class

VersionID

Methods

IsEmpty

Item

Events

None

Word Pro: StatusBarButton class members

Properties

Application AS WPAApplication class

Description

GetButtonId

GetButtonType

IsValid

Name

Parent AS BaseObject class

VersionID

Methods

AddPopupGraphicItem

AddPopupPointSizeItem

AddPopupTextItem

DeleteButton

InvalidateButton

SetButtonPercent

SetOverrideGraphic

SetOverrideText

SetPopupAlignment

SetPopupIndex

SetPopupWidth

SetPopupWidthType

SimulateButtonClick

Events

StatusBarButtonClicked

StatusBarButtonFillPopupList

StatusBarButtonItemSelected

StatusBarButtonOverrideGraphic

StatusBarButtonOverrideText

StatusBarButtonOverrideTextAndGraphic

Word Pro: StatusBar class members

Properties

Active

Application AS WPAApplication class

Caption

Description

Height

IsValid

Left

Name

Parent AS BaseObject class

StatusBarButtons AS StatusBarButtonCollection class

Top

VersionID

Visible

Width

Methods

Close

CreateNewButton

GetStandardButtonId

HideStatusBar

InvalidateWholeBar

Move

Open

Resize

ShowStatusBar

Update

Events

Moved

StatusBarButtonClicked

StatusBarButtonFillPopupList

StatusBarButtonOverrideText

Word Pro: StringCollection class members

Properties

Application AS WPAApplication class

Count

Description

IsValid

Name

Parent AS BaseObject class

VersionID

Methods

IsEmpty

Item

Events

None

Word Pro: SubPageContainer class members

Properties

[AbsoluteTextOrientation](#)
[Application](#) AS [WPAApplication class](#)
[ClientHeight](#)
[ClientWidth](#)
[ContentHeight](#)
[ContentName](#)
[ContentWidth](#)
[Description](#)
[DisplayablePageNum](#)
[DivisionInfo](#) AS [DivisionInfo class](#)
[DivisionName](#)
[Height](#)
[IsFooter](#)
[IsHeader](#)
[IsInOleDivision](#)
[IsValid](#)
[Layout](#) AS [Layout class](#)
[MaxContentHeight](#)
[MaxContentWidth](#)
[Name](#)
[NumContainers](#)
[PageNum](#)
[Parent](#) AS [BaseObject class](#)
[PositionXOnPage](#)
[PositionYOnPage](#)
[Presentation](#) AS [Presentation class](#)
[RelativePageNum](#)
[SectionName](#)
[TextOrientation](#)
[VersionID](#)
[Width](#)

Methods

[Abandon](#)
[AddContainer](#)
[Adopt](#)
[Anchor](#)
[Backward](#)
[ConnectContainer](#)
[DeleteContainer](#)
[Disconnect](#)
[DoesMultiCellPaste](#)
[EditWrap](#)
[Ending](#)
[Forward](#)

GetObjectList

GetPasteFormatCategories

GoToContainer

Hide

IsPointWithin

linkcontainers

RevertToStyle

SetStyle

ShowContainers

Start

unlinkcontainers

Events

None

Word Pro: SuperPageContainer class members

Properties

[AbsoluteTextOrientation](#)
[Application](#) AS [WPAApplication class](#)
[ClientHeight](#)
[ClientWidth](#)
[ContentHeight](#)
[ContentName](#)
[ContentWidth](#)
[Description](#)
[DisplayablePageNum](#)
[DivisionInfo](#) AS [DivisionInfo class](#)
[DivisionName](#)
[Height](#)
[IsFooter](#)
[IsHeader](#)
[IsInOleDivision](#)
[IsValid](#)
[Layout](#) AS [Layout class](#)
[MaxContentHeight](#)
[MaxContentWidth](#)
[Name](#)
[NumContainers](#)
[PageNum](#)
[Parent](#) AS [BaseObject class](#)
[PositionXOnPage](#)
[PositionYOnPage](#)
[Presentation](#) AS [Presentation class](#)
[RelativePageNum](#)
[SectionName](#)
[TextOrientation](#)
[VersionID](#)
[Width](#)

Methods

[Abandon](#)
[AddContainer](#)
[Adopt](#)
[Anchor](#)
[Backward](#)
[ConnectContainer](#)
[DeleteContainer](#)
[Disconnect](#)
[DoesMultiCellPaste](#)
[EditWrap](#)
[Ending](#)
[Forward](#)

GetObjectList

GetPasteFormatCategories

GoToContainer

Hide

IsPointWithin

linkcontainers

RevertToStyle

SetStyle

ShowContainers

Start

unlinkcontainers

Events

None

Word Pro: SuperTableCollection class members

Properties

Application AS WPAApplication class

Count

Description

IsValid

Name

Parent AS BaseObject class

VersionID

Methods

IsEmpty

Item

Events

None

Word Pro: SuperTableContainer class members

Properties

[AbsoluteTextOrientation](#)
[Application](#) AS [WPAApplication class](#)
[ClientHeight](#)
[ClientWidth](#)
[ContentHeight](#)
[ContentName](#)
[ContentWidth](#)
[Description](#)
[DisplayablePageNum](#)
[DivisionInfo](#) AS [DivisionInfo class](#)
[DivisionName](#)
[Height](#)
[IsInOleDivision](#)
[IsValid](#)
[Layout](#) AS [Layout class](#)
[MaxContentHeight](#)
[MaxContentWidth](#)
[Name](#)
[NumContainers](#)
[PageNum](#)
[Parent](#) AS [BaseObject class](#)
[PositionXOnPage](#)
[PositionYOnPage](#)
[Presentation](#) AS [Presentation class](#)
[RelativePageNum](#)
[SectionName](#)
[TextOrientation](#)
[VersionID](#)
[Width](#)

Methods

[Abandon](#)
[AddContainer](#)
[Adopt](#)
[Anchor](#)
[Backward](#)
[ConnectContainer](#)
[DeleteContainer](#)
[Disconnect](#)
[DoesMultiCellPaste](#)
[EditWrap](#)
[Ending](#)
[Forward](#)
[GetObjectList](#)
[GetPasteFormatCategories](#)

GoToContainer

Hide

IsPointWithin

linkcontainers

RevertToStyle

SetStyle

ShowContainers

Start

unlinkcontainers

Events

None

Word Pro: SuperTableGroupLayout class members

Properties

[AbsoluteOn](#)
[AbsoluteXPos](#)
[AbsoluteYPos](#)
[Accelerator](#)
[AllowResizeWhenCrop](#)
[AmtTether](#)
[AmtToRotateContent](#)
[AmtToTetherFrom](#)
[Application](#) AS [WPAApplication class](#)
[Background](#) AS [Background class](#)
[BaseLineOffset](#)
[BinName](#)
[BorderLines](#) AS [BorderLines class](#)
[BorderOffset](#)
[BottomExternalMargin](#)
[Center](#)
[ChildLayouts](#) AS [LayoutCollection class](#)
[ClassName](#)
[ColumnBalance](#)
[ColumnGap](#)
[ConditionType](#)
[Content](#)
[ContentName](#)
[ContentStyleName](#)
[Definition](#)
[Description](#)
[DirectionDown](#)
[DirectionLeft](#)
[DirectionRight](#)
[DirectionUp](#)
[DropCapPosition](#)
[EditorName](#)
[Footer](#) AS [Layout class](#)
[GridDistance](#)
[GridType](#)
[Gutter](#) AS [Gutter class](#)
[Header](#) AS [Layout class](#)
[Height](#)
[IsBreakable](#)
[IsChildSpannable](#)
[IsCollapsed](#)
[IsCollapsible](#)
[IsColumnBreakable](#)
[IsComplex](#)

IsConnected
IsErrorChecking
IsExpandDown
IsExpandLeft
IsExpandRight
IsExpandUp
IsIndexMarkAble
IsLocal
IsLocked
IsMarginSameAsParent
IsNotCopyable
IsNotGroupable
IsNoUICommAllowed
IsOverridden
IsOverride
IsPageBreak
IsPartOfGroup
IsPrintable
IsProtected
IsScrollable
IsSingleClickEntry
IsSizable
IsSnapTo
IsStyle
IsTableHeading
IsTOC
IsTOCMarkAble
IsValid
Join AS Join class
Justifiable
Layer AS Layout class
LayerName
LeaderDotType
LeftExternalMargin
LeftPage AS Layout class
LeftTopCellId
LineLocation
LinkFrame
MaintainAspectRatio
MarginBottom
MarginLeft
MarginRight
MarginTop
MasterName
MinBottomMargin
MinHeight
MinLeftMargin

[MinRightMargin](#)
[MinTopMargin](#)
[Name](#)
[NameBasedOnStyle](#)
[NumberOfLines](#)
[NumCols](#)
[NumColsSpannedOneCell](#)
[NumericFormat](#) AS [NumericFormat class](#)
[NumRowsSpannedOneCell](#)
[PageToUseLayoutOn](#)
[PaperName](#)
[Parent](#) AS [BaseObject class](#)
[RelativeType](#)
[RelativeXDistance](#)
[RelativeYDistance](#)
[RevisionType](#)
[RightExternalMargin](#)
[RightPage](#) AS [Layout class](#)
[ScaleHeight](#)
[ScaleMode](#)
[ScalePercentage](#)
[ScaleWidth](#)
[SelectType](#)
[Shadow](#) AS [Shadow class](#)
[Span](#)
[Style](#) AS [Layout class](#)
[StyleExceptions](#)
[TabRack](#) AS [TabRack class](#)
[TextOrient](#)
[Tile](#)
[TopExternalMargin](#)
[TopLeftCellRowId](#)
[UseFooter](#)
[UseHeader](#)
[UsePrinterSettings](#)
[UseWhen](#) AS [UseWhen class](#)
[VersionID](#)
[VertAlign](#)
[WasDeletedInRevMarkMode](#)
[WasInsertedInRevMarkMode](#)
[Where](#)
[Width](#)
[WPDataSets](#) AS [WPDataSetCollection class](#)
[WrapType](#)
[XOffset](#)
[XPosition](#)
[YOffset](#)

YPosition

Methods

AddChildToLayout

Backward

CreateLayer

DeleteContents

DeleteLayout

DoesMarkerNameMatch

FindClass

Forward

GetMarkerName

GetNamedProperty

GoToLayout

HasNamedProperty

ImportWatermarkGraphic

Mark

MirrorPage

MoveToBack

MoveToFront

RegisterWPDataSet

RemoveChildFromLayout

RemoveNamedProperty

RevisionAcceptLayoutChange

RevisionCancelLayoutChange

SetAllMargins

SetMinimumOrigin

SetNamedProperty

SetPaperNameAndUpdateSize

SetupForCropping

UnregisterWPDataSet

Update

Events

EnterLayout

KeyStroke

MouseDown

MouseUp

Word Pro: SuperTableLayoutCollection class members

Properties

Application AS WPAApplication class

Count

Description

IsValid

Name

Parent AS BaseObject class

VersionID

Methods

IsEmpty

Item

Events

None

Word Pro: SuperTableLayout class members

Properties

[AbsoluteOn](#)
[AbsoluteXPos](#)
[AbsoluteYPos](#)
[Accelerator](#)
[AllowResizeWhenCrop](#)
[AmtTether](#)
[AmtToRotateContent](#)
[AmtToTetherFrom](#)
[Application](#) AS [WPAApplication class](#)
[Background](#) AS [Background class](#)
[BaseLineOffset](#)
[BinName](#)
[BorderLines](#) AS [BorderLines class](#)
[BorderOffset](#)
[BottomExternalMargin](#)
[Center](#)
[ChildLayouts](#) AS [LayoutCollection class](#)
[ClassName](#)
[ColumnBalance](#)
[ColumnGap](#)
[ConditionType](#)
[Content](#)
[ContentName](#)
[ContentStyleName](#)
[Definition](#)
[Description](#)
[DirectionDown](#)
[DirectionLeft](#)
[DirectionRight](#)
[DirectionUp](#)
[DropCapPosition](#)
[EditorName](#)
[Footer](#) AS [Layout class](#)
[GridDistance](#)
[GridType](#)
[Gutter](#) AS [Gutter class](#)
[Header](#) AS [Layout class](#)
[Height](#)
[IsBreakable](#)
[IsChildSpannable](#)
[IsCollapsed](#)
[IsCollapsible](#)
[IsColumnBreakable](#)
[IsComplex](#)

IsConnected
IsErrorChecking
IsExpandDown
IsExpandLeft
IsExpandRight
IsExpandUp
IsIndexMarkAble
IsLocal
IsLocked
IsMarginSameAsParent
IsNotCopyable
IsNotGroupable
IsNoUICommAllowed
IsOverridden
IsOverride
IsPageBreak
IsPartOfGroup
IsPrintable
IsProtected
IsScrollable
IsSingleClickEntry
IsSizable
IsSnapTo
IsStyle
IsTableHeading
IsTOC
IsTOCMarkAble
IsValid
Join AS Join class
Justifiable
Layer AS Layout class
LayerName
LeaderDotType
LeftExternalMargin
LeftPage AS Layout class
LeftTopCellId
LineLocation
LinkFrame
MaintainAspectRatio
MarginBottom
MarginLeft
MarginRight
MarginTop
MasterName
MinBottomMargin
MinHeight
MinLeftMargin

[MinRightMargin](#)
[MinTopMargin](#)
[Name](#)
[NameBasedOnStyle](#)
[NumberOfLines](#)
[NumCols](#)
[NumColsSpannedOneCell](#)
[NumericFormat](#) AS [NumericFormat class](#)
[NumRowsSpannedOneCell](#)
[PageToUseLayoutOn](#)
[PaperName](#)
[Parent](#) AS [BaseObject class](#)
[RelativeType](#)
[RelativeXDistance](#)
[RelativeYDistance](#)
[RevisionType](#)
[RightExternalMargin](#)
[RightPage](#) AS [Layout class](#)
[ScaleHeight](#)
[ScaleMode](#)
[ScalePercentage](#)
[ScaleWidth](#)
[SelectType](#)
[Shadow](#) AS [Shadow class](#)
[Span](#)
[Style](#) AS [Layout class](#)
[StyleExceptions](#)
[TabRack](#) AS [TabRack class](#)
[TextOrient](#)
[Tile](#)
[TopExternalMargin](#)
[TopLeftCellRowId](#)
[UseFooter](#)
[UseHeader](#)
[UsePrinterSettings](#)
[UseWhen](#) AS [UseWhen class](#)
[VersionID](#)
[VertAlign](#)
[WasDeletedInRevMarkMode](#)
[WasInsertedInRevMarkMode](#)
[Where](#)
[Width](#)
[WPDataSets](#) AS [WPDataSetCollection class](#)
[WrapType](#)
[XOffset](#)
[XPosition](#)
[YOffset](#)

YPosition

Methods

AddChildToLayout

Backward

CreateLayer

DeleteContents

DeleteLayout

DoesMarkerNameMatch

FindClass

Forward

GetMarkerName

GetNamedProperty

GoToLayout

HasNamedProperty

ImportWatermarkGraphic

Mark

MirrorPage

MoveToBack

MoveToFront

RegisterWPDataSet

RemoveChildFromLayout

RemoveNamedProperty

RevisionAcceptLayoutChange

RevisionCancelLayoutChange

SetAllMargins

SetMinimumOrigin

SetNamedProperty

SetPaperNameAndUpdateSize

SetupForCropping

UnregisterWPDataSet

Update

Events

EnterLayout

KeyStroke

MouseDown

MouseUp

Word Pro: SuperTable class members

Properties

Application AS WPAApplication class

CanEmbed

ContentType

Description

IsEmpty

IsReplaceable

IsValid

Name

Parent AS BaseObject class

VersionID

Methods

None

Events

None

Word Pro: TableCollection class members

Properties

Application AS WPAApplication class

Count

Description

IsValid

Name

Parent AS BaseObject class

VersionID

Methods

IsEmpty

Item

Events

None

Word Pro: TableContainer class members

Properties

[AbsoluteTextOrientation](#)
[Application](#) AS [WPAApplication class](#)
[ClientHeight](#)
[ClientWidth](#)
[ContentHeight](#)
[ContentName](#)
[ContentWidth](#)
[Description](#)
[DisplayablePageNum](#)
[DivisionInfo](#) AS [DivisionInfo class](#)
[DivisionName](#)
[Height](#)
[IsInOleDivision](#)
[IsValid](#)
[Layout](#) AS [Layout class](#)
[MaxContentHeight](#)
[MaxContentWidth](#)
[Name](#)
[NumContainers](#)
[PageNum](#)
[Parent](#) AS [BaseObject class](#)
[PositionXOnPage](#)
[PositionYOnPage](#)
[Presentation](#) AS [Presentation class](#)
[RelativePageNum](#)
[SectionName](#)
[TextOrientation](#)
[VersionID](#)
[Width](#)

Methods

[Abandon](#)
[AddContainer](#)
[Adopt](#)
[Anchor](#)
[Backward](#)
[ConnectContainer](#)
[DeleteContainer](#)
[Disconnect](#)
[DoesMultiCellPaste](#)
[EditWrap](#)
[Ending](#)
[Forward](#)
[GetObjectList](#)
[GetPasteFormatCategories](#)

GoToContainer

Hide

IsPointWithin

LinkContainers

ShowContainers

Start

UnlinkContainers

Events

None

Word Pro: TableFill class members

Properties

Application AS WPAApplication class

Background AS Background class

Description

FillStyle

IsValid

Name

Parent AS BaseObject class

VersionID

Methods

SetPattern

Events

None

Word Pro: TableHeadingCollection class members

Properties

Application AS WPAApplication class

Count

Description

IsValid

Name

Parent AS BaseObject class

VersionID

Methods

IsEmpty

Item

Events

None

Word Pro: TableHeadingLayoutCollection class members

Properties

Application AS WPAApplication class

Count

Description

IsValid

Name

Parent AS BaseObject class

VersionID

Methods

IsEmpty

Item

Events

None

Word Pro: TableHeadingLayout class members

Properties

[AbsoluteOn](#)

[AbsoluteXPos](#)

[AbsoluteYPos](#)

[Accelerator](#)

[AllowResizeWhenCrop](#)

[AmtTether](#)

[AmtToRotateContent](#)

[AmtToTetherFrom](#)

[Application](#) AS [WPAApplication class](#)

[Background](#) AS [Background class](#)

[BaseLineOffset](#)

[BinName](#)

[BorderLines](#) AS [BorderLines class](#)

[BorderOffset](#)

[BottomExternalMargin](#)

[Center](#)

[ChildLayouts](#) AS [LayoutCollection class](#)

[ClassName](#)

[ColumnBalance](#)

[ColumnGap](#)

[ConditionType](#)

[Content](#)

[ContentName](#)

[ContentStyleName](#)

[Definition](#)

[Description](#)

[DirectionDown](#)

[DirectionLeft](#)

[DirectionRight](#)

[DirectionUp](#)

[DropCapPosition](#)

[EditorName](#)

[Footer](#) AS [Layout class](#)

[GridDistance](#)

[GridType](#)

[Gutter](#) AS [Gutter class](#)

[Header](#) AS [Layout class](#)

[Height](#)

[IsBreakable](#)

[IsChildSpannable](#)

[IsCollapsed](#)

[IsCollapsible](#)

[IsColumnBreakable](#)

[IsComplex](#)

IsConnected
IsErrorChecking
IsExpandDown
IsExpandLeft
IsExpandRight
IsExpandUp
IsIndexMarkAble
IsLocal
IsLocked
IsMarginSameAsParent
IsNotCopyable
IsNotGroupable
IsNoUICommAllowed
IsOverridden
IsOverride
IsPageBreak
IsPartOfGroup
IsPrintable
IsProtected
IsScrollable
IsSingleClickEntry
IsSizable
IsSnapTo
IsStyle
IsTableHeading
IsTOC
IsTOCMarkAble
IsValid
Join AS Join class
Justifiable
Layer AS Layout class
LayerName
LeaderDotType
LeftExternalMargin
LeftPage AS Layout class
LeftTopCellId
LineLocation
LinkFrame
MaintainAspectRatio
MarginBottom
MarginLeft
MarginRight
MarginTop
MasterName
MinBottomMargin
MinHeight
MinLeftMargin

[MinRightMargin](#)
[MinTopMargin](#)
[Name](#)
[NameBasedOnStyle](#)
[NumberOfLines](#)
[NumCols](#)
[NumColsSpannedOneCell](#)
[NumericFormat](#) AS [NumericFormat class](#)
[NumRowsSpannedOneCell](#)
[PageToUseLayoutOn](#)
[PaperName](#)
[Parent](#) AS [BaseObject class](#)
[RelativeType](#)
[RelativeXDistance](#)
[RelativeYDistance](#)
[RevisionType](#)
[RightExternalMargin](#)
[RightPage](#) AS [Layout class](#)
[ScaleHeight](#)
[ScaleMode](#)
[ScalePercentage](#)
[ScaleWidth](#)
[SelectType](#)
[Shadow](#) AS [Shadow class](#)
[Span](#)
[Style](#) AS [Layout class](#)
[StyleExceptions](#)
[TabRack](#) AS [TabRack class](#)
[TextOrient](#)
[Tile](#)
[TopExternalMargin](#)
[TopLeftCellRowId](#)
[UseFooter](#)
[UseHeader](#)
[UsePrinterSettings](#)
[UseWhen](#) AS [UseWhen class](#)
[VersionID](#)
[VertAlign](#)
[WasDeletedInRevMarkMode](#)
[WasInsertedInRevMarkMode](#)
[Where](#)
[Width](#)
[WPDataSets](#) AS [WPDataSetCollection class](#)
[WrapType](#)
[XOffset](#)
[XPosition](#)
[YOffset](#)

YPosition

Methods

AddChildToLayout

Backward

CreateLayer

DeleteContents

DeleteLayout

DoesMarkerNameMatch

FindClass

Forward

GetMarkerName

GetNamedProperty

GoToLayout

HasNamedProperty

ImportWatermarkGraphic

Mark

MirrorPage

MoveToBack

MoveToFront

RegisterWPDataSet

RemoveChildFromLayout

RemoveNamedProperty

RevisionAcceptLayoutChange

RevisionCancelLayoutChange

SetAllMargins

SetMinimumOrigin

SetNamedProperty

SetPaperNameAndUpdateSize

SetupForCropping

UnregisterWPDataSet

Update

Events

EnterLayout

KeyStroke

MouseDown

MouseUp

Word Pro: TableHeading class members

Properties

Application AS WPAApplication class

CanEmbed

CellLayouts AS StringCollection class

ColumnLayouts AS StringCollection class

ContentType

CurrentCell AS CellLayout class

CurrentColumn AS Layout class

CurrentRow AS RowLayout class

DefCellStyleName

DefColWidth

DefRowHeight

Description

EndingColOfSelection

EndingRowOfSelection

IsAutoGrow

IsEmpty

IsParagraphNumberingDown

IsReplaceable

IsResetParagraphNumber

IsSizingViaMouse

IsValid

Layout AS Layout class

MaxBottomBorder

MaxBottomGutter

MaxLeftBorder

MaxLeftGutter

MaxNumColsAllowed

MaxNumRowsAllowed

MaxRightBorder

MaxRightGutter

MaxSplitCols

MaxSplitRows

MaxTopBorder

MaxTopGutter

Name

NumCols

NumRows

Parent AS BaseObject class

RowLayouts AS StringCollection class

SelectionType

SingleCellSelected

StartingColOfSelection

StartingColStringOfSelection

StartingRowOfSelection

TableFill AS TableFill class

TableLine AS TableLine class

VersionID

Methods

CellLayout

Connect

Copy

DeleteTable

DisconnectCells

DoesMarkerNameMatch

FindCellLayout

GetMarkerName

GoToTableCell

InsertRowOrColumn

Mark

SelectTableItem

Split

Events

None

Word Pro: TableLayoutCollection class members

Properties

Application AS WPAApplication class

Count

Description

IsValid

Name

Parent AS BaseObject class

VersionID

Methods

IsEmpty

Item

Events

None

Word Pro: TableLayout class members

Properties

[AbsoluteOn](#)

[AbsoluteXPos](#)

[AbsoluteYPos](#)

[Accelerator](#)

[AllowResizeWhenCrop](#)

[AmtTether](#)

[AmtToRotateContent](#)

[AmtToTetherFrom](#)

[Application](#) AS [WPAApplication class](#)

[Background](#) AS [Background class](#)

[BaseLineOffset](#)

[BinName](#)

[BorderLines](#) AS [BorderLines class](#)

[BorderOffset](#)

[BottomExternalMargin](#)

[Center](#)

[ChildLayouts](#) AS [LayoutCollection class](#)

[ClassName](#)

[ColumnBalance](#)

[ColumnGap](#)

[ConditionType](#)

[Content](#)

[ContentName](#)

[ContentStyleName](#)

[Definition](#)

[Description](#)

[DirectionDown](#)

[DirectionLeft](#)

[DirectionRight](#)

[DirectionUp](#)

[DropCapPosition](#)

[EditorName](#)

[Footer](#) AS [Layout class](#)

[GridDistance](#)

[GridType](#)

[Gutter](#) AS [Gutter class](#)

[Header](#) AS [Layout class](#)

[Height](#)

[IsBreakable](#)

[IsChildSpannable](#)

[IsCollapsed](#)

[IsCollapsible](#)

[IsColumnBreakable](#)

[IsComplex](#)

IsConnected
IsErrorChecking
IsExpandDown
IsExpandLeft
IsExpandRight
IsExpandUp
IsIndexMarkAble
IsLocal
IsLocked
IsMarginSameAsParent
IsNotCopyable
IsNotGroupable
IsNoUICommAllowed
IsOverridden
IsOverride
IsPageBreak
IsPartOfGroup
IsPrintable
IsProtected
IsScrollable
IsSingleClickEntry
IsSizable
IsSnapTo
IsStyle
IsTableHeading
IsTOC
IsTOCMarkAble
IsValid
Join AS Join class
Justifiable
Layer AS Layout class
LayerName
LeaderDotType
LeftExternalMargin
LeftPage AS Layout class
LeftTopCellId
LineLocation
LinkFrame
MaintainAspectRatio
MarginBottom
MarginLeft
MarginRight
MarginTop
MasterName
MinBottomMargin
MinHeight
MinLeftMargin

[MinRightMargin](#)
[MinTopMargin](#)
[Name](#)
[NameBasedOnStyle](#)
[NumberOfLines](#)
[NumCols](#)
[NumColsSpannedOneCell](#)
[NumericFormat](#) AS [NumericFormat class](#)
[NumRowsSpannedOneCell](#)
[PageToUseLayoutOn](#)
[PaperName](#)
[Parent](#) AS [BaseObject class](#)
[RelativeType](#)
[RelativeXDistance](#)
[RelativeYDistance](#)
[RevisionType](#)
[RightExternalMargin](#)
[RightPage](#) AS [Layout class](#)
[ScaleHeight](#)
[ScaleMode](#)
[ScalePercentage](#)
[ScaleWidth](#)
[SelectType](#)
[Shadow](#) AS [Shadow class](#)
[Span](#)
[Style](#) AS [Layout class](#)
[StyleExceptions](#)
[TabRack](#) AS [TabRack class](#)
[TextOrient](#)
[Tile](#)
[TopExternalMargin](#)
[TopLeftCellRowId](#)
[UseFooter](#)
[UseHeader](#)
[UsePrinterSettings](#)
[UseWhen](#) AS [UseWhen class](#)
[VersionID](#)
[VertAlign](#)
[WasDeletedInRevMarkMode](#)
[WasInsertedInRevMarkMode](#)
[Where](#)
[Width](#)
[WPDataSets](#) AS [WPDataSetCollection class](#)
[WrapType](#)
[XOffset](#)
[XPosition](#)
[YOffset](#)

YPosition

Methods

AddChildToLayout

Backward

CreateLayer

DeleteContents

DeleteLayout

DoesMarkerNameMatch

FindClass

Forward

GetMarkerName

GetNamedProperty

GoToLayout

HasNamedProperty

ImportWatermarkGraphic

Mark

MirrorPage

MoveToBack

MoveToFront

RegisterWPDataSet

RemoveChildFromLayout

RemoveNamedProperty

RevisionAcceptLayoutChange

RevisionCancelLayoutChange

SetAllMargins

SetMinimumOrigin

SetNamedProperty

SetPaperNameAndUpdateSize

SetupForCropping

UnregisterWPDataSet

Update

Events

EnterLayout

KeyStroke

MouseDown

MouseUp

Word Pro: TableLine class members

Properties

[Application](#) AS [WPAApplication class](#)

[BorderLines](#) AS [BorderLines class](#)

[Description](#)

[DiagonalLines](#) AS [BorderLines class](#)

[IsValid](#)

[LineMix](#)

[Name](#)

[OutlineBorderLines](#) AS [BorderLines class](#)

[Parent](#) AS [BaseObject class](#)

[VersionID](#)

Methods

[BeginCustomLines](#)

[ChgLineStyle](#)

[EndCustomLines](#)

[GetLineMix](#)

[GetLineStyle](#)

[SetLineOneSide](#)

[SetLinesAllSides](#)

Events

None

Word Pro: TableMarkerCollection class members

Properties

Application AS WPAApplication class

Count

Description

IsValid

Name

Parent AS BaseObject class

VersionID

Methods

IsEmpty

Item

Events

None

Word Pro: TableMarker class members

Properties

[Application](#) AS [WPAApplication class](#)
[Description](#)
[DivisionInfo](#) AS [DivisionInfo class](#)
[IsChanged](#)
[IsMarkerValid](#)
[IsRegistered](#)
[IsValid](#)
[Layout](#) AS [Layout class](#)
[MarkerClass](#)
[Name](#)
[NumCols](#)
[NumRows](#)
[PageNumber](#)
[PageNumberAsText](#)
[Parent](#) AS [BaseObject class](#)
[StartColumns](#)
[StartRow](#)
[StateID](#)
[VersionID](#)
[WPDataSets](#) AS [WPDataSetCollection class](#)

Methods

[DeleteContents](#)
[DeleteMarker](#)
[GetContents](#)
[GetMarkedText](#)
[GetNamedProperty](#)
[GetParagraphNumber](#)
[GetParagraphNumberString](#)
[GoTo](#)
[HasNamedProperty](#)
[InsertMarker](#)
[RegisterWPDataSet](#)
[RemoveNamedProperty](#)
[SetNamedProperty](#)
[UnregisterWPDataSet](#)

Events

None

Word Pro: TableOnlyCollection class members

Properties

Application AS WPAApplication class

Count

Description

IsValid

Name

Parent AS BaseObject class

VersionID

Methods

IsEmpty

Item

Events

None

Word Pro: TableOnlyCont class members

Properties

[AbsoluteTextOrientation](#)
[Application](#) AS [WPAApplication class](#)
[ClientHeight](#)
[ClientWidth](#)
[ContentHeight](#)
[ContentName](#)
[ContentWidth](#)
[Description](#)
[DisplayablePageNum](#)
[DivisionInfo](#) AS [DivisionInfo class](#)
[DivisionName](#)
[Height](#)
[IsInOleDivision](#)
[IsValid](#)
[Layout](#) AS [Layout class](#)
[MaxContentHeight](#)
[MaxContentWidth](#)
[Name](#)
[NumContainers](#)
[PageNum](#)
[Parent](#) AS [BaseObject class](#)
[PositionXOnPage](#)
[PositionYOnPage](#)
[Presentation](#) AS [Presentation class](#)
[RelativePageNum](#)
[SectionName](#)
[TextOrientation](#)
[VersionID](#)
[Width](#)

Methods

[Abandon](#)
[AddContainer](#)
[Adopt](#)
[Anchor](#)
[Backward](#)
[ConnectContainer](#)
[DeleteContainer](#)
[Disconnect](#)
[DoesMultiCellPaste](#)
[EditWrap](#)
[Ending](#)
[Forward](#)
[GetObjectList](#)
[GetPasteFormatCategories](#)

GoToContainer

Hide

IsPointWithin

LinkContainers

ShowContainers

Start

UnlinkContainers

Events

None

Word Pro: Table class members

Properties

Application AS WPAApplication class

CanEmbed

CellEngine AS CellEngine class

CellLayouts AS StringCollection class

ColumnLayouts AS StringCollection class

ContentType

CurrentCell AS CellLayout class

CurrentColumn AS Layout class

CurrentRow AS RowLayout class

DefCellStyleName

DefColWidth

DefRowHeight

Description

EndingColOfSelection

EndingRowOfSelection

IsAutoGrow

IsEmpty

IsParagraphNumberingDown

IsReplaceable

IsResetParagraphNumber

IsSizingViaMouse

IsValid

Layout AS Layout class

MaxBottomBorder

MaxBottomGutter

MaxLeftBorder

MaxLeftGutter

MaxNumColsAllowed

MaxNumRowsAllowed

MaxRightBorder

MaxRightGutter

MaxSplitCols

MaxSplitRows

MaxTopBorder

MaxTopGutter

Name

NumCols

NumRows

Parent AS BaseObject class

RowLayouts AS StringCollection class

SelectionType

SingleCellSelected

StartingColOfSelection

StartingColStringOfSelection

StartingRowOfSelection

TableFill AS TableFill class

TableLine AS TableLine class

VersionID

Methods

CellLayout

Connect

Copy

DeleteTable

DisconnectCells

DoesMarkerNameMatch

FindCellLayout

GetMarkerName

GoToTableCell

InsertRowOrColumn

Mark

SelectTableItem

Split

Sum

Events

None

Word Pro: TabRack class members

Properties

Application AS WPAApplication class

Description

IsValid

Name

NumTabs

Parent AS BaseObject class

VersionID

Methods

ClearAll

DeleteTab

GetTabAlignChar

GetTabLeaderType

GetTabPosition

GetTabRelativeType

GetTabType

InsertOne

RevertToStyle

Events

None

Word Pro: TextCollection class members

Properties

Application AS WPAApplication class

Count

Description

IsValid

Name

Parent AS BaseObject class

VersionID

Methods

IsEmpty

Item

Events

None

Word Pro: TextDocument class members

Properties

[Application](#) AS [WPAApplication class](#)
[AutoRunMacro](#) AS [AutoRunMacro class](#)
[BaseURL](#)
[BookmarkManager](#) AS [BookmarkManager class](#)
[CanCreatePreviewBitmap](#)
[Changed](#)
[ContextLocked](#)
[CreatePreviewBitmap](#)
[DdeLinkManager](#) AS [DdeLinkManager class](#)
[DdeOutboundInfo](#)
[DefaultFileType](#)
[DemandLoad](#)
[Description](#)
[DivisionInfo](#) AS [DivisionInfo class](#)
[DivisionNames](#) AS [StringCollection class](#)
[DivisionOptions](#) AS [DivisionOptions class](#)
[Divisions](#) AS [DivisionCollection class](#)
[DocControl](#) AS [DocControl class](#)
[DocInfo](#) AS [DocInfo class](#)
[DocOptions](#) AS [Options class](#)
[EditorManager](#) AS [EditorManager class](#)
[EditorName](#)
[Embedded](#)
[Epoch](#)
[FileType](#)
[FinishedSpellChecking](#)
[FirstChild](#)
[FirstCursorableDivision](#)
[FirstDivision](#)
[FirstName](#)
[FirstPage](#)
[FootnoteOptions](#) AS [FootnoteOptions class](#)
[FrameCaptionOptions](#) AS [FrameCaptionOptions class](#)
[FullName](#)
[HasIndex](#)
[HasToc](#)
[IsAnnotateOnly](#)
[IsChanged](#)
[IsChangedOtherThanLinkTo](#)
[IsChangedSinceTimeSave](#)
[IsChangedToLinkTo](#)
[IsDocLoading](#)
[IsEmptyDoc](#)
[IsEndnoteDivision](#)

IsFrameCaptionOptions
IsLockedForRevisions
IsOpen
IsProtected
IsSpellBarUp
IsStyleSheet
IsUndoOn
IsValid
LastChild
LastCursorableDivision
LastDivision
LastName
LastPage
LineNumberOptions AS LineNumberOptions class
Location
Locked
LockLevel
MailRouting AS MailRouting class
Master AS TextDocument class
MergeOptions AS MergeOptions class
Name
NextCursorableDivision
NextName
NextNeighbor
NonUserDocument
NumDisplayablePagesInDoc
NumPagesInDoc
NumWindowsViewingDoc
NumWordsInDoc
OleContainer
OleContainerDocName
OleEmbeddedSize
OleMinHeight
OrigFileType
Parent AS BaseObject class
ParentName
Path
PathName
PreviousCursorableDivision
PreviousName
PreviousNeighbor
PrintManager AS PrintManager class
PrintSettings AS PrintSettings class
ReadCompressed
ReadOnly
RevisionMarkMode
Saved

SelectionType
SortOptions AS SortOptions class
StateID
StyleSheetFullPath
StyleSheetName
StyleSheetPath
TabSpacing
TitleBarDocNumber
UseContents
UsedFirstPageHeight
VersionID
VersionManager AS VersionManager class
WPDataSets AS WPDataSetCollection class

Methods

Activate
AddDivision
AdviseOnRename
AdviseOnSave
AnyEdits
BeginChange
Clear
Close
ConsistencyCheck
Copy
CopySelection
CutSelection
Destroy
EndChange
ForceDocToLoad
FXGetNotesString
FXGetNotesWriteHandle
FXSetNotesString
GetActiveList
GetNamedProperty
GetNameFromPage
HasNamedProperty
Hit
Link
Localize
Move
Paste
Present
Print
PrintOut
Purge
Redo

RegisterWPDataSet
RemoveDepOnDocFile
RemoveDivision
RemoveNamedProperty
Save
SaveAs
SaveDivision
SavePreviewFile
SetDocumentEpoch
SetNamedProperty
Undo
Unlink
UnregisterWPDataSet
UpdateOle
UpdatePowerFields
UpdatePowerFieldsOnNew

Events

Created
DocumentInsert
DocumentInserted
EnterClickHere
EnterLayout
ExitClickHere
GotFocus
ImportInsert
ImportInserted
KeyStroke
LostFocus
MouseDown
MouseUp
Opened
PreClose
PrePrint
Printed
Save
SaveAs
Saved
SavedAs
WMCommand

Word Pro: TextMarkerCollection class members

Properties

Application AS WPAApplication class

Count

Description

IsValid

Name

Parent AS BaseObject class

VersionID

Methods

IsEmpty

Item

Events

None

Word Pro: TextMarker class members

Properties

[Alignment](#) AS [Alignment class](#)
[Amikake](#) AS [Amikake class](#)
[Application](#) AS [WPAApplication class](#)
[AtBeginning](#)
[AtBeginningOfLine](#)
[AtBeginningOfObject](#)
[AtBeginningOfParagraph](#)
[AtBeginningOfWord](#)
[AtEnd](#)
[AtEndOfLine](#)
[AtEndOfObject](#)
[AtEndOfParagraph](#)
[AtEndOfWord](#)
[Attributes](#) AS [Attributes class](#)
[Breaks](#) AS [Breaks class](#)
[Bullet](#) AS [Bullet class](#)
[CharacterBorder](#) AS [CharacterBorder class](#)
[CharacterStyle](#) AS [CharacterStyle class](#)
[CharacterStyleName](#)
[CodePage](#)
[Collapsible](#)
[ColumnNumber](#)
[ColumnWidth](#)
[CurrentLanguage](#)
[Description](#)
[DivisionInfo](#) AS [DivisionInfo class](#)
[EffectiveColumnWidth](#)
[Expandable](#)
[FieldType](#)
[Font](#) AS [Font class](#)
[HasLocalTabs](#)
[Indent](#) AS [Indent class](#)
[IsChanged](#)
[IsInBulletEditMode](#)
[IsMarkerValid](#)
[IsOverridden](#)
[IsParagraphParent](#)
[IsRegistered](#)
[IsRevisionMark](#)
[IsValid](#)
[Kinsoku](#) AS [Kinsoku class](#)
[Language](#) AS [Language class](#)
[LastEditorName](#)
[Layout](#) AS [Layout class](#)

[LayoutName](#)
[MaintainEditor](#)
[MarkerClass](#)
[MultiCompareParaTag](#)
[MultiCompareParaTagSet](#)
[Name](#)
[NextText](#)
[NormalParagraph](#)
[Numbering AS Numbering class](#)
[NumberOfRevisions](#)
[NumCharsInParagraph](#)
[NumCols](#)
[NumRows](#)
[ObjectType](#)
[PageNumber](#)
[PageNumberAsText](#)
[ParagraphBackground AS ParagraphBackground class](#)
[ParagraphBorder AS ParagraphBorder class](#)
[ParagraphHasDropCap](#)
[ParagraphHasText](#)
[ParagraphStyle AS ParagraphStyle class](#)
[ParagraphStyleName](#)
[Parent AS BaseObject class](#)
[Partial](#)
[PositionXInContainer](#)
[PositionXOnPage](#)
[PositionYInContainer](#)
[PositionYOnPage](#)
[RelativeIndent AS RelativeIndent class](#)
[RenderedPageNumber](#)
[RevisionMark AS Revision class](#)
[SectionName](#)
[SectionUserName](#)
[SelectionHidden](#)
[SelectionType](#)
[Spacing AS Spacing class](#)
[StartColumns](#)
[StartRow](#)
[StateID](#)
[StyleExceptions](#)
[TabRack AS TabRack class](#)
[TextMarkersInSelection AS StringCollection class](#)
[TextViewAttributes](#)
[VersionID](#)
[WPDataSets AS WPDataSetCollection class](#)

Methods

Add
AdjustShade
Backspace
Backward
CalculateSmartLevels
CanRepeatClickHere
Clear
CloseObject
ContractOutlineLevel
DeleteChars
DeleteContents
DeleteMarker
Demote
Deselect
DoesMultiCellPaste
Embed
ExpandOutline
Find
Forward
GetContents
GetCopyFormatCategories
GetCount
GetCurrentMarkerName
GetMarkedText
GetMisspelledWord
GetNamedProperty
getparagraphnumber
GetParagraphNumberString
GetParaNumber
GetPasteFormatCategories
GetPosition
GetSpellStatus
GetSpellUserDictStatus
GetText
GetWordMisspelled
GoTo
HasNamedProperty
HideCaretAndSelection
IndexAll
InsertBreak
InsertDocInfo
InsertHardSpace
InsertMarker
InsertNumber
InsertPageNumber
InsertSpecialTab
InsertTab

InsertText
InternalCopy
InternalCut
InternalPaste
IsMarkerEqualToSelection
IsPointWithin
Mark
MorphSelectionToTable
MoveDown
MoveParagraph
MoveToEnd
MoveToStart
MoveUp
Next
NextToObject
OpenObject
Previous
Promote
registerwpdataset
Remove
RemoveNamedProperty
Replace
RevertToStyle
RevisionAccept
RevisionCancel
Select
SetCursorPosition
SetNamedProperty
SetStyle
Shade
ShowCaretAndSelection
ShowCursor
Skip
SortParagraphs
SpellWord
SplitParagraph
SRReplace
UnregisterWPDataSet

Events

None

Word Pro: TextStyleCollection class members

Properties

Application AS WPAApplication class

Count

Description

IsValid

Name

Parent AS BaseObject class

VersionID

Methods

IsEmpty

Item

Events

None

Word Pro: Text class members

Properties

[Alignment](#) AS [Alignment class](#)
[Amikake](#) AS [Amikake class](#)
[Application](#) AS [WPAApplication class](#)
[AtBeginning](#)
[AtBeginningOfLine](#)
[AtBeginningOfObject](#)
[AtBeginningOfParagraph](#)
[AtBeginningOfWord](#)
[AtEnd](#)
[AtEndOfLine](#)
[AtEndOfObject](#)
[AtEndOfParagraph](#)
[AtEndOfWord](#)
[Attributes](#) AS [Attributes class](#)
[Breaks](#) AS [Breaks class](#)
[Bullet](#) AS [Bullet class](#)
[CanEmbed](#)
[CharacterBorder](#) AS [CharacterBorder class](#)
[CharacterStyle](#) AS [CharacterStyle class](#)
[CharacterStyleName](#)
[CodePage](#)
[ColumnNumber](#)
[ColumnWidth](#)
[ContentType](#)
[CurrentLanguage](#)
[Description](#)
[EffectiveColumnWidth](#)
[FieldType](#)
[Font](#) AS [Font class](#)
[HasLocalTabs](#)
[Indent](#) AS [Indent class](#)
[IsEmpty](#)
[IsInBulletEditMode](#)
[IsOverridden](#)
[IsParagraphParent](#)
[IsReplaceable](#)
[IsRevisionMark](#)
[IsValid](#)
[Kinsoku](#) AS [Kinsoku class](#)
[Language](#) AS [Language class](#)
[LastEditorName](#)
[LayoutName](#)
[MaintainEditor](#)
[MultiCompareParaTag](#)

[MultiCompareParaTagSet](#)
[Name](#)
[NextText](#)
[NormalParagraph](#)
[Numbering AS Numbering class](#)
[NumberOfRevisions](#)
[NumCharsInParagraph](#)
[ObjectType](#)
[ParagraphBackground AS ParagraphBackground class](#)
[ParagraphBorder AS ParagraphBorder class](#)
[ParagraphHasDropCap](#)
[ParagraphHasText](#)
[ParagraphStyle AS ParagraphStyle class](#)
[ParagraphStyleName](#)
[Parent AS BaseObject class](#)
[Partial](#)
[PositionXInContainer](#)
[PositionXOnPage](#)
[PositionYInContainer](#)
[PositionYOnPage](#)
[RelativeIndent AS RelativeIndent class](#)
[RenderedPageNumber](#)
[RevisionMark AS Revision class](#)
[SectionName](#)
[SectionUserName](#)
[SelectionHidden](#)
[SelectionType](#)
[Spacing AS Spacing class](#)
[StyleExceptions](#)
[TabRack AS TabRack class](#)
[TextMarkersInSelection AS StringCollection class](#)
[TextViewAttributes](#)
[VersionID](#)
[WPDataSets AS WPDataSetCollection class](#)

Methods

[Add](#)
[AdjustShade](#)
[Backspace](#)
[Backward](#)
[CalculateSmartLevels](#)
[CanRepeatClickHere](#)
[Clear](#)
[CloseObject](#)
[ContractOutlineLevel](#)
[DeleteChars](#)
[Demote](#)

Deselect
DoesMultiCellPaste
Embed
ExpandOutline
Find
Forward
GetCopyFormatCategories
GetCount
GetCurrentMarkerName
GetMisspelledWord
GetParaNumber
GetPasteFormatCategories
GetPosition
GetSpellStatus
GetSpellUserDictStatus
GetText
GetWordMisspelled
HideCaretAndSelection
IndexAll
InsertBreak
InsertDocInfo
InsertHardSpace
InsertNumber
InsertPageNumber
InsertSpecialTab
InsertTab
InsertText
InternalCopy
InternalCut
InternalPaste
IsMarkerEqualToSelection
IsPointWithin
Mark
MorphSelectionToTable
MoveDown
MoveParagraph
MoveToEnd
MoveToStart
MoveUp
Next
NextToObject
OpenObject
Previous
Promote
RegisterWPDataSet
Remove
Replace

RevertToStyle

RevisionAccept

RevisionCancel

Select

SetCursorPosition

SetStyle

Shade

ShowCaretAndSelection

ShowCursor

Skip

SortParagraphs

SpellWord

SplitParagraph

SRReplace

UnregisterWPDataSet

Events

None

Word Pro: TOCSuperTableLayout class members

Properties

[AbsoluteOn](#)

[AbsoluteXPos](#)

[AbsoluteYPos](#)

[Accelerator](#)

[AllowResizeWhenCrop](#)

[AmtTether](#)

[AmtToRotateContent](#)

[AmtToTetherFrom](#)

[Application](#) AS [WPAApplication class](#)

[Background](#) AS [Background class](#)

[BaseLineOffset](#)

[BinName](#)

[BorderLines](#) AS [BorderLines class](#)

[BorderOffset](#)

[BottomExternalMargin](#)

[Center](#)

[ChildLayouts](#) AS [LayoutCollection class](#)

[ClassName](#)

[ColumnBalance](#)

[ColumnGap](#)

[ConditionType](#)

[content](#)

[ContentName](#)

[ContentStyleName](#)

[Definition](#)

[Description](#)

[DirectionDown](#)

[DirectionLeft](#)

[DirectionRight](#)

[DirectionUp](#)

[DivisionName](#)

[DropCapPosition](#)

[EditorName](#)

[Footer](#) AS [Layout class](#)

[GridDistance](#)

[GridType](#)

[Gutter](#) AS [Gutter class](#)

[Header](#) AS [Layout class](#)

[Height](#)

[IsBreakable](#)

[IsChildSpannable](#)

[IsCollapsed](#)

[IsCollapsible](#)

[IsColumnBreakable](#)

[IsComplex](#)
[IsConnected](#)
[IsErrorChecking](#)
[IsExpandDown](#)
[IsExpandLeft](#)
[IsExpandRight](#)
[IsExpandUp](#)
[IsIndexMarkAble](#)
[IsLocal](#)
[IsLocked](#)
[IsMarginSameAsParent](#)
[IsNotCopyable](#)
[IsNotGroupable](#)
[IsNoUICommAllowed](#)
[IsOverridden](#)
[IsOverride](#)
[IsPageBreak](#)
[IsPartOfGroup](#)
[IsPrintable](#)
[IsProtected](#)
[IsScrollable](#)
[IsSingleClickEntry](#)
[IsSizable](#)
[IsSnapTo](#)
[IsStyle](#)
[IsTableHeading](#)
[IsTOC](#)
[IsTOCMarkAble](#)
[IsValid](#)
[Join AS Join class](#)
[Justifiable](#)
[Layer AS Layout class](#)
[LayerName](#)
[LeaderDotType](#)
[LeftExternalMargin](#)
[LeftPage AS Layout class](#)
[LeftTopCellId](#)
[LineLocation](#)
[LinkFrame](#)
[MaintainAspectRatio](#)
[MarginBottom](#)
[MarginLeft](#)
[MarginRight](#)
[MarginTop](#)
[MasterName](#)
[MinBottomMargin](#)
[MinHeight](#)

[MinLeftMargin](#)
[MinRightMargin](#)
[MinTopMargin](#)
[Name](#)
[NameBasedOnStyle](#)
[NumberOfLines](#)
[NumCols](#)
[NumColsSpannedOneCell](#)
[NumericFormat](#) AS [NumericFormat class](#)
[NumRowsSpannedOneCell](#)
[PageToUseLayoutOn](#)
[PaperName](#)
[Parent](#) AS [BaseObject class](#)
[ParentName](#)
[RelativeType](#)
[RelativeXDistance](#)
[RelativeYDistance](#)
[RevisionType](#)
[RightExternalMargin](#)
[RightPage](#) AS [Layout class](#)
[ScaleHeight](#)
[ScaleMode](#)
[ScalePercentage](#)
[ScaleWidth](#)
[SectionName](#)
[SelectType](#)
[Shadow](#) AS [Shadow class](#)
[Span](#)
[Style](#) AS [Layout class](#)
[StyleExceptions](#)
[TabRack](#) AS [TabRack class](#)
[TextOrient](#)
[Tile](#)
[TOCRange](#)
[TOCSource](#)
[TopExternalMargin](#)
[TopLeftCellRowId](#)
[UseFooter](#)
[UseHeader](#)
[UsePrinterSettings](#)
[UseWhen](#) AS [UseWhen class](#)
[VersionID](#)
[VertAlign](#)
[WasDeletedInRevMarkMode](#)
[WasInsertedInRevMarkMode](#)
[Where](#)
[Width](#)

WPDataSets AS WPDataSetCollection class

WrapType

XOffset

XPosition

YOffset

YPosition

Methods

AddChildToLayout

AddTOCEntry

Backward

Clear

createlayer

DeleteContents

DeleteLayout

DoesMarkerNameMatch

FindClass

Forward

GetMarkerName

GetNamedProperty

GoToLayout

HasNamedProperty

ImportWatermarkGraphic

Mark

MirrorPage

MoveToBack

MoveToFront

registerwpdataset

RemoveChildFromLayout

RemoveNamedProperty

RevisionAcceptLayoutChange

RevisionCancelLayoutChange

setallmargins

SetMinimumOrigin

SetNamedProperty

SetPaperNameAndUpdateSize

SetupForCropping

UnregisterWPDataSet

Update

Events

EnterLayout

KeyStroke

MouseDown

MouseUp

Word Pro: UserInterfacePrefs class members

Properties

[Application](#) AS [WPAApplication class](#)
[AutoBackup](#)
[AutoSave](#)
[AutoSaveMinutes](#)
[BackupPath](#)
[BackupPaths](#) AS [StringCollection class](#)
[BulletFonts](#)
[CheckForScriptEnumError](#)
[CheckForScriptPropertyError](#)
[CleanScreenMode](#)
[CleanScreenOnStartUp](#)
[DefaultNewCategory](#)
[Description](#)
[DisableConsistencyCheck](#)
[DisableHarmlessMessages](#)
[DocPath](#)
[DocumentPaths](#) AS [StringCollection class](#)
[FontUnitName](#)
[FontUnits](#)
[GlossaryDataFileName](#)
[GlossaryDataFiles](#) AS [StringCollection class](#)
[GlossaryDataPaths](#) AS [StringCollection class](#)
[GlossaryPath](#)
[GrammarOptions](#) AS [Grammar class](#)
[GraphicPath](#)
[GraphicPaths](#) AS [StringCollection class](#)
[HorizontalSplitWindow](#)
[HTMLOptions](#) AS [HTMLOptions class](#)
[IconPath](#)
[IconPaths](#) AS [StringCollection class](#)
[IsDisableWarningMessages](#)
[IsHighlightNote](#)
[IsValid](#)
[IsWorkingDir](#)
[LastDocPath](#)
[LastGraphicPath](#)
[LastGraphicType](#)
[LastInsertSymbolFont](#)
[LastMacroPath](#)
[LastUsedDateFormula](#)
[LeaderDotType](#)
[LoadFilesMaximized](#)
[MacroPath](#)
[MacroPaths](#) AS [StringCollection class](#)

[MaximizeOnStartUp](#)
[MenuPath](#)
[MenuPaths AS StringCollection class](#)
[Name](#)
[NumOfRecentFiles](#)
[OpenDocsVisible](#)
[OpenExisingFileInWelcomeBox](#)
[OpenReadOnly](#)
[Parent AS BaseObject class](#)
[RetainNameOfImportedFile](#)
[RunMacroOnDocEvents](#)
[RunMacroOnLoad](#)
[ShowBubbleHelp](#)
[ShowExportWarningMessages](#)
[ShowFileNew](#)
[ShowGraphicPreview](#)
[ShowHeaderFooterBar](#)
[ShowMailDisabled](#)
[ShowMenuCustomization](#)
[ShowMenuWarningMessage](#)
[ShowNoWelcomeBox](#)
[SizingUnitName](#)
[SizingUnits](#)
[SpacingUnitName](#)
[SpacingUnits](#)
[SpellCheckInitialCaps](#)
[SpellCheckRepeatedWords](#)
[SpellCheckUserDictAlternatives](#)
[SpellCheckWordsWithNums](#)
[SplitPercentage](#)
[StatusSpellReplaceAll](#)
[StylePath](#)
[StylePaths AS StringCollection class](#)
[StyleSheetName](#)
[TabRelativeTo](#)
[TabType](#)
[ThesMaxNumSynonymsReturned](#)
[TimedSaveFileExtension](#)
[Units AS StringCollection class](#)
[UseCycleKeys](#)
[UserDefinedFilter](#)
[UserDictFiles](#)
[UserDictionaryFiles AS StringCollection class](#)
[UserDictionaryPath](#)
[UserDictionaryPaths AS StringCollection class](#)
[UserDictStates](#)
[VersionID](#)

VerticalSplitWindow

WatermarksPath

WinViewPrefs AS WinViewPrefs class

WorkingType

Methods

GetArrayProp

InsertPath

SetArrayProp

Events

None

Word Pro: UseWhen class members

Properties

Application AS WPAApplication class

Condition

Description

IsValid

Name

PageNo

Parent AS BaseObject class

Start

VersionID

Methods

None

Events

None

Word Pro: VersionCollection class members

Properties

Application AS WPAApplication class

Count

Description

IsValid

Name

Parent AS BaseObject class

VersionID

Methods

IsEmpty

Item

Events

None

Word Pro: VersionManager class members

Properties

Application AS WPAApplication class

CurrentVersion AS Version class

Description

DocVersionID

GetRedoWhatDesc

GetUndoWhatDesc

IsValid

Name

Parent AS BaseObject class

VersionID

Versions AS VersionCollection class

Methods

CreateVersion

DeleteVersion

LeastRecentVersion

MostRecentVersion

NextVersion

NumberOfVersions

PreviousVersion

Events

None

Word Pro: Version class members

Properties

Application AS WPAApplication class

AuthorName

CreationDateString

CreationTimeString

DateCreatedValue

DateRevisedValue

Description

DocVersionID

EditorName

Editors AS StringCollection class

IsValid

ModifiedDateString

ModifiedTimeString

Name

NumberOfRevisions

Parent AS BaseObject class

VersionID

VersionName

Methods

CreateRemark

Events

None

Word Pro: Window class members

Properties

Active

Application AS WPAApplication class

Caption

Description

Height

IsValid

Left

Name

Parent AS BaseObject class

Top

VersionID

Visible

Width

Methods

Close

Move

Open

Resize

Update

Events

Moved

Word Pro: WinViewPrefs class members

Properties

Application AS WPAApplication class

CustomViewLevel

Description

IsDisplayMisspelled

IsHideHeaderFooter

IsHorizontalScrollBarCleanScrn

IsInDraft

IsInOutline

IsInPageSort

IsMarginsInColor

IsNoPrintZone

IsPageBreakMarks

IsSpellMode

IsValid

IsVerticalScrollBarCleanScrn

IsViewAnchor

IsViewBookmarks

IsViewClickHereBlocks

IsViewColGuides

IsViewColumnBreakMarks

IsViewDDELinks

IsViewHorzRuler

IsViewHorzScrollBar

IsViewMenuCleanScrn

IsViewNotes

IsViewOutlineFlowToScreen

IsViewOutlineIndent

IsViewPageLayoutMarks

IsViewParallelColumnBorder

IsViewPictures

IsViewReturnIconCleanScrn

IsViewReturns

IsViewRubyMarks

IsViewRulerMarks

IsViewSectionBreakMarks

IsViewSectionTabs

IsViewSmartIconsCleanScrn

IsViewStatusBarCleanScrn

IsViewStatusSpell

IsViewTableCellSelection

IsViewTableGridLines

IsViewTableHeadings

IsViewTabs

IsViewThumbBarCleanScrn

IsViewTitleBarCleanScrn

IsViewVertRuler

IsViewVertScrollBar

Name

NumCols

OutlineButtons

OutlineHeadingButtonsOnly

OutlineOnlyHeadingsWhenCollapsed

Parent AS BaseObject class

VersionID

ViewType

Methods

ClearSplits

ClearUpdate

Refresh

RestorePreviousView

Events

None

Word Pro: WPAApplication class members

Properties

[ActiveDocument](#) AS [TextDocument class](#)
[ActiveDocWindow](#) AS [DocWindow class](#)
[AppFoundry](#) AS [Foundry class](#)
[Application](#) AS [WPAApplication class](#)
[ApplicationWindow](#) AS [ApplicationWindow class](#)
[AppViewPrefs](#) AS [AppViewPrefs class](#)
[BaseTable](#)
[Cell](#) AS [CellContainer class](#)
[Container](#)
[Content](#)
[CurrentCell](#) AS [CellLayout class](#)
[CurrentColumn](#) AS [Layout class](#)
[CurrentRow](#) AS [RowLayout class](#)
[DefaultFilePath](#)
[Description](#)
[Division](#) AS [Division class](#)
[Divisions](#) AS [DivisionCollection class](#)
[Documents](#) AS [Documents class](#)
[FindAndReplace](#) AS [FindAndReplace class](#)
[FormatCheckPreferences](#) AS [FormatCheckPref class](#)
[Foundry](#) AS [Foundry class](#)
[Frame](#) AS [FrameContainer class](#)
[FullName](#)
[GetHomeDirectory](#)
[Graphic](#) AS [Graphic class](#)
[GraphicOleObject](#)
[Interactive](#)
[IsValid](#)
[KeyboardLanguage](#)
[Language](#)
[LastMessageID](#)
[LastMessageStr](#)
[LastMessageType](#)
[Layout](#) AS [Layout class](#)
[Location](#)
[Name](#)
[OleObject](#) AS [OleObject class](#)
[OSType](#)
[Page](#) AS [PageContainer class](#)
[ParallelColumns](#) AS [ParallelColumns class](#)
[Parent](#) AS [BaseObject class](#)
[Path](#)
[Preferences](#) AS [Preferences class](#)
[ProductVersion](#)

[ReleaseNumber](#)
[SmartCorrect](#) AS [SmartCorrect](#) class
[SmartCorrects](#) AS [SmartCorrectCollection](#) class
[SmartFill](#) AS [SmartFillCollection](#) class
[SuperTableContainer](#) AS [SuperTableContainer](#) class
[Table](#) AS [Table](#) class
[TableContainer](#)
[TableOnlyContainer](#) AS [TableOnlyCont](#) class
[TempFindAndReplace](#) AS [FindAndReplace](#) class
[TempFoundry](#) AS [Foundry](#) class
[Text](#) AS [Text](#) class
[VersionID](#)
[Visible](#)
[WPDataSets](#) AS [WPDataSetCollection](#) class

Methods

[AcquireTWAINImage](#)
[AddContainer](#)
[AddIndexAllEntry](#)
[AddIndexEntry](#)
[AddTOCEntry](#)
[AnswerMsgBox](#)
[BeginChange](#)
[Bold](#)
[BringFrameToFront](#)
[BringFrameToFrontOne](#)
[CalculateSmartLevels](#)
[CascadeWindow](#)
[CellRevert](#)
[ChangeSmartMaster](#)
[CheckFieldEntries](#)
[ClearParaRevisionTags](#)
[ClearSpellHints](#)
[ClearTempFoundry](#)
[Close](#)
[CloseAll](#)
[CloseDocWindow](#)
[CloseMergeDataFile](#)
[CombineDivisions](#)
[CombineSections](#)
[CompareFiles](#)
[ConnectCells](#)
[ConnectContainer](#)
[ConnectRows](#)
[ContractOutlineLevel](#)
[CopyDivision](#)
[CopySelection](#)

[CreateDataFile](#)
[CreateDivision](#)
[CreateDropCap](#)
[CreateExternalDivision](#)
[CreateFrame](#)
[CreateGlossary](#)
[CreateGlossaryEntry](#)
[CreateGraphic](#)
[CreateOleEmbeddedFile](#)
[CreateOleFromFile](#)
[CreateOleLinkedFile](#)
[CreateOleNew](#)
[CreateParallelColumns](#)
[CreateTable](#)
[CreateTextEntryField](#)
[CutDivision](#)
[CutSelection](#)
[DbUnderline](#)
[DeleteClickHere](#)
[DeleteDivision](#)
[DeleteKey](#)
[DeleteParallelColumns](#)
[DeleteSection](#)
[DeleteTable](#)
[DemoteOutlineLevel](#)
[Deselect](#)
[Disconnect](#)
[DisconnectCells](#)
[DoVerb](#)
[EditHyperlink](#)
[Embed](#)
[EndChange](#)
[EnvelopeBarCode](#)
[EnvelopePrint](#)
[ExpandOutlineLevel](#)
[FrameRevert](#)
[GetCopyFormatCategories](#)
[GetCurrentMarkerName](#)
[GetEnum](#)
[GetInternetFile](#)
[GetPasteFormatCategories](#)
[GetProfileString](#)
[GlossaryInsert](#)
[GlossaryOpen](#)
[GoToBookmark](#)
[GoToClickHere](#)
[GoToDivision](#)

[GoToLayout](#)
[GotoNextParallelColumn](#)
[GoToObject](#)
[GoToPage](#)
[GoToSection](#)
[GroupDivision](#)
[HandsOffStorage](#)
[Help](#)
[Hide](#)
[HighlightToggle](#)
[HourGlass](#)
[ImportGraphic](#)
[InitFindAndReplace](#)
[InsertBullet](#)
[InsertClickHere](#)
[InsertColumnBreak](#)
[InsertDate](#)
[InsertDocInfo](#)
[InsertDocument](#)
[InsertField](#)
[InsertFootnote](#)
[InsertHyperlink](#)
[InsertIndex](#)
[InsertNote](#)
[InsertOleDivision](#)
[InsertPageBreak](#)
[InsertPageLayout](#)
[InsertPageNumber](#)
[InsertRuby](#)
[InsertSection](#)
[InsertTOC](#)
[InternalCopy](#)
[InternalCut](#)
[InternalPaste](#)
[InternetFileExists](#)
[IsWMCCommandValid](#)
[Italic](#)
[LabelCreate](#)
[LabelMerge](#)
[LinkFrameContents](#)
[LowerCase](#)
[MacroEndRecord](#)
[MacroPlay](#)
[MacroRecord](#)
[MailDocument](#)
[MakeTableFromText](#)
[ManualFrame](#)

[ManualLinkFrames](#)
[ManualTable](#)
[Mark](#)
[MarkRevisionInsert](#)
[Merge](#)
[MergeAddDataRecord](#)
[MergeSetDataFile](#)
[MergeStart](#)
[Messages](#)
[MoveDivision](#)
[NewDivision](#)
[NewDocument](#)
[NewFrame](#)
[NewInfoBusLink](#)
[NewWindow](#)
[NextCycleAlign](#)
[NextCycleAttribute](#)
[NextCycleBullet](#)
[NextCycleFont](#)
[NextCycleFontSize](#)
[NextCycleIndent](#)
[NextCycleNumber](#)
[NextCycleStyle](#)
[NormalText](#)
[OpenDataFile](#)
[OpenDocument](#)
[OpenDocumentFromNotes](#)
[OpenFromStorage](#)
[OutlineMoveTextDown](#)
[OutlineMoveTextUp](#)
[PageDown](#)
[PageUp](#)
[Paste](#)
[PasteDivision](#)
[PasteLink](#)
[PasteSpecial](#)
[PColConnectCells](#)
[PColConnectRows](#)
[PColDisconnectCells](#)
[PColSelectColumn](#)
[PColSelectRow](#)
[PColSelectTable](#)
[Print](#)
[PrintOut](#)
[PromoteOutlineLevel](#)
[QuickAlignFrame](#)
[QuickAlignTable](#)

Quit
Redo
RegisterWPDataSet
RemoveHyperlink
RemoveDepOnDocFile
RemoveIndexEntry
RemoveTOCEntry
Render
Replace
ReplaceAll
ReplaceCmd
ResetFindAndReplace
ResetNumberOpts
RetrieveInternetFile
RetrieveInternetFileAndOpen
RevertToSaved
RevertToStyle
ReviseAcceptAll
ReviseCancelAll
RunScript
Save
SaveAs
SaveAsToNotes
SaveGlossary
SaveInternetFile
SaveMergeDataFile
SaveThumbnailBitmap
SaveToStorage
SaveToStorageComplete
SaveVersion
Search
Select
SelectCell
SelectColumn
SelectDoc
SelectEntireCellRange
SelectEntireColumn
SelectEntirePCol
SelectEntirePColCellRange
SelectEntirePColColumn
SelectEntirePColRow
SelectEntireRow
SelectEntireTable
SelectParagraph
SelectPColCell
SelectRow
SelectSection

SelectSentence
SelectTable
SelectTWAInSource
SelectWord
SendFrameToBack
SendFrameToBackOne
SendMailAndAttach
SendMailSelectedText
SetCustomNumber
SetIndexInfo
SetJapanIndexInfo
SetMultiCellPaste
SetNumberingLevelInfo
SetPageBottomMargin
SetPageTopMargin
SetStorage
SetStyle
SetTOCLevelContent
SetTOCLevelPageInfo
SetupEnvelopeMerge
ShowAnyGreeting
SmallCaps
SmartSumColumn
SmartSumRow
SortParagraphs
SpecialView
SpellAddToUserDict
SpellClearSkippedWords
SpellMarkSkippedWords
SpellSkipAll
SplitDivision
SplitWindow
StartEditMergeData
StartEnvelopeDiv
StartFieldInsert
StoreInternetFile
StrikeThru
SubScript
SuperScript
TeamMail
TileWindowHorz
TileWindowVert
TimedSave
ToggleCleanScreen
ToggleIconBar
Type
Underline

Undo
UnlinkFrameContents
UnregisterWPDataSet
UpdateFootersText
UpdateHeadersText
UpdateIndexSection
UpdateOle
UpdateTOC
UpdateUI
UpperCase
UseLSX
WMCommand
WordCount
WordUnderline
WriteProfileString

Events

DocumentClose
DocumentClosed
DocumentCreate
DocumentCreated
DocumentExport
DocumentExported
DocumentImport
DocumentImported
DocumentInsert
DocumentInserted
DocumentOpen
DocumentOpened
DocumentPrint
DocumentPrinted
DocumentSave
DocumentSaveAs
DocumentSaved
DocumentSavedAs
EnterClickHere
EnterLayout
ExitClickHere
ImportInsert
ImportInserted
KeyStroke
MouseDown
MouseUp
Quit
WMCommand

Word Pro: WPDataSetCollection class members

Properties

Application AS WPAApplication class

Count

Description

IsValid

Name

Parent AS BaseObject class

VersionID

Methods

IsEmpty

Item

Events

None

Word Pro: WPDataSet class members

Properties

Application AS WPAApplication class

DataNames AS StringCollection class

Description

IsPersistent

IsValid

Name

Parent AS BaseObject class

VersionID

Methods

GetData

IsDataNameUsed

SetData

Events

None

Word Pro: ReviewVersions method

{button ,AL(^H_REVIEWVERSIONS_CLASS',0)} [See list of classes](#)

Syntax

[objectreference].ReviewVersions()

Parameters

Return value

Usage

Word Pro: ReviseAcceptAll method

{button ,AL('H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL('H_REVISEACCEPTALL_METHOD_EXSCRIPT',1)} [See example](#)

Accepts revisions according to the instructions provided in the parameters. You can accept revisions for the current paragraph or the entire document, from a single editor or all editors, and in one area of a document or the entire document.

Syntax

[objectreference].ReviseAcceptAll(ReviseCurrentPara, [EditorName,] [MarkerName])

Parameters

ReviseCurrentPara

Allows you to specify whether you want to accept the revisions for the current paragraph (True) or for the entire document (False). Data type is Integer. The legal values for this parameter are -1 and 0 but you may use the LotusScript constants True (-1) and False (0).

EditorName

A String expression specifying the name of the editor whose revisions you want to accept. Optional parameter. If you do not provide a value for this parameter, Word Pro accepts all the revisions without regard to their source.

MarkerName

A String expression specifying the name of a marker that identifies the part of the document in which you want to accept revisions. Use to mark an area of a document for revision when you want to accept revisions only in the marked area. Optional parameter.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Word Pro: ReviseCancelAll method

{button ,AL('H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL('H_REVISECANCELALL_METHOD_EXSCRIPT',1)} [See example](#)

Rejects revisions according to the instructions provided in the parameters. You can reject revisions for the current paragraph or the entire document, from a single editor or all editors, and in one area of a document or the entire document.

Syntax

[objectreference].ReviseCancelAll(ReviseCurrentPara, [EditorName,] [MarkerName])

Parameters

ReviseCurrentPara

Allows you to specify whether you want to reject the revisions for the current paragraph (True) or for the entire document (False). Data type is Integer. The legal values for this parameter are -1 and 0 but you may use the LotusScript constants of True (-1) and False (0).

EditorName

A String expression specifying the name of the editor whose revisions you want to reject. Optional parameter. If you do not provide a value for this parameter, Word Pro will reject all the revisions without regard to their source.

MarkerName

A String expression specifying the name of a marker that identifies the part of the document in which you want to reject revisions. Use when you mark an area of a document for revision and you want to reject revisions only in the marked area. Optional parameter.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Word Pro: RevisionAcceptLayoutChange method

```
{button ,AL('H_LAYOUT_CLASS;H_CELLGROUPLAYOUT_CLASS;H_CELLLAYOUT_CLASS;H_CONNECTEDLAY  
OUT_CLASS;H_COLUMNGROUPLAYOUT_CLASS;H_FOOTERLAYOUT_CLASS;H_FRAMELAYOUT_CLASS;H  
_FRAMEGROUPLAYOUT_CLASS;H_DROP_CAPLAYOUT_CLASS;H_GROUPLAYOUT_CLASS;H_HEADERLAY  
OUT_CLASS;H_NOTELAYOUT_CLASS;H_PAGELAYOUT_CLASS;H_ROWGROUPLAYOUT_CLASS;H_ROWLA  
YOUT_CLASS;H_RUBYLAYOUT_CLASS;H_SUPERTABLEGROUPLAYOUT_CLASS;H_SUPERTABLELAYOUT  
_CLASS;H_TABLELAYOUT_CLASS;H_ENDNOTELAYOUT_CLASS;H_FOOTNOTELAYOUT_CLASS;H_TABLE  
HEADINGLAYOUT_CLASS;H_TOCSUPERTABLELAYOUT_CLASS',0)} See list of classes
```

```
{button ,AL('H_REVISIONACCEPTLAYOUTCHANGE_METHOD_EXSCRIPT',1)} See example
```

Accepts revision markup changes made in a Layout object.

Syntax

[objectreference].RevisionAcceptLayoutChange(ReviseAll, [EditorName])

Parameters

ReviseAll

Indicates whether you want to accept all revisions or just the revision at the insertion point. If text is selected and you use True (-1), Word Pro ignores the selection and accepts the revisions for the entire document or named range.

Data type is Integer. The legal values for this parameter are -1 and 0 but you may use the LotusScript constants True (-1) and False (0) instead of the integer values. There is no default value.

EditorName

A String expression which specifies the name of the editor whose revisions you want to accept. If no editor name is provided, all editors' revisions are accepted. Optional parameter.

Return value

The return value for this method will always be -1 or 0. When testing the return value, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Word Pro: RevisionAccept method

{button ,AL('H_CLICKHERE_CLASS;H_TEXTMARKER_CLASS;H_TEXT_CLASS',0)} [See list of classes](#)

Accepts one or all revisions from all editors or a named editor.

Syntax

[objectreference].RevisionAccept(ReviseAll, [EditorName,] [MarkerName])

Parameters

ReviseAll

Indicates whether you want to accept all revisions or just the revision at the insertion point. If text is selected and you use True (-1), Word Pro ignores the selection and accepts the revisions for the entire document or named range.

Data type is Integer. The legal values for this parameter are -1 and 0 but you may use the LotusScript constants True (-1) and False (0) instead of the integer values. There is no default value.

EditorName

A String expression which specifies the name of the editor whose revisions you want to accept. If no editor name is provided, all editors' revisions are accepted. Optional parameter.

MarkerName

A String expression which specifies the name of a marked range. If you provide a range name in this parameter and use True for the ReviseAll parameter, Word Pro accepts all the revisions within the named range. Optional parameter.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Word Pro: RevisionCancelLayoutChange method

```
{button ,AL('H_LAYOUT_CLASS;H_CELLGROUPLAYOUT_CLASS;H_CELLLAYOUT_CLASS;H_CONNECTEDLAY  
OUT_CLASS;H_COLUMNGROUPLAYOUT_CLASS;H_FOOTERLAYOUT_CLASS;H_FRAMELAYOUT_CLASS;H  
_FRAMEGROUPLAYOUT_CLASS;H_DROP_CAPLAYOUT_CLASS;H_GROUPLAYOUT_CLASS;H_HEADERLAY  
OUT_CLASS;H_NOTELAYOUT_CLASS;H_PAGELAYOUT_CLASS;H_ROWGROUPLAYOUT_CLASS;H_ROWLA  
YOUT_CLASS;H_RUBYLAYOUT_CLASS;H_SUPERTABLEGROUPLAYOUT_CLASS;H_SUPERTABLELAYOUT  
_CLASS;H_TABLELAYOUT_CLASS;H_ENDNOTELAYOUT_CLASS;H_FOOTNOTELAYOUT_CLASS;H_TABLE  
HEADINGLAYOUT_CLASS;H_TOCSUPERTABLELAYOUT_CLASS',0)} See list of classes
```

```
{button ,AL('H_REVISIONCANCELLAYOUTCHANGE_METHOD_EXSCRIPT',1)} See example
```

Cancels revision markup changes made in a Layout object.

Syntax

```
[objectreference].RevisionCancelLayoutChange(ReviseAll, [EditorName])
```

Parameters

ReviseAll

Indicates whether you want to cancel all revisions or just the revision at the insertion point. If text is selected and you use True (-1), Word Pro ignores the selection and cancels the revisions for the entire document or named range. Data type is Integer. The legal values for this parameter are -1 and 0 but you may use the LotusScript constants True (-1) and False (0) instead of the integer values. There is no default value.

EditorName

A String expression which specifies the name of the editor whose revisions you want to cancel. If no editor name is provided, all editor revisions are cancelled. Optional parameter.

Return value

The return value for this method will always be -1 or 0. When testing the return value, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Word Pro: RevisionCancel method

{button ,AL('H_CLICKHERE_CLASS;H_TEXTMARKER_CLASS;H_TEXT_CLASS',0)} [See list of classes](#)

Cancels one or all revisions from all editors or a named editor.

Syntax

[objectreference].RevisionCancel(ReviseAll, [EditorName,] [MarkerName])

Parameters

ReviseAll

Indicates whether you want to cancel all revisions or just the revision at the insertion point. If text is selected and you use True (-1), Word Pro ignores the selection and cancels the revisions for the entire document or named range.

Data type is Integer. The legal values for this parameter are -1 and 0 but you may use the LotusScript constants True (-1) and False (0) instead of the integer values. There is no default value.

EditorName

A String expression which specifies the name of the editor whose revisions you want to cancel. If no editor name is provided, all editor revisions are cancelled. Optional parameter.

MarkerName

A String expression which specifies the name of a marked range. If you provide a range name in this parameter and use True for the ReviseAll parameter, Word Pro cancels all the revisions within the named range. Optional parameter.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Word Pro: RunScript method

{button ,AL(`H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL(`H_RUNSCRIPT_METHOD_EXSCRIPT',1)} [See example](#)

Runs the specified Word Pro script.

Syntax

[objectreference].RunScript(ScriptModule, ScriptFunction)

Parameters

ScriptModule

A String expression specifying the name and path of the file containing the script you want to run.

ScriptFunction

A String expression specifying the script you want to run.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Word Pro: Run method

{button ,AL(^H_MACRO_CLASS',0)} [See list of classes](#)

Syntax

[objectreference].Run(p1)

Parameters

p1

Data type is Variant.

Return value

Usage

Word Pro: SaveAsToNotes method

{button ,AL('H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

Saves the active document as an attachment in the specified Notes document.

Note This method only works with Notes Version 3.3 or higher.

Syntax

[objectreference].SaveAsToNotes(UID, Attached, Field, Database, Server, FileType)

Parameters

UID

A String expression which specifies Universal Identifier (UNID) for the Notes document to which you want to attach the active Word Pro document. The UNID can be found in the UniversalID property of the NotesDocument object in the Notes object model.

Attached

A String expression which specifies a name for the file you are attaching to the Notes document. The original active document retains its own name.

Field

A String expression which specifies the name of a Rich Text Field in the specified Notes document. The active Word Pro document will be saved as an attachment within the field you name.

Database

A String expression which specifies the name and path of the Notes database which contains the document you specified in the UID parameter.

Server

A String expression which specifies the name of the Notes server which contains the database you specified in the Database parameter.

FileType

An optional String value that indicates the file type in which the document will be saved. A null string saves the document as a Word Pro file. Some of the usual file types are listed in the table below, but each user's list of available file types is derived from the list of text filters installed during the Word Pro installation.

DCA/RTF	Lotus Manuscript 2.x	MS Word for Windows 1.0
DIF	Lotus Organizer 1.x	MS Word for Windows 2.0
DisplayWrite	Lotus Word Pro	MS Word for Windows 6.0
HTML	Lotus Word Pro SmartMaster	MS Word for Windows95 7.0
Lotus 1-2-3	MS Excel	MS WordPad 1.0
Lotus 1-2-3 for OS/2	MS Excel 3.0	OfficeWriter 4,5,6
Lotus 1-2-3 R3	MS Excel 4.0	Rich Text Format(RTF)
Lotus 1-2-3 R4,5	MS Excel 5.0	SAMNA Word
Lotus 1-2-3 R6	MS Excel 7.0	WordPerfect 5.0
Lotus Ami Pro	MS Windows Write 3.x	WordPerfect 5.1
Lotus Ami Pro 3.x Macro	MS Word for DOS 3,4,5,6	WordPerfect 6.x
Lotus Ami Pro 3.x Styles	MS Word for OS/2	WordStar 2000 R3

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Word Pro: SaveAs method

{button ,AL('H_WPAPPLICATION_CLASS;H_DOCUMENT_CLASS;H_TEXTDOCUMENT_CLASS',0)} [See list of classes](#)

{button ,AL('H_SAVEAS_METHOD_EXSCRIPT',1)} [See example](#)

Saves the active document with a different name or file type.

Syntax

[objectreference].SaveAs([DocName,] [Location,] [DocType,] [Backup,][AddToLastFileOpenList,] [SaveCopyAs])

Parameters

DocName

An optional String value which indicates the name under which the document will be saved.

Location

An optional String value which indicates the path or directory in which the document will be saved.

DocType

An optional String value that indicates the file type in which the document will be saved. A null string saves the document as a Word Pro file. Some of the usual file types are listed in the table below, but each user's list of available file types is derived from the list of text filters installed during the Word Pro installation.

DCA/RFT	Lotus Manuscript 2.x	MS Word for Windows 1.0
DIF	Lotus Organizer 1.x	MS Word for Windows 2.0
DisplayWrite	Lotus Word Pro	MS Word for Windows 6.0
HTML	Lotus Word Pro SmartMaster	MS Word for Windows95 7.0
Lotus 1-2-3	MS Excel	MS WordPad 1.0
Lotus 1-2-3 for OS/2	MS Excel 3.0	OfficeWriter 4,5,6
Lotus 1-2-3 R3	MS Excel 4.0	Rich Text Format(RTF)
Lotus 1-2-3 R4,5	MS Excel 5.0	SAMNA Word
Lotus 1-2-3 R6	MS Excel 7.0	WordPerfect 5.0
Lotus Ami Pro	MS Windows Write 3.x	WordPerfect 5.1
Lotus Ami Pro 3.x Macro	MS Word for DOS 3,4,5,6	WordPerfect 6.x
Lotus Ami Pro 3.x Styles	MS Word for OS/2	WordStar 2000 R3

Backup

An optional Integer value that indicates whether or not Word Pro should make a backup copy of this file before saving.

AddToLastFileOpenList

An optional Integer value that indicates whether or not Word Pro should add this file's name to list of recently opened files on the File menu. Default is False, indicating that the file name will not be added.

SaveCopyAs

An optional Integer value that indicates whether Word Pro should leave open the original document or the newly saved document. Default is False, indicating that Word Pro should close the original and leave open the newly saved file.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Word Pro: SaveData method

{button ,AL('H_GRAPHICOLEBJECT_CLASS;H_GRAPHIC_CLASS;H_OLEOBJECT_CLASS',0)} [See list of classes](#)

Syntax

[objectreference].SaveData(P1)

Parameters

P1

Data type is Long.

Return value

The return value for this method will always be -1 or 0. When testing the return value, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Word Pro: SaveDivision method

{button ,AL('H_DIVISION_CLASS;H_TEXTDOCUMENT_CLASS',0)} [See list of classes](#)

{button ,AL('H_SAVEDIVISION_METHOD_EXSCRIPT',1)} [See example](#)

Saves changes made to an external division.

Syntax

[objectreference].SaveDivision()

Parameters**Return value**

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

This method only saves divisions that are external to the current document. For example, you can use this method to save changes made to divisions in a master document.

Word Pro: SaveGlossary method

{button ,AL('H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL('H_SAVEGLOSSARY_METHOD_EXSCRIPT',1)} [See example](#)

Saves the currently active Glossary data file. You can save the glossary or you can save a copy of the glossary under a different name or in another file type.

Syntax

[objectreference].SaveGlossary(FilePath, FileType[, AddToLastFileOpenList] [, SaveCopyAs])

Parameters

FilePath

A String expression specifying the path and name for the Glossary file.

FileType

A String expression specifying a file type. Use this to change the file type of the document when you want to use or read it in another application.

AddToLastFileOpenList

Allows you to show or hide the Glossary file from the last file opened list. Data type is Integer. The legal values for this parameter are -1 and 0 but you may use the LotusScript constants True (-1) and False (0). Optional parameter. Default is False, which hides the Glossary file from the list.

SaveCopyAs

A String expression specifying a new name for the Glossary file. Use this to save a copy of the Glossary file under a different name.

Return value

None.

Usage

Word Pro: SaveMacro method

{button ,AL(^H_MACRO_CLASS',0)} [See list of classes](#)

Syntax

[objectreference].SaveMacro()

Parameters

Return value

Usage

Word Pro: SaveSnapshot method

{button ,AL('H_GRAPHICOLEBJECT_CLASS;H_GRAPHIC_CLASS;H_OLEOBJECT_CLASS',0)} [See list of classes](#)

Syntax

[objectreference].SaveSnapshot(P1)

Parameters

P1

Data type is Long.

Return value

The return value for this method will always be -1 or 0. When testing the return value, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Word Pro: SaveThumbnailBitmap method

{button ,AL(^H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

Syntax

[objectreference].SaveThumbnailBitmap([Filename])

Parameters

Filename

An optional String expression that indicates the name of the file.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Word Pro: SaveToStorageComplete method

{button ,AL(^H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

Used to complete the use of the SaveToStorage method.

Syntax

[objectreference].SaveToStorageComplete(plStorage)

Parameters

plStorage

A Numeric expression which specifies the IStorage space to which you want to save the embedded Word Pro object.

A null value indicates that you want to keep using the original IStorage space. Any other value specifies which IStorage you want to switch to once the save is complete. Data type is Long.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Word Pro: SaveToStorage method

{button ,AL(^H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

Saves an embedded Word Pro OLE object to the specified IStorage.

Note OLE is not supported under OS/2.

Syntax

[objectreference].SaveToStorage(pIStorage, FileType, SameStorageAsLoad)

Parameters

pIStorage

A Numeric expression which specifies the IStorage space to which you want to save the embedded Word Pro object. Data type is Long.

FileType

A String expression indicating the type of Word Pro or Ami Pro object being saved. The file types include:

- Lotus Ami Pro
- Lotus Ami Pro 3.x Macro
- Lotus Ami Pro 3.x Styles
- Lotus Word Pro
- Lotus Word Pro SmartMaster

SameStorageAsLoad

Data type is Boolean. Indicates whether the IStorage named in pIStorage is the same as the IStorage space from which Word Pro loaded this Word Pro OLE object.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

When you use this method, you must also call the SaveToStorageComplete method.

Word Pro: SaveUserDefaults method

{button ,AL('H_APPLICATIONWINDOW_CLASS',0)} [See list of classes](#)

{button ,AL('H_SAVEUSERDEFAULTS_METHOD_EXSCRIPT',1)} [See example](#)

Saves the following settings from the Word Pro Preferences dialog box and the Welcome dialog box:

- number of undo levels
- number of recent files
- all markup options
- list of recently used SmartMaster templates that display in the Open dialog box

Syntax

[objectreference].SaveUserDefaults()

Parameters

None.

Return value

The return value for this method will always be -1 or 0. When testing the return value, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Word Pro: SaveVersion method

{button ,AL(^H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL(^H_SAVEVERSION_METHOD_EXSCRIPT',1)} [See example](#)

Saves the selected version as a file. Equivalent to choosing File - Versions and clicking Save as File.

Syntax

[objectreference].SaveVersion(FilePath,Version)

Parameters

FilePath

Data type is String.

Version

Data type is Long.

Return value

Usage

Word Pro: Save method

{button ,AL('H_WPAPPLICATION_CLASS;H_DOCUMENT_CLASS;H_TEXTDOCUMENT_CLASS',0)} [See list of classes](#)

{button ,AL('H_SAVE_METHOD_EXSCRIPT',1)} [See example](#)

Saves the currently active document. Equivalent to choosing File - Save.

Syntax

[objectreference].Save()

Parameters

None.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Word Pro: SelectCell method

{button ,AL('H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL('H_SELECTCELL_METHOD_EXSCRIPT',1)} [See example](#)

Selects the contents of the cell at the insertion point.

Note This method does not work for parallel column cells.

Syntax

[objectreference].SelectCell()

Parameters

None.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Word Pro: SelectColumn method

{button ,AL(^H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL(^H_SELECTCOLUMN_METHOD_EXSCRIPT',1)} [See example](#)

Selects the column at the insertion point. If cells from more than one column are selected, all the columns represented in that selection are selected.

Note This method does not work for parallel columns.

Syntax

[objectreference].SelectColumn()

Parameters

None.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Word Pro: SelectCustomIcon method

{button ,AL('H_ICONBARMANAGER_CLASS',0)} [See list of classes](#)

Selects a custom icon so that you can query or attach properties to it.

Syntax

[objectreference].SelectCustomIcon(GraphicPath, MacroPath)

Parameters

GraphicPath

Data type is String. Required parameter.

MacroPath

Data type is String. Required parameter.

Return value

The return value for this method will always be -1 or 0. When testing the return value, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

You must select an icon using either this method or the SelectStandardIcon method before you can assign functional properties to it.

Word Pro: SelectDoc method

{button ,AL(^H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL(^H_SELECTDOC_METHOD_EXSCRIPT',1)} [See example](#)

Selects the entire active division.

Syntax

[objectreference].SelectDoc()

Parameters

None.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Word Pro: SelectEntireCellRange method

{button ,AL('H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL('H_SELECTENTIRECELLRANGE_METHOD_EXSCRIPT',1)} [See example](#)

Selects the cell itself and the contents in the cell. The insertion point must be in the cell. If more than one cell is selected when you call this method, Word Pro selects all the cells and their contents.

Note This method does not work for parallel columns.

Syntax

[objectreference].SelectEntireCellRange()

Parameters

None.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Word Pro: SelectEntireColumn method

{button ,AL(`H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL(`H_SELECTENTIRECOLUMN_METHOD_EXSCRIPT',1)} [See example](#)

Selects the column that contains the currently active cell. If cells are selected from more than one column, this method selects the entire column for each selected cell.

Note This method does not work for parallel columns.

Syntax

[objectreference].SelectEntireColumn()

Parameters

None.

Return value

None.

Usage

Word Pro: SelectEntirePColCellRange method

{button ,AL('H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL('H_SELECTENTIREPCOLCELLRANGE_METHOD_EXSCRIPT',1)} [See example](#)

Selects both the parallel column cell and its contents. If more than one cell is in the focus, Word Pro selects all the cells and their contents.

Syntax

[objectreference].SelectEntirePColCellRange()

Parameters

None.

Return value

This method returns a value of -1 (True) or 0 (False), indicating that the method succeeded or failed respectively.

Usage

Word Pro: SelectEntirePColColumn method

{button ,AL(^H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL(^H_SELECTENTIREPCOLCOLUMN_METHOD_EXSCRIPT',1)} [See example](#)

Selects the parallel column that has the focus. If text is selected in more than one parallel column, this method selects the entire parallel column for each text selection.

Syntax

[objectreference].SelectEntirePColColumn()

Parameters

None.

Return value

None.

Usage

Word Pro: SelectEntirePColRow method

{button ,AL(^H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL(^H_SELECTENTIREPCOLROW_METHOD_EXSCRIPT',1)} [See example](#)

Selects the parallel column row that has the focus.

Syntax

[objectreference].SelectEntirePColColumn()

Parameters

None.

Return value

None.

Usage

Word Pro: SelectEntirePCol method

{button ,AL('H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL('H_SELECTENTIREPCOL_METHOD_EXSCRIPT',1)} [See example](#)

Selects all the columns and rows in the ParallelColumns object that has the focus. If there is no ParallelColumns object in the focus, this method returns 0, indicating failure.

Syntax

[objectreference].SelectEntirePCol()

Parameters

None.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Word Pro: SelectEntireRow method

{button ,AL(^H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL(^H_SELECTENTIREROW_METHOD_EXSCRIPT',1)} [See example](#)

Selects the table row that has the focus and the contents of that row. If cells from more than one row are selected, Word Pro selects all the rows represented in the selection and their contents.

Syntax

[objectreference].SelectEntireRow()

Parameters

None.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Word Pro: SelectEntireTable method

{button ,AL(^H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL(^H_SELECTENTIRETABLE_METHOD_EXSCRIPT',1)} [See example](#)

Selects the table that has the focus and its contents. Equivalent to choosing Table - Select - Entire Table.

Syntax

[objectreference].SelectEntireTable()

Parameters

None.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Word Pro: SelectParagraph method

{button ,AL(^H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL(^H_SELECTPARAGRAPH_METHOD_EXSCRIPT',1)} [See example](#)

Selects the paragraph in which the insertion point is located.

Syntax

[objectreference].SelectParagraph()

Parameters

None.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Word Pro: SelectPColCell method

{button ,AL(^H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL(^H_SELECTPCOLCELL_METHOD_EXSCRIPT',1)} [See example](#)

Selects the cell in a parallel column in which the insertion point is located.

Syntax

[objectreference].SelectPColCell()

Parameters

None.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Word Pro: SelectRow method

{button ,AL('H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL('H_SELECTROW_METHOD_EXSCRIPT',1)} [See example](#)

Selects the contents of the table row in which the insertion point is located. Equivalent to choosing Table - Select - Row Contents.

Syntax

[objectreference].SelectRow()

Parameters

None.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Word Pro: SelectSection method

{button ,AL('H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL('H_SELECTSECTION_METHOD_EXSCRIPT',1)} [See example](#)

Selects the contents within a specified section of the document. Equivalent to clicking the right mouse button on a section tab and choosing Select Section.

Syntax

[objectreference].SelectSection([SectionName])

Parameters

SectionName

Optional String expression which specifies the name of the section whose contents you want to select. If you do not provide this parameter, Word Pro selects the contents of the section in which the insertion point is located.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Word Pro: SelectSentence method

{button ,AL(^H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL(^H_SELECTSENTENCE_METHOD_EXSCRIPT',1)} [See example](#)

Selects the sentence in which the insertion point is located.

Syntax

[objectreference].SelectSentence()

Parameters

None.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Word Pro: SelectStandardIcon method

{button ,AL('H_ICONBARMANAGER_CLASS',0)} [See list of classes](#)

Selects a standard icon so that you can query or attach properties to it.

Syntax

[objectreference].SelectStandardIcon(MenuID)

Parameters

MenuID

The data type for this parameter is Variant which allows the value of this parameter to be one of the constants listed below or its numeric equivalent (in parentheses). There is no default value.

<u>Value</u>	<u>Effect</u>
LwpMenuCharmenu (&HFA)	
LwpMenuCreateInfobusLink (&H373)	
LwpMenuCreatemenu (&H672)	
LwpMenuCustomend (&HCFFF)	
LwpMenuCustomstart (&HC000)	
LwpMenuDebug (&H7F6)	
LwpMenuEditmenu (&H96)	
LwpMenuEnvAddress (&H67D)	
LwpMenuEnvPrint (&H682)	
LwpMenuEnvSave (&H681)	
LwpMenuFcsDeletecolumnmenu (&H32)	
LwpMenuFcsDeletenotemenu (&H34)	
LwpMenuFcsDeletepagebreakmenu (&H30)	
LwpMenuFcsDeletepagelayoutmenu (&H33)	
LwpMenuFcsDeletesectionmenu (&H31)	
LwpMenuFcsDivisionmenu (&H6)	
LwpMenuFcsFootermenu (&HB)	
LwpMenuFcsFootnotetextmenu (&H28)	
LwpMenuFcsFramegraphicmenu (&H35)	
LwpMenuFcsFramemenu (&H1)	
LwpMenuFcsFramemenudropcap (&H2E)	
LwpMenuFcsFrametextmenu (&H2)	
LwpMenuFcsFrametextmenudropcap (&H2D)	
LwpMenuFcsHeadermenu (&HA)	
LwpMenuFcsIndextextmenu (&H29)	
LwpMenuFcsNotemenu (&HD)	
LwpMenuFcsOledivisionmenu (&HF)	
LwpMenuFcsOutlinemenu (&HE)	
LwpMenuFcsParallelcolgraphicmenu (&H37)	
LwpMenuFcsParallelcolmenu (&H9)	
LwpMenuFcsParallelcoltextmenu (&H5)	
LwpMenuFcsPowerfieldmenu (&H4)	
LwpMenuFcsRubymenu (&H14)	
LwpMenuFcsRulermenu (&HC)	
LwpMenuFcsSectionmenu (&H7)	

LwpMenuFcsTablecolumnmenu (&H12)
LwpMenuFcsTablecornermenu (&H10)
LwpMenuFcsTablecornermenutoc (&H2B)
LwpMenuFcsTablegraphicmenu (&H36)
LwpMenuFcsTablemenu (&H8)
LwpMenuFcsTablerowmenu (&H11)
LwpMenuFcsTabletextmenu (&H3)
LwpMenuFcsTabletextmenutoc (&H2C)
LwpMenuFcsTextdatemenu (&H2F)
LwpMenuFcsTextmenu (&H0)
LwpMenuFcsToctextmenu (&H2A)
LwpMenuFcsVertrulermenu (&H13)
LwpMenuFilemenu (&H64)
LwpMenuFirstpopupid (&H39D)
LwpMenuFramemenu (&H190)
LwpMenuHelpmenu (&H226)
LwpMenuIbShowfrommenu (&H7AB)
LwpMenuMafidHelp (&H384)
LwpMenuMafidNotFound (&H385)
LwpMenuMain (&H7FB)
LwpMenuMaNotesflow (&H3E81)
LwpMenuMaNotesflowmax (&H3F48)
LwpMenuMcAddpicture (&H372)
LwpMenuMcAlignment (&H107)
LwpMenuMcAllcaps (&H111)
LwpMenuMcAlllower (&H11F)
LwpMenuMcBold (&HFE)
LwpMenuMcBulletedpanel (&H11D)
LwpMenuMcCaps (&H108)
LwpMenuMcCenter (&H104)
LwpMenuMcChart (&H1A9)
LwpMenuMcClickhereglossary (&H7E1)
LwpMenuMcClickherekeyword (&H7D4)
LwpMenuMcClosegraphicole (&H78A)
LwpMenuMcCommenttools (&H37E)
LwpMenuMcCommenttoolsOn (&H379)
LwpMenuMcCreateobjectchart (&H7D1)
LwpMenuMcCreateobjectcomment (&H7CD)
LwpMenuMcCreateobjectdraw (&H7CE)
LwpMenuMcCreateobjectschedule (&H7CF)
LwpMenuMcCreateobjectspreadsheet
(&H7D0)
LwpMenuMcCreateobjectviewer (&H7CC)
LwpMenuMcDooledragdrop (&H44E)
LwpMenuMcDoubleunder (&H10E)
LwpMenuMcDrawinfoxmouse (&H263)

LwpMenuMcDrawing (&H1AA)
LwpMenuMcEnargetext (&HFC)
LwpMenuMcEquation (&H1AB)
LwpMenuMcExtract (&HF6)
LwpMenuMcExtracton (&H346)
LwpMenuMcFastformat (&H123)
LwpMenuMcFont (&HFB)
LwpMenuMcFontrevert (&HF3)
LwpMenuMcHighlight (&H126)
LwpMenuMcHighnote (&H37F)
LwpMenuMcIndent (&H102)
LwpMenuMcIndentall (&HF7)
LwpMenuMcIndentfirst (&HF8)
LwpMenuMcIndentrest (&HF9)
LwpMenuMcInnitcaps (&H120)
LwpMenuMcInsertdefbullet (&H7CA)
LwpMenuMcInsertdefnumber (&H7C9)
LwpMenuMcInternettools (&H7BB)
LwpMenuMcInternettoolson (&H7BA)
LwpMenuMcItalic (&HFF)
LwpMenuMcJustfont (&H116)
LwpMenuMcJustify (&H106)
LwpMenuMcJustifyall (&H789)
LwpMenuMcJustpointsize (&H117)
LwpMenuMcLeft (&H103)
LwpMenuMcMacrofastformat (&HEE)
LwpMenuMcNextalignment (&H3B7)
LwpMenuMcNextattribute (&H3B8)
LwpMenuMcNextbullet (&H3B9)
LwpMenuMcNextfacename (&H3B6)
LwpMenuMcNextnumbering (&H3BA)
LwpMenuMcNextstyle (&H3BB)
LwpMenuMcNormal (&HFD)
LwpMenuMcOutdent (&H11C)
LwpMenuMcOutlineassistant (&H674)
LwpMenuMcReductext (&HF5)
LwpMenuMcRight (&H105)
LwpMenuMcSmallcaps (&H121)
LwpMenuMcStrikethru (&H118)
LwpMenuMcSubscript (&H110)
LwpMenuMcSuperscript (&H10F)
LwpMenuMcTextinfo box (&H11B)
LwpMenuMcTextinfo box mouse (&H262)
LwpMenuMcTextlinepanel (&H688)
LwpMenuMcTextstyleinfo box (&H845)

LwpMenuMcToggleskipbullet (&H77F)
LwpMenuMcUnderline (&H100)
LwpMenuMcUpdateindex (&H7C3)
LwpMenuMcUpdatetoc (&H7C4)
LwpMenuMcWebauthorbegin (&H7DD)
LwpMenuMcWebauthorend (&H7E8)
LwpMenuMcWebcreateformcontrol (&H7E0)
LwpMenuMcWebcreatehr (&H7DE)
LwpMenuMcWebcreatelink (&H7DD)
LwpMenuMcWebcreatewallpaper (&H7DF)
LwpMenuMcWordunder (&H101)
LwpMenuMdDebugAtom (&H3FC)
LwpMenuMdDebugAtomCount (&H420)
LwpMenuMdDebugAtomWatch (&H41F)
LwpMenuMdDebugBack (&H415)
LwpMenuMdDebugCheckpoint (&H40F)
LwpMenuMdDebugChint (&H402)
LwpMenuMdDebugCol (&H3ED)
LwpMenuMdDebugCont (&H3EC)
LwpMenuMdDebugContent (&H40C)
LwpMenuMdDebugCounts (&H405)
LwpMenuMdDebugDemandload (&H412)
LwpMenuMdDebugDialog (&H414)
LwpMenuMdDebugDivinfo (&H3EB)
LwpMenuMdDebugDivision (&H3F4)
LwpMenuMdDebugDoc (&H3E9)
LwpMenuMdDebugDoclay (&H3F8)
LwpMenuMdDebugDwrdarg (&H3FE)
LwpMenuMdDebugEnd (&H44B)
LwpMenuMdDebugEnumobjhead (&H419)
LwpMenuMdDebugFnlistlist (&H424)
LwpMenuMdDebugFontmgr (&H41D)
LwpMenuMdDebugFootnote (&H425)
LwpMenuMdDebugFoundry (&H404)
LwpMenuMdDebugger (&H3E8)
LwpMenuMdDebugHeadtailholder (&H416)
LwpMenuMdDebugHeap (&H3F0)
LwpMenuMdDebugInfo (&H3FB)
LwpMenuMdDebugLay (&H3F5)
LwpMenuMdDebugLine (&H3EE)
LwpMenuMdDebugMain (&H3F1)
LwpMenuMdDebugMemory (&H40A)
LwpMenuMdDebugMemorydump (&H40B)
LwpMenuMdDebugMenuset (&H426)
LwpMenuMdDebugMeter (&H407)

LwpMenuMdDebugMsgtrace (&H403)
LwpMenuMdDebugNamedobjectholder (&H41B)
LwpMenuMdDebugNotify (&H41A)
LwpMenuMdDebugNotifypersistent (&H417)
LwpMenuMdDebugNumberhint (&H413)
LwpMenuMdDebugObjectholder (&H418)
LwpMenuMdDebugObjectSizes (&H422)
LwpMenuMdDebugPara (&H3FA)
LwpMenuMdDebugPhint (&H401)
LwpMenuMdDebugPieceCount (&H421)
LwpMenuMdDebugPres (&H3EA)
LwpMenuMdDebugPresinfo (&H406)
LwpMenuMdDebugProf (&H3F2)
LwpMenuMdDebugPtrholder (&H409)
LwpMenuMdDebugSanity (&H41E)
LwpMenuMdDebugSection (&H41C)
LwpMenuMdDebugSfb (&H411)
LwpMenuMdDebugSilver (&H40D)
LwpMenuMdDebugSline (&H400)
LwpMenuMdDebugSocket (&H410)
LwpMenuMdDebugStack (&H408)
LwpMenuMdDebugStrarg (&H3FF)
LwpMenuMdDebugTextstyle (&H423)
LwpMenuMdDebugVcnt (&H3F3)
LwpMenuMdDebugVerctrl (&H3F7)
LwpMenuMdDebugVermgr (&H3F6)
LwpMenuMdDebugVernode (&H40E)
LwpMenuMdDebugVobj (&H3EF)
LwpMenuMdDebugWrdarg (&H3FD)
LwpMenuMdivCollapseddivision (&H2BE)
LwpMenuMdivCombineddivisions (&H34E)
LwpMenuMdivCopydivision (&H2C3)
LwpMenuMdivCutdivision (&H2C2)
LwpMenuMdivDeleteddivision (&H34F)
LwpMenuMdivDivisionproperties (&H34C)
LwpMenuMdivExpanddivision (&H350)
LwpMenuMdivGroupdivision (&H356)
LwpMenuMdivNewdivision (&H34D)
LwpMenuMdivPasteddivision (&H2C4)
LwpMenuMeAdddocdescvar (&HA7)
LwpMenuMeAddmergevar (&HA6)
LwpMenuMeAddnote (&HB5)
LwpMenuMeBullet (&H344)
LwpMenuMeChangelink (&H24E)
LwpMenuMeCopy (&H9A)

LwpMenuMeCut (&H99)
LwpMenuMeDefaults (&HA2)
LwpMenuMeDefback (&HB0)
LwpMenuMeDefpath (&HAE)
LwpMenuMeDefpaths (&HAA)
LwpMenuMeDefpathsnw (&H324)
LwpMenuMeDefsty (&HAF)
LwpMenuMeDelete (&HB3)
LwpMenuMeDescvar (&HB7)
LwpMenuMeDcoptions (&HAD)
LwpMenuMeDropcaps (&H7B7)
LwpMenuMeFootnoteinsert (&HB1)
LwpMenuMeFootnotes (&HA0)
LwpMenuMeFootopts (&HAB)
LwpMenuMeFormatcheckbar (&HB8)
LwpMenuMeGetglosname (&HC6)
LwpMenuMeIndexentry (&HB4)
LwpMenuMeInsertdate (&HA5)
LwpMenuMeInsertnote (&HB2)
LwpMenuMeInsertobject (&H24D)
LwpMenuMeInsertsymbol (&H7B5)
LwpMenuMeInsvariable (&H9E)
LwpMenuMeMacpath (&H11E)
LwpMenuMeMenuedit (&H7FC)
LwpMenuMeMergedatafile (&H232)
LwpMenuMeMergeextdesc (&H234)
LwpMenuMeMergevar (&HB6)
LwpMenuMeMergevarsdatafile (&H236)
LwpMenuMeNbDefaults (&HA4)
LwpMenuMeNohyphen (&HBD)
LwpMenuMeNotes (&HA1)
LwpMenuMeNotes20 (&H11A)
LwpMenuMenualternate (&H805)
LwpMenuMePagedown (&HBA)
LwpMenuMePageup (&HB9)
LwpMenuMePaste (&H9B)
LwpMenuMePastelinks (&H9D)
LwpMenuMePastespecial (&H24C)
LwpMenuMeProtected (&HBC)
LwpMenuMeRedo (&HAC)
LwpMenuMeRevinsertion (&HBB)
LwpMenuMeRevisionmode (&H257)
LwpMenuMergeScriptNotify (&H38E)
LwpMenuMeSeclevaction (&HA9)
LwpMenuMeSelectdivision (&H2C0)

LwpMenuMeSelectfile (&H256)
LwpMenuMeSelectparagraph (&H255)
LwpMenuMeSelectsentence (&H254)
LwpMenuMeSelectword (&H253)
LwpMenuMeSmartcorrect (&H5E1)
LwpMenuMeTocassist (&H684)
LwpMenuMeTocentry (&H343)
LwpMenuMeTogglesmartselect (&H7CB)
LwpMenuMeTypeover (&HA3)
LwpMenuMeUndo (&H97)
LwpMenuMeUndoredo (&H9C)
LwpMenuMeUseworkpath (&H24F)
LwpMenuMeViewhideallmarks (&H775)
LwpMenuMeViewmarginguides (&H772)
LwpMenuMeViewpagegauge (&H773)
LwpMenuMeViewpcolguides (&H771)
LwpMenuMeViewshowallmarks (&H774)
LwpMenuMeViewtablegrid (&H76F)
LwpMenuMeViewtableheading (&H770)
LwpMenuMeWordcount (&H7B6)
LwpMenuMfAcquiretwinimage (&H841)
LwpMenuMfAllfidsrename (&H76)
LwpMenuMfAppendtext (&H6A)
LwpMenuMfAsciiopts (&H7B)
LwpMenuMfAttribnw (&H8F)
LwpMenuMfCanmergeprint (&H92)
LwpMenuMfCanprint (&H7C)
LwpMenuMfChgprinter (&H70)
LwpMenuMfClose (&H90)
LwpMenuMfCloseall (&H2DB)
LwpMenuMfCloseallreplacelast (&H2DC)
LwpMenuMfCloseandreturn (&H2BF)
LwpMenuMfClosefile (&H93)
LwpMenuMfCreatedatafile (&H6E)
LwpMenuMfCreatedesc (&H30F)
LwpMenuMfCreatedivfromselectedtext (&H840)
LwpMenuMfCreatedivision (&H683)
LwpMenuMfCreatemerge (&H311)
LwpMenuMfCreateoledivision (&H77D)
LwpMenuMfCreatetabdivision (&H686)
LwpMenuMfCreateversion (&H368)
LwpMenuMfCreateversionDivision (&H389)
LwpMenuMfDde (&H73)
LwpMenuMfDocdesc (&H6D)
LwpMenuMfDocdescExternal (&H25E)

LwpMenuMfDocdescnw (&H81)
LwpMenuMfDummymrgext (&H7A)
LwpMenuMfDummymrgsel (&H79)
LwpMenuMfEditdatafile (&H313)
LwpMenuMfEditorgreet (&H2DD)
LwpMenuMfEnvelopeprint (&H112)
LwpMenuMfExit (&H71)
LwpMenuMfExitandreturn (&H2C1)
LwpMenuMfExport (&H8E)
LwpMenuMfExternalfileexit (&H79E)
LwpMenuMfExternalfilesavedivision (&H79F)
LwpMenuMfFilemanage (&H6C)
LwpMenuMfFilenewbrowse (&H833)
LwpMenuMfFilesep (&HEB)
LwpMenuMfFldrename (&H7D)
LwpMenuMfFramerevert (&H78B)
LwpMenuMfFtpconvert (&H7A2)
LwpMenuMfFtpexternalize (&H7A4)
LwpMenuMfFtpinsert (&H7A1)
LwpMenuMfFtpopen (&H7A0)
LwpMenuMfFtpopendivision (&H7A9)
LwpMenuMfFtpopenurl (&H7AA)
LwpMenuMfFtpopenurlinbrowser (&H830)
LwpMenuMfFtpoptions (&H7A6)
LwpMenuMfFtpsaveas (&H7A3)
LwpMenuMfHidewebauthor (&H7DC)
LwpMenuMfHtmlassist (&H82F)
LwpMenuMfHtmloptions (&H7C2)
LwpMenuMfImport (&H6B)
LwpMenuMfImportdraw (&H335)
LwpMenuMfImportnw (&H80)
LwpMenuMfImportpicture (&H8D)
LwpMenuMfImporttext (&H98)
LwpMenuMfJustprint (&H75)
LwpMenuMfLabeldlg (&H7EF)
LwpMenuMfLastopen1 (&H85)
LwpMenuMfLastopen2 (&H86)
LwpMenuMfLastopen3 (&H87)
LwpMenuMfLastopen4 (&H88)
LwpMenuMfLastopen5 (&H89)
LwpMenuMfLastopenplaceholder (&H82E)
LwpMenuMfLock (&H94)
LwpMenuMfMail (&HE5)
LwpMenuMfMailEditroute (&H387)
LwpMenuMfMailNew (&H2E1)

LwpMenuMfMailNextstop (&H386)
LwpMenuMfMailread (&HE6)
LwpMenuMfMailSendmessage (&H7C8)
LwpMenuMfMasterdocument (&H2E0)
LwpMenuMfMasternw (&H82)
LwpMenuMfMergeaction (&H74)
LwpMenuMfMergelabels (&H8B)
LwpMenuMfMergeopen (&H31A)
LwpMenuMfMergeprintall (&H319)
LwpMenuMfMergeprintopts (&H235)
LwpMenuMfMergeviewprint (&H84)
LwpMenuMfMrgdelimit (&H30C)
LwpMenuMfMrgenvelope (&H312)
LwpMenuMfMrgenvelopesetup (&H261)
LwpMenuMfMrggotorecord (&H30E)
LwpMenuMfMrglabel (&H25F)
LwpMenuMfMrgletter (&H30D)
LwpMenuMfMrglettersetup (&H260)
LwpMenuMfMrgsort (&H310)
LwpMenuMfMrgviewprint (&H264)
LwpMenuMfMrgwelcome (&H7C1)
LwpMenuMfNew (&H65)
LwpMenuMfNewanytab (&H69D)
LwpMenuMfNewdivision (&H2DE)
LwpMenuMfNewpreview (&H317)
LwpMenuMfNewtabdivision (&H78E)
LwpMenuMfNextmdiwindow (&H2D7)
LwpMenuMfNosopen (&H7C6)
LwpMenuMfNossaveas (&H7C7)
LwpMenuMfNwimport (&H83)
LwpMenuMfOdmaconvert (&H6A0)
LwpMenuMfOdmaexport (&H6A3)
LwpMenuMfOdmaexternalize (&H6A2)
LwpMenuMfOdmaimportpicture (&H7B8)
LwpMenuMfOdmainsert (&H69F)
LwpMenuMfOdmaopen (&H69E)
LwpMenuMfOdmasaveas (&H6A1)
LwpMenuMfOdmashowattributes (&H7A7)
LwpMenuMfOpen (&H66)
LwpMenuMfOpenandeditdatafile (&H315)
LwpMenuMfOpendivision (&H2DF)
LwpMenuMfOpennostyle (&H8A)
LwpMenuMfOpenpreview (&H316)
LwpMenuMfOpentabdivision (&H78D)
LwpMenuMfOtherflds (&H8C)

LwpMenuMfPassword (&H91)
LwpMenuMfPassword2 (&H122)
LwpMenuMfPlainnew (&H784)
LwpMenuMfPrint (&H6F)
LwpMenuMfPrintcancel (&H78)
LwpMenuMfPrintdialonly (&H44F)
LwpMenuMfPrintenvelope (&HED)
LwpMenuMfPrintopt (&H7E)
LwpMenuMfPrintreset (&H72)
LwpMenuMfPrintwrongpaper (&H77)
LwpMenuMfPublishweb (&H7D5)
LwpMenuMfQuickopen (&HE7)
LwpMenuMfQuickprint (&H3B2)
LwpMenuMfRevert (&H69)
LwpMenuMfSaContinueRoute (&H2E2)
LwpMenuMfSaRoute (&H2E3)
LwpMenuMfSave (&H67)
LwpMenuMfSaveas (&H68)
LwpMenuMfSaveasnw (&H7F)
LwpMenuMfSavecopyas (&H124)
LwpMenuMfSeldatafile (&H30C)
LwpMenuMfSelecttwainsource (&H843)
LwpMenuMfSharedoccontrol (&H25A)
LwpMenuMfShowwebauthor (&H7DB)
LwpMenuMfStandalonep (&H147)
LwpMenuMfToggleclickhereprompts (&H77E)
LwpMenuMfToggleinfo box (&H367)
LwpMenuMfUpdateobject (&H125)
LwpMenuMfUsecurasmrgdoc (&H314)
LwpMenuMfWelcome (&H7AD)
LwpMenuMh2000 (&H22C)
LwpMenuMhAbout (&H228)
LwpMenuMhCompatible (&H22B)
LwpMenuMhDoHelp (&H341)
LwpMenuMhForUpgraders (&H342)
LwpMenuMhHowDol (&H340)
LwpMenuMhIdxhelp (&H227)
LwpMenuMhInternetsearch (&H7D2)
LwpMenuMhKeyboard (&H33F)
LwpMenuMhLotusftpsite (&H7BE)
LwpMenuMhLotushomepage (&H7BC)
LwpMenuMhLotussmartsuite (&H7D3)
LwpMenuMhLotussupport (&H7BD)
LwpMenuMhMacrohelp (&H229)
LwpMenuMhSearch (&H36F)

LwpMenuMhTour (&H77B)
LwpMenuMhTutorial (&H345)
LwpMenuMhUsingHelp (&H33E)
LwpMenuMkPagedown (&H44D)
LwpMenuMkPageup (&H44C)
LwpMenuMIAddtouserdictionary (&H782)
LwpMenuMIAssignmacro (&H1CF)
LwpMenuMIBookmark (&H1D2)
LwpMenuMICall (&H1ED)
LwpMenuMIChglang (&H1E1)
LwpMenuMICompilemacro (&H5DD)
LwpMenuMICrossreference (&H7D6)
LwpMenuMICyclekeysetup (&H207)
LwpMenuMIDoccompare (&H1F4)
LwpMenuMIDoccontrol (&H258)
LwpMenuMIDoccontrolEditorpanel (&H25B)
LwpMenuMIDoccontrolExternal (&H25C)
LwpMenuMIExechand (&H23E)
LwpMenuMIField (&H238)
LwpMenuMIFieldadd (&H23C)
LwpMenuMIFieldauto (&HE0)
LwpMenuMIFielddoauto (&HE2)
LwpMenuMIFieldeval (&H23B)
LwpMenuMIFieldhide (&H33D)
LwpMenuMIFieldlock (&H10C)
LwpMenuMIFieldnext (&H239)
LwpMenuMIFieldprev (&H23A)
LwpMenuMIFieldremove (&HE3)
LwpMenuMIFieldsave (&H10D)
LwpMenuMIFieldshowrt (&HE1)
LwpMenuMIFieldtog (&H10A)
LwpMenuMIFieldupall (&H10B)
LwpMenuMIFieldupdate (&H23D)
LwpMenuMIGenerateindex (&H1D6)
LwpMenuMIGeneratetoc (&H1D1)
LwpMenuMIGlossary (&H1D3)
LwpMenuMIGlosset (&H1D4)
LwpMenuMIGoto (&H1C4)
LwpMenuMIGotolast (&H1CA)
LwpMenuMIGrammar (&H1FB)
LwpMenuMIGrammarcancel (&H201)
LwpMenuMIGrammarchange (&H204)
LwpMenuMIGrammarinit (&H1FE)
LwpMenuMIGrammarnext (&H200)
LwpMenuMIGrammaroptionstext (&H203)

LwpMenuMIGrammaropts (&H1FC)
LwpMenuMIGrammarstart (&H1FD)
LwpMenuMIGrammarstats (&H205)
LwpMenuMIGrammarsugtext (&H202)
LwpMenuMIIconpath (&H206)
LwpMenuMIIndexnextfile (&H119)
LwpMenuMIIndexopts (&HE4)
LwpMenuMIInsertreference (&H271)
LwpMenuMIMacassign (&H1EA)
LwpMenuMIMacdde (&H1EB)
LwpMenuMIMacpause (&H1D5)
LwpMenuMIMacplay (&H1E9)
LwpMenuMIMacresume (&H22A)
LwpMenuMIMacroaccel (&H1A6)
LwpMenuMIMacroappend (&H5DF)
LwpMenuMIMacroedit (&H1F5)
LwpMenuMIMacroendrecord (&H336)
LwpMenuMIMacrooptions (&H1F8)
LwpMenuMIMacroplay (&H1F6)
LwpMenuMIMacrorecord (&H1F7)
LwpMenuMIMacrorun (&H5DE)
LwpMenuMIMacros (&H1D0)
LwpMenuMIMactype (&H1EC)
LwpMenuMIModcreatorule (&H1FF)
LwpMenuMINewspell (&H209)
LwpMenuMIPrototype (&H5E0)
LwpMenuMIQuickendrecord (&H33A)
LwpMenuMIQuickgoto (&H7F0)
LwpMenuMIQuickplay (&H1FA)
LwpMenuMIQuickrec (&H1F9)
LwpMenuMIRenumnotes (&H95)
LwpMenuMIRenumseq (&H1E2)
LwpMenuMIReplace (&H1E0)
LwpMenuMIRepopts (&H1E6)
LwpMenuMIRevaccept (&H36D)
LwpMenuMIRevacceptall (&H36B)
LwpMenuMIRevcancel (&H36E)
LwpMenuMIRevcancelall (&H36C)
LwpMenuMIReviewrevs (&H22D)
LwpMenuMIRevisionbar (&H20A)
LwpMenuMIRevmarkingoff (&H33C)
LwpMenuMIRevmarkoptions (&H1CD)
LwpMenuMISammy2 (&H1EE)
LwpMenuMIsAutoapp (&H369)
LwpMenuMIsCompile (&H365)

LwpMenuMIsCreatedialog (&H360)
LwpMenuMIsCreatescript (&H35F)
LwpMenuMISearch (&H1C3)
LwpMenuMISearchonly (&H1E8)
LwpMenuMISetmacrofile (&H5DC)
LwpMenuMISIdeBreakpoints (&H3C9)
LwpMenuMISIdeBrowser (&H3CA)
LwpMenuMISIdeCheckAll (&H3DC)
LwpMenuMISIdeCheckScript (&H3CD)
LwpMenuMISIdeClear (&H3DE)
LwpMenuMISIdeClearAll (&H3C7)
LwpMenuMISIdeCloseWin (&H3D6)
LwpMenuMISIdeCodeWindow (&H3DB)
LwpMenuMISIdeContinue (&H3C3)
LwpMenuMISIdeDebug (&H3D0)
LwpMenuMISIdeDeselectAll (&H3BE)
LwpMenuMISIdeDisAll (&H3C8)
LwpMenuMISIdeDisBp (&H3C6)
LwpMenuMISIdeEditClearAll (&H3BC)
LwpMenuMISIdeExport (&H3D4)
LwpMenuMISIdeFindnext (&H3DF)
LwpMenuMISIdeHelpAbout (&H3E5)
LwpMenuMISIdeHelpLs (&H3E2)
LwpMenuMISIdeHelpSe (&H3E4)
LwpMenuMISIdeHelpWp (&H3E3)
LwpMenuMISIdeImport (&H3D3)
LwpMenuMISIdeInsertLso (&H3E6)
LwpMenuMISIdeNewFun (&H3CF)
LwpMenuMISIdeNewSub (&H3CE)
LwpMenuMISIdeNextSub (&H3D9)
LwpMenuMISIdeOutput (&H3CB)
LwpMenuMISIdePrevSub (&H3D8)
LwpMenuMISIdeRedo (&H3D1)
LwpMenuMISIdeRuncursub (&H3E1)
LwpMenuMISIdeSaveLso (&H3DD)
LwpMenuMISIdeScriptPref (&H3D5)
LwpMenuMISIdeSelectAll (&H3BD)
LwpMenuMISIdeSetBp (&H3C5)
LwpMenuMISIdeShowicon (&H3E0)
LwpMenuMISIdeStepExit (&H3C2)
LwpMenuMISIdeStepIn (&H3C0)
LwpMenuMISIdeStepOver (&H3C1)
LwpMenuMISIdeStop (&H3C4)
LwpMenuMISIdeToggle (&H3D7)
LwpMenuMISIdeTogglebp (&H3DA)

LwpMenuMIsIdeVariables (&H3CC)
LwpMenuMISkipallmisspelledword (&H781)
LwpMenuMISkipmisspelledword (&H780)
LwpMenuMISmartfill (&H208)
LwpMenuMISort (&H1C8)
LwpMenuMISpell (&H1DB)
LwpMenuMISpelladddict (&H1D7)
LwpMenuMISpellcancel (&H1DE)
LwpMenuMISpellcheck (&H1DD)
LwpMenuMISpellinit (&H1DC)
LwpMenuMISpellopts (&H1C5)
LwpMenuMISpellopts2 (&H1CB)
LwpMenuMISpellreplace (&H1DA)
LwpMenuMISpellskip (&H1DF)
LwpMenuMISpellskipall (&H1D8)
LwpMenuMISratrs (&H1E5)
LwpMenuMISrcancel (&H1F2)
LwpMenuMISrfindmatch (&H1F0)
LwpMenuMISrinit (&H1F3)
LwpMenuMISropts (&H1E7)
LwpMenuMISrreplacethenfind (&H1F1)
LwpMenuMIsRunscript (&H31B)
LwpMenuMIsShowide (&H3D2)
LwpMenuMITables (&H1C9)
LwpMenuMIThesaurus (&H1C7)
LwpMenuMIToa (&H20B)
LwpMenuMIToc (&H1CE)
LwpMenuMITocopts (&H1EF)
LwpMenuMITogglebubblehelp (&H783)
LwpMenuMIUpdatealldates (&H240)
LwpMenuMIUpdatedate (&H23F)
LwpMenuMIUserdict (&H1C6)
LwpMenuMMailmemo (&H64A)
LwpMenuMmCaps (&H644)
LwpMenuMmCascades (&H640)
LwpMenuMmFields (&H641)
LwpMenuMmMacros (&H647)
LwpMenuMmOtherframe (&H651)
LwpMenuMmOtherstuff (&H655)
LwpMenuMmOthertable (&H652)
LwpMenuMmShowoptions (&H64E)
LwpMenuMnCloseall (&H380)
LwpMenuMnClosenote (&H37B)
LwpMenuMnDeleteall (&H382)
LwpMenuMnDeletenote (&H37A)

LwpMenuMnNextnote (&H37C)
LwpMenuMnNumbernotes (&H384)
LwpMenuMnOpenall (&H381)
LwpMenuMnPrevnote (&H37D)
LwpMenuMnShowinitials (&H383)
LwpMenuMoApplstyle (&H68A)
LwpMenuMoAttribnw (&HC3)
LwpMenuMoCreatenw (&HC0)
LwpMenuMoHideinfobox (&H7AC)
LwpMenuMolContract (&H2F4)
LwpMenuMolContractsingle (&H2F7)
LwpMenuMolDemote (&H2F0)
LwpMenuMolExpand (&H2F3)
LwpMenuMolExpandsingle (&H2F6)
LwpMenuMolLinksnw (&HC2)
LwpMenuMolListnw (&HC1)
LwpMenuMolMovedown (&H2F2)
LwpMenuMolMoveup (&H2F1)
LwpMenuMolOutstylesequence (&H2ED)
LwpMenuMolPromote (&H2EF)
LwpMenuMolSetshowlevel1 (&H2F9)
LwpMenuMolSetshowlevel10 (&H302)
LwpMenuMolSetshowlevel2 (&H2FA)
LwpMenuMolSetshowlevel3 (&H2FB)
LwpMenuMolSetshowlevel4 (&H2FC)
LwpMenuMolSetshowlevel5 (&H2FD)
LwpMenuMolSetshowlevel6 (&H2FE)
LwpMenuMolSetshowlevel7 (&H2FF)
LwpMenuMolSetshowlevel8 (&H300)
LwpMenuMolSetshowlevel9 (&H301)
LwpMenuMolShowlevels (&H2F5)
LwpMenuMolUseoutlinestyle (&H2F8)
LwpMenuMoOpennw (&HBF)
LwpMenuMpColumnbreak (&H76E)
LwpMenuMpCreate (&H17E)
LwpMenuMpCreatelayout (&H15F)
LwpMenuMpDellayout (&H175)
LwpMenuMpEditfooter (&H802)
LwpMenuMpEditheader (&H801)

Return value

The return value for this method will always be -1 or 0. When testing the return value, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

You must select an icon using either this method or the SelectCustomIcon method before you can assign functional properties to it.

Word Pro: SelectTableItem method

{button ,AL(^H_BASETAble_CLASS;H_FOOTNOTETAble_CLASS;H_PARALLELCOLUMNS_CLASS;H_GLOSSARY_CLASS;H_TABLEHEADING_CLASS;H_TABLE_CLASS',0)} [See list of classes](#)

{button ,AL(^H_SELECTTABLEITEM_METHOD_EXSCRIPT',1)} [See example](#)

Allows you to select a specific type of table item. In order to use this method, the insertion point must be inside the table that contains the items to be selected.

Syntax

[objectreference].SelectTableItem(TableSelection, [p2,] [p3,] [p4,] [EndCol,] [SelectWholeCell])

Parameters

TableSelection

Data type is Variant, which allows its value to be one of the string constants below or its numeric equivalent. This parameter specifies which type of table item to select.

<u>Value</u>	<u>Effect</u>
\$LwpTableSelectionCell (1899)	Moves the insertion point into a specified cell.
\$LwpTableSelectionRow (1900)	Selects a specified range of rows.
\$LwpTableSelectionColumn (1901)	Selects a specified range of columns.
\$LwpTableSelectionTable (1902)	Selects the entire table.
\$LwpTableSelectionRange (1903)	Selects a specified range of cells.

p2

Optional Integer parameter. The purpose of this parameter varies according to the TableSelection parameter value. For details on each TableSelection parameter value, see the Usage section.

p3

Optional Integer parameter. The purpose of this parameter varies according to the TableSelection parameter value. For details on each TableSelection parameter value, see the Usage section.

p4

Optional Integer parameter. The purpose of this parameter varies according to the TableSelection parameter value. For details on each TableSelection parameter value, see the Usage section.

EndCol

Optional Integer parameter. The purpose of this parameter varies according to the TableSelection parameter value. For details on each TableSelection parameter value, see the Usage section.

SelectWholeCell

Optional Integer parameter. The purpose of this parameter varies according to the TableSelection parameter value. For details on each TableSelection parameter value, see the Usage section.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

\$LwpTableSelectionCell (1899)

When you use this value in the TableSelection parameter, the method moves the insertion point into a specific cell. You must also provide values for the P2 and P3 parameters.

P2

Optional Integer value which represents the row ID of the cell into which the insertion point will move. There is no default value.

P3

Optional Integer value which represents the column ID of the cell into which the insertion point will move. There is no default value.

None of the other parameters are used when you use this value for the TableSelection parameter.

The row ID and column ID are zero based values. In order to select the first cell in a table, use a row ID value of 0 and a column ID value of 0.

\$LwpTableSelectionRow (1900)

When you use this value in the TableSelection parameter, the method selects a range of rows within a table. You can also provide values for the P2, P3, and P4 parameters. When you do not supply a value for any other parameter, the contents of the cells within the current row are selected.

P2

Optional Integer value which represents the ID of the first row to be included in the selection. Default is 0.

P3

Optional Integer value which represents the ID of the last row to be included in the selection. Default is 0.

P4

A Boolean value which specifies whether whole cells should be selected. Default is 0.

None of the other parameters are used when you use this value for the TableSelection parameter.

The row ID is a zero based value. In order to select the first row of a table, use a row ID value of 0.

\$LwpTableSelectionColumn (1901)

When you use this value in the TableSelection parameter, the method selects a range of columns within a table. You can also provide values for the P2, P3, and P4 parameters. When you do not supply a value for any other parameter, the contents of the cells within the current column are selected.

P2

Optional Integer value which represents the ID of the first column to be included in the selection. Default is 0.

P3

Optional Integer value which represents the ID of the last column to be included in the selection. Default is 0.

P4

A Boolean value which specifies whether whole cells should be selected. Default is 0.

None of the other parameters are used when you use this value for the TableSelection parameter.

The column ID is a zero based value. In order to select the first column of a table, use a column ID value of 0.

\$LwpTableSelectionTable (1902)

When you use this value in the TableSelection parameter, the method selects an entire table. You can also provide a value for the P2 parameter.

P2

An optional Boolean value that specifies whether whole cells should be selected. Default is 0.

None of the other parameters are used when you use this value for the TableSelection parameter.

\$LwpTableSelectionRange (1903)

When you use this value in the TableSelection parameter, the method selects a range of cells within a table. You can also provide values for the P2, P3, P4, EndCol, and SelectWholeCell parameters. When you do not supply a value for any other parameter, the contents of the current cell are selected.

P2

Optional Integer value which represents the Row ID of the first cell to be selected. Default is 0.

P3

Optional Integer value which represents the Column ID of the first cell to be selected. Default is 0.

P4

Optional Integer value which represents the Row ID of the last cell to be selected. Default is 0.

EndCol

Optional Integer value which represents the Column ID of the last cell to be selected. Default is 0.

SelectWholeCell

Optional Boolean value which specifies whether whole cells should be selected. Default is 0.

The row ID and column ID are zero based values, which means the first row and column of a table have an ID value of 0.

Word Pro: SelectTable method

{button ,AL('H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL('H_SELECTTABLE_METHOD_EXSCRIPT',1)} [See example](#)

Selects the contents of the table in which the insertion point is located. Equivalent to choosing Table - Select - Entire Table Contents.

Syntax

[objectreference].SelectTable()

Parameters

None.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Word Pro: SelectWord method

{button ,AL('H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL('H_SELECTWORD_METHOD_EXSCRIPT',1)} [See example](#)

Selects the word at the insertion point.

Syntax

[objectreference].SelectWord()

Parameters

None.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

A word is comprised of a contiguous string of alphanumeric characters. Punctuation and spaces are seen as the end of a word. If the insertion point is between two spaces, Word Pro selects all the spaces on both sides of the insertion point, as well as the word preceding the spaces.

Word Pro: Select method

{button ,AL(^H_WPAPPLICATION_CLASS;H_GRAPHICOLEOBJECT_CLASS;H_GRAPHIC_CLASS;H_OLEOBJECT_CLASS;H_CLICKHERE_CLASS;H_TEXTMARKER_CLASS;H_TEXT_CLASS',0)} [See list of classes](#)

{button ,AL(^H_SELECT_METHOD_EXSCRIPT',1)} [See example](#)

Selects the type of object specified in the ObjectType parameter.

Syntax

[objectreference].Select(ObjectType)

Parameters

ObjectType

Specifies what type of object to select. Data type is Variant, which allows the value of this parameter to be one of the constants listed below or its numeric equivalent (in parentheses). There is no default value.

<u>Value</u>	<u>Effect</u>
\$LwpSelectObjectTypeBullet (1757)	Selects the next bullet object in the text stream.
\$LwpSelectObjectTypeChunk (1750)	A chunk is comprised of a single word (a group of characters with no spaces) and all the contiguous spaces following that word. If the insertion point is at the beginning, the end, or anywhere within a word, Word Pro selects that word and the spaces which follow it. If the insertion point is between two spaces, Word Pro selects all the spaces following the insertion point to the beginning of the next word. If there is no word between the spaces and the end of the paragraph, Word Pro selects to the end of the paragraph.
\$LwpSelectObjectTypeDocument (1754)	Selects the contents of the entire document.
\$LwpSelectObjectTypeLevel (1756)	Undefined.
\$LwpSelectObjectTypeObject (1748)	Selects the next object of any type.
\$LwpSelectObjectTypeParagraph (1753)	Selects the paragraph object in which the insertion point is located.
\$LwpSelectObjectTypeSection (1759)	Selects the contents of the section in which the insertion point is located.
\$LwpSelectObjectTypeSentence (1752)	Selects the sentence in which the insertion point is located.
\$LwpSelectObjectTypeStream (1755)	Selects the text stream in which the insertion point is located. A text stream is comprised of all paragraphs of text and tables, but no frames or OLE objects. Note OLE is not supported under OS/2.
\$LwpSelectObjectTypeTombstoneset (1758)	Undefined.
\$LwpSelectObjectTypeTuna (1751)	

\$LwpSelectObjectTypeWord (1749)

Undefined.

Selects the word at the insertion point. In this case, a word is comprised of a contiguous string of alphanumeric characters. Punctuation and spaces are seen as the end of a word. If the insertion point is between two spaces, Word Pro selects all the spaces on both sides of the insertion point, as well as the word preceding the spaces.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Word Pro: SendFrameToBackOne method

{button ,AL('H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL('H_SENDFRAMETOBACKONE_METHOD_EXSCRIPT',1)} [See example](#)

Changes the priority of the selected frame so that it is one level back from its original position.

Syntax

[objectreference].SendFrameToBackOne()

Parameters

None.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

To call this method, the "Place frame" option for the frame (from the Frame InfoBox, click the Placement tab, click Placement and Anchoring Options to go to the Placement Options dialog box) must be one of the following: on all pages, on left/right pages, on current page. You can change the frame's placement using the Anchor method for the desired frame.

Word Pro: SendFrameToBack method

{button ,AL('H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL('H_SENDFRAMETOBACK_METHOD_EXSCRIPT',1)} [See example](#)

Changes the priority of the currently selected frame so that it is behind all the other frames on the page.

Syntax

[objectreference].SendFrameToBack()

Parameters

None.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

To call this method, the "Place frame" option for the frame (from the Frame InfoBox, click the Placement tab, click Placement and Anchoring Options to go to the Placement Options dialog box) must be one of the following: on all pages, on left/right pages, on current page. You can change the frame's placement using the Anchor method for the desired frame.

Word Pro: SendMailAndAttach method

{button ,AL(^H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

Syntax

[objectreference].SendMailAndAttach(MessageToAll, Subject, [Location1], [Message1], [Location2], [Message2], [Location3], [Message3], [Location4], [Message4], [Location5], [Message5])

Parameters

MessageToAll

Data type is String

Subject

Data type is String

Location1 - Location5

Data type is String. Optional parameter.

Message1 - Message5

Data type is String. Optional parameter.

Return value

Integer

Usage

Word Pro: SendMailSelectedText method

{button ,AL('H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL('H_SENDMAILSELECTEDTEXT_METHOD_EXSCRIPT',1)} [See example](#)

Uses the selected text as the body of a mail message that is mailed using the MAPI application specified in your WIN.INI file. Equivalent to choosing File - TeamMail and selecting "Message with current selection's text as message body."

Syntax

[objectreference].SendMailSelectedText()

Parameters

None.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Word Pro: SetArrayProp method

{button ,AL('H_SILVERBULLET_CLASS;H_USERINTERFACEPREFS_CLASS',0)} [See list of classes](#)

Sets the properties of an array. This method is defined in the following classes:

[SilverBullet]

[UserInterfacePrefs]

Syntax

[objectreference].SilverBullet.SetArrayProp(BulletArrayProp,Level,NewValue)

[objectreference].UserInterfacePrefs.SetArrayProp(PrefPropScope,Index,New)

Parameters

BulletArrayProp

Data type is Variant. The value of this parameter must be one of the strings below or its code equivalent.

<u>Value</u>	<u>Effect</u>
\$LwpBulletArrayPropCumulative (79)	
\$LwpBulletArrayPropDivision (81)	
\$LwpBulletArrayPropLesser (78)	
\$LwpBulletArrayPropLesserspecific (77)	
\$LwpBulletArrayPropSection (80)	

Level

Data type is Integer.

NewValue

Data type is Integer.

Positions

Data type is Variant. The value of this parameter must be one of the strings below or its code equivalent.

<u>Value</u>	<u>Effect</u>
\$LwpPositionsAfter (1639)	
\$LwpPositionsBefore (1637)	
\$LwpPositionsEqual (1638)	
\$LwpPositionsUnknown (1636)	

Index

Data type is Integer. Legal values are 0, 1, 2, and 3..

NewValue

Data type is String. The strings contain the text that you want to set in the array, in this case, the text you want in the "Find" and "Replace with" boxes.

Return value

Both SilverBullet and UserInterfacePrefs objects return a Boolean value.

Usage

[SilverBullet]

[UserInterfacePrefs]

Sets the most recently used list of Find strings or Replace strings that display in the user interface. It does not set the actual Find or Replace string.

Word Pro: SetData method

{button ,AL('H_SCRIPTDATASET_CLASS;H_WPDATASET_CLASS',0)} [See list of classes](#)

{button ,AL('H_SETDATA_METHOD_EXSCRIPT',1)} [See example](#)

Creates a data set variable and a value, or updates an existing data set variable value.

Syntax

[objectreference].SetData(DataName, NewData)

Parameters

DataName

The variable name in a data set. Data type is String.

NewData

The new data set item that you are creating. Data type is String.

Return value

[Variant](#)

Usage

If there is no data set variable with the name specified in the DataName parameter, a new data set variable is created and assigned the value specified by the NewData parameter.

If a data set variable already exists with the name specified in the DataName parameter, its value is updated with the value specified in the NewData parameter.

Word Pro: SetDocumentEpoch method

{button ,AL(^H_DIVISION_CLASS;H_TEXTDOCUMENT_CLASS',0)} [See list of classes](#)

Syntax

[objectreference].SetDocumentEpoch()

Parameters

Return value

Usage

Word Pro: SetFieldFormula method

{button ,AL('H_POWERFIELD_CLASS',0)} [See list of classes](#)

Syntax

[objectreference].SetFieldFormula(Formula, [Type])

Parameters

Formula

Data type is String.

Type

Data type is Variant. Optional parameter. Default is 0. The value of this parameter must be one of the strings below or its code equivalent.

\$LwpPFTypeBookmark (2003)

\$LwpPFTypeButton (2007)

\$LwpPFTypeDde (2002)

\$LwpPFTypeDocvar (2011)

\$LwpPFTypeField (2004)

\$LwpPFTypeIndex (2009)

\$LwpPFTypeMarker (2010)

\$LwpPFTypeMergevar (2013)

\$LwpPFTypePrtescape (2008)

\$LwpPFTypeSeq (2005)

\$LwpPFTypeSet (2006)

\$LwpPFTypeToc (2012)

Return value

Usage

Word Pro: SetFocus method

{button ,AL('H_DOCWINDOW_CLASS',0)} [See list of classes](#)

Syntax

[objectreference].SetFocus()

Parameters

Return value

Boolean.

Usage

Word Pro: SetFormula method

{button ,AL('H_CELLENGINE_CLASS',0)} [See list of classes](#)

{button ,AL('H_SETFORMULA_METHOD_EXSCRIPT',1)} [See example](#)

Inserts a formula into the current table cell.

Syntax

[objectreference].SetFormula(Row, Column, Formula)

Parameters*Row*

This Integer parameter allows you to specify the row ID of the cell in which you want to set a formula.

Column

This Integer parameter allows you to specify the column ID of the cell in which you want to set a formula.

Formula

A String value which represents the formula to be inserted into the specified table cell.

Return value**Usage**

Equivalent to choosing Table - Insert Formula.

Word Pro: SetLastUsedFilter method

{button ,AL('H_FILTER_CLASS',0)} [See list of classes](#)

{button ,AL('H_SETLASTUSEDFILTER_METHOD_EXSCRIPT',1)} [See example](#)

Syntax

[objectreference].SetLastUsedFilter(Type, Filter)

Parameters

FilterType

Data type is Variant. The value of this parameter must be one of the strings below or its code equivalent.

\$LwpFilterTypeGraphic (280)

\$LwpFilterTypeTable (281)

\$LwpFilterTypeText (279)

Filter

Data type is String.

Return value

Usage

Word Pro: SetLineOneSide method

{button ,AL('H_TABLELINE_CLASS',0)} [See list of classes](#)

{button ,AL('H_SETLINEONESIDE_METHOD_EXSCRIPT',1)} [See example](#)

This method is used to select the line style for a specific side of a table.

Syntax

[objectreference].SetLineOneSide(LinePlacement, LineStyle, LineWidth, LineColor, TableMix)

Parameters

LinePlacement

Indicates which side of the table to set a specific line style. Data type is Variant. The value of this parameter must be one of the strings below or its numeric equivalent (in parentheses).

LwpLinePlacementAllsides (&HF)

LwpLinePlacementBottom (&H8)

LwpLinePlacementLeft (&H1)

LwpLinePlacementRight (&H2)

LwpLinePlacementTop (&H4)

LineStyle

Allows you to set one of the line styles below to a specific side of a table. Data type is Variant. The value of this parameter must be one of the strings below or its numeric equivalent (in parentheses).

\$LtsBorderPatternBorderDot (1056964663)

\$LtsBorderPatternDashDot (1056964659)

\$LtsBorderPatternDashDotDot (1056964660)

\$LtsBorderPatternDashed (1056964662)

\$LtsBorderPatternDot (2498)

\$LtsBorderPatternDouble (1056964666)

\$LtsBorderPatternLongDash (1056964661)

\$LtsBorderPatternNone (1056964657)

\$LtsBorderPatternSolid (1056964658)

\$LwpBorderPattern13space (36)

\$LwpBorderPattern31space (37)

\$LwpBorderPatternButtondown (35)

\$LwpBorderPatternButtonup (34)

\$LwpBorderPatternCircle (41)

\$LwpBorderPatternDbIThick (51)

\$LwpBorderPatternDbIWavy (56)

\$LwpBorderPatternDeco1 (44)

\$LwpBorderPatternDeco2 (45)

\$LwpBorderPatternDeco3 (50)

\$LwpBorderPatternDiagonal (38)

\$LwpBorderPatternGirder (2571)

\$LwpBorderPatternPin (47)

\$LwpBorderPatternRain (46)

\$LwpBorderPatternRope (43)

\$LwpBorderPatternRose (48)

\$LwpBorderPatternStar (42)

\$LwpBorderPatternSunf (49)

\$LwpBorderPatternTaro (39)

\$LwpBorderPatternThickDbwavy (58)
\$LwpBorderPatternThickThin (53)
\$LwpBorderPatternThickWavy (57)
\$LwpBorderPatternThinThick (54)
\$LwpBorderPatternThinThickThin (52)
\$LwpBorderPatternWarning (2379)
\$LwpBorderPatternWavy (55)
\$LwpLtsBorderPatternDot (40)

LineWidth

Specifies the width for the line style of a specific side of a table. Data type is Twips.

LineColor

Specifies the line color for the line style of a specific side of a table. Data type is Long.

TableMix

Allows you to specify if a table contains more than one type of line style. Data type is Variant. The value of this parameter must be one of the strings below or its code equivalent.

Value	Effect
\$LwpTableMixAllmixed (1893)	Assigns a different line style to each line of a table cell.
\$LwpTableMixBottommixed (1890)	Assigns a line style to the bottom line of a table cell that is different from the line style of the top, right, and left line of a table cell.
\$LwpTableMixLeftmixed (1891)	Assigns a line style to the left line of a table cell that is different from the line style of the top, right, and bottom line of a table cell.
\$LwpTableMixRightmixed (1892)	Assigns a line style to the right line of a table cell that is different from the line style of the top, bottom, and left line of a table cell.
\$LwpTableMixTopmixed (1889)	Assigns a line style to the top line of a table cell that is different from the line style of the bottom, right, and left line of a table cell.

Return value

Usage

To use this method, you must precede it with the method `BeginCustomLines`, and you must follow it with the method `EndCustomLines`.

Note This method will never appear in the IDE during the process of recording a script.

Word Pro: SetLinesAllSides method

{button ,AL('H_TABLELINE_CLASS',0)} [See list of classes](#)

Allows you to simultaneously set a specific line style to all sides of an entire table, or a specific row, column or cell.

Syntax

[objectreference].SetLinesAllSides(LinesAroundCells, LineStyle, LineWidth, LineColor, OutlineLineStyle, OutlineLineWidth, OutlineLineColor)

Parameters

LinesAroundCells

Allows you to specify where the line style should be applied in the table object. Data type is Variant.

- \$LwpTableLineStyleAll (1878)
- \$LwpTableLineStyleCols (1882)
- \$LwpTableLineStyleCustom (1886)
- \$LwpTableLineStyleInnercols (1887)
- \$LwpTableLineStyleInnerRowscols (1888)
- \$LwpTableLineStyleMixed (1885)
- \$LwpTableLineStyleNone (1877)
- \$LwpTableLineStyleOutline (1879)
- \$LwpTableLineStyleOutlineall (1880)
- \$LwpTableLineStyleOutlinecols (1884)
- \$LwpTableLineStyleOutlinerows (1883)
- \$LwpTableLineStyleRows (1881)

LineStyle

Allows you set one of the lines styles listed below to a table object. Data type is Variant.

- \$LtsBorderPatternBorderDot (1056964663)
- \$LtsBorderPatternDashDot (1056964659)
- \$LtsBorderPatternDashDotDot (1056964660)
- \$LtsBorderPatternDashed (1056964662)
- \$LtsBorderPatternDot (2498)
- \$LtsBorderPatternDouble (1056964666)
- \$LtsBorderPatternLongDash (1056964661)
- \$LtsBorderPatternNone (1056964657)
- \$LtsBorderPatternSolid (1056964658)
- \$LwpBorderPattern13space (36)
- \$LwpBorderPattern31space (37)
- \$LwpBorderPatternButtondown (35)
- \$LwpBorderPatternButtonup (34)
- \$LwpBorderPatternCircle (41)
- \$LwpBorderPatternDbIThick (51)
- \$LwpBorderPatternDbIWavy (56)
- \$LwpBorderPatternDeco1 (44)
- \$LwpBorderPatternDeco2 (45)
- \$LwpBorderPatternDeco3 (50)
- \$LwpBorderPatternDiagonal (38)
- \$LwpBorderPatternGirder (2571)
- \$LwpBorderPatternPin (47)
- \$LwpBorderPatternRain (46)

\$LwpBorderPatternRope (43)
\$LwpBorderPatternRose (48)
\$LwpBorderPatternStar (42)
\$LwpBorderPatternSunf (49)
\$LwpBorderPatternTaro (39)
\$LwpBorderPatternThickDblwavy (58)
\$LwpBorderPatternThickThin (53)
\$LwpBorderPatternThickWavy (57)
\$LwpBorderPatternThinThick (54)
\$LwpBorderPatternThinThickThin (52)
\$LwpBorderPatternWarning (2379)
\$LwpBorderPatternWavy (55)
\$LwpLtsBorderPatternDot (40)

LineWidth

Specifies the width of the selected line in Twips. There are 1440 Twips per inch. Data type is Long. For more information on Twips, see [Overview: Word Pro LotusScript Units Of Measurement](#).

LineColor

This Long parameter specifies the color for the selected line style. For more information on representing color information with a single value, see the GetRGB method topic.

OutlineLineStyle

Specifies the line style for the outline. Data type is Variant.

\$LtsBorderPatternDashDot (1056964659)
\$LtsBorderPatternDashDotDot (1056964660)
\$LtsBorderPatternDashed (1056964662)
\$LtsBorderPatternDot (1056964663)
\$LtsBorderPatternDouble (1056964666)
\$LtsBorderPatternLongDash (1056964661)
\$LtsBorderPatternNone (1056964657)
\$LtsBorderPatternSolid (1056964658)
\$LwpBorderPattern13space (36)
\$LwpBorderPattern31space (37)
\$LwpBorderPatternButtondown (35)
\$LwpBorderPatternButtonup (34)
\$LwpBorderPatternCircle (41)
\$LwpBorderPatternDblThick (51)
\$LwpBorderPatternDblWavy (56)
\$LwpBorderPatternDeco1 (44)
\$LwpBorderPatternDeco2 (45)
\$LwpBorderPatternDeco3 (50)
\$LwpBorderPatternDiagonal (38)
\$LwpBorderPatternPin (47)
\$LwpBorderPatternRain (46)
\$LwpBorderPatternRope (43)
\$LwpBorderPatternRose (48)
\$LwpBorderPatternStar (42)
\$LwpBorderPatternSunf (49)

\$LwpBorderPatternTaro (39)
\$LwpBorderPatternThickDblwavy (58)
\$LwpBorderPatternThickThin (53)
\$LwpBorderPatternThickWavy (57)
\$LwpBorderPatternThinThick (54)
\$LwpBorderPatternThinThickThin (52)
\$LwpBorderPatternWavy (55)
\$LwpLtsBorderPatternDot (40)

OutlineLineWidth

Specifies the width of the outline in Twips. There are 1440 Twips per inch. Data type is Long. For more information on Twips, see [Overview: Word Pro LotusScript Units Of Measurement](#).

OutlineLineColor

This Long parameter specifies the color of the outline. For more information on representing color information with a single value, see the GetRGB method topic.

Return value

Usage

Word Pro: SetLinkSource method

{button ,AL(^H_GRAPHIC_CLASS;H_OLEOBJECT_CLASS',0)} [See list of classes](#)

Note This method is not implemented for the OLEObject class within OS/2.

Syntax

[objectreference].SetLinkSource(LinkDisplayName, FileNameLength, ValidateSource, LinkCookie)

[objectreference].SetLinkSource(LinkDisplayName, FileNameLength, ValidateSource)

Parameters

LinkDisplayName

Data type is String.

FileNameLength

Data type is Long.

ValidateSource

Data type is Integer.

LinkCookie

Data type is Long.

Return value

Long.

Usage

Word Pro: SetMinimumOrigin method

```
{button ,AL(^H_LAYOUT_CLASS;H_CELLGROUPLAYOUT_CLASS;H_CELLLAYOUT_CLASS;H_CONNECTEDLAY  
OUT_CLASS;H_COLUMNGROUPLAYOUT_CLASS;H_FOOTERLAYOUT_CLASS;H_FRAMELAYOUT_CLASS;H  
_FRAMEGROUPLAYOUT_CLASS;H_DROP_CAPLAYOUT_CLASS;H_GROUPLAYOUT_CLASS;H_HEADERLAY  
OUT_CLASS;H_NOTELAYOUT_CLASS;H_PAGELAYOUT_CLASS;H_ROWGROUPLAYOUT_CLASS;H_ROWLA  
YOUT_CLASS;H_RUBYLAYOUT_CLASS;H_SUPERTABLEGROUPLAYOUT_CLASS;H_SUPERTABLELAYOUT  
_CLASS;H_TABLELAYOUT_CLASS;H_ENDNOTELAYOUT_CLASS;H_FOOTNOTELAYOUT_CLASS;H_TABLE  
HEADINGLAYOUT_CLASS;H_TOCSUPERTABLELAYOUT_CLASS',0)} See list of classes
```

Moves the layout to make the length of the anchor tether as short as the anchor position allows.

Syntax

```
[objectreference].SetMinimumOrigin()
```

Parameters

Return value

The return value for this method will always be -1 or 0. When testing the return value, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Word Pro: SetNamedProperty method

```
{button ,AL('H_CHARACTERSTYLE_CLASS;H_DIVISION_CLASS;H_LAYOUT_CLASS;H_CELLGROUPLAYOUT_
CLASS;H_CELLLAYOUT_CLASS;H_CONNECTEDLAYOUT_CLASS;H_COLUMNGROUPLAYOUT_CLASS;H_F
OOTERLAYOUT_CLASS;H_FRAMELAYOUT_CLASS;H_FRAMEGROUPLAYOUT_CLASS;H_DROPCAPLAYOU
T_CLASS;H_GROUPLAYOUT_CLASS;H_HEADERLAYOUT_CLASS;H_NOTELAYOUT_CLASS;H_PAGELAYOU
T_CLASS;H_ROWGROUPLAYOUT_CLASS;H_ROWLAYOUT_CLASS;H_RUBYLAYOUT_CLASS;H_SUPERTAB
LEGROUPLAYOUT_CLASS;H_SUPERTABLELAYOUT_CLASS;H_TABLELAYOUT_CLASS;H_ENDNOTELAYOU
T_CLASS;H_FOOTNOTELAYOUT_CLASS;H_TABLEHEADINGLAYOUT_CLASS;H_TOCSUPERTABLELAYOUT
_CLASS;H_MARKER_CLASS;H_POWERFIELD_CLASS;H_CLICKHERE_CLASS;H_TEXTMARKER_CLASS;H_
RUBYMARKER_CLASS;H_TABLEMARKER_CLASS;H_TEXTDOCUMENT_CLASS',0)} See list of classes
```

```
{button ,AL('H_SETNAMEDPROPERTY_METHOD_EXSCRIPT',1)} See example
```

Creates and assigns a value to a named property.

Syntax

```
[objectreference].SetNamedProperty(PropertyName, NewValue)
```

Parameters

PropertyName

A String expression representing the name of the property to which you are assigning a value.

NewValue

A String expression representing the new value you want to assign to the property.

Return value

The return value for this method will always be -1 or 0. When testing the return value, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

A named property is a user-defined property assigned to an object. Unlike variables, named properties are persistent. They continue to exist when a script stops executing, and when a document is closed and reopened.

If you want to modify the value of an existing named property, pass its name to the method in the *PropertyName* parameter. The named property is then assigned the string value that you specify in the *NewValue* parameter.

Word Pro: SetOverrideGraphic method

{button ,AL(^H_STATUSBARBUTTON_CLASS',0)} [See list of classes](#)

Sets the graphic on a button. This method is called when responding to the StatusBarButtonOverrideGraphic event, or the StatusBarButtonOverrideTextAndGraphic event.

Syntax

[objectreference].SetOverrideGraphic(hGraphic)

Parameters

hGraphic

Data type is Long and represents the handle to the bitmap that you want to appear on the button. The bitmap is deleted for you, so do not use the bitmap again after you make this call.

Return value

Integer

Usage

This method must be used within the StatusBarButtonOverrideGraphic event. If you want to force the graphic to change, you can use the InvalidateButton method to force the StatusBarButtonOverrideGraphic event.

Word Pro: SetOverrideText method

{button ,AL('H_STATUSBARBUTTON_CLASS',0)} [See list of classes](#)

{button ,AL('H_SETOVERRIDE TEXT_METHOD_EXSCRIPT',1)} [See example](#)

Sets the text on a button. This method is called when responding to the StatusBarButtonOverrideText event or the StatusBarButtonOverrideTextAndGraphic event.

Syntax

[objectreference].SetOverrideText(Text)

Parameters

Text

Data type is String and represents the text you want to appear on the button. The text is deleted for you, so do not use it again after you make this call.

Return value

Integer.

Usage

This method must be used within the StatusBarButtonOverrideText event. If you want to force the text to change, you can use the InvalidateButton method to force the StatusBarButtonOverrideText event. If the text on the status bar button is never going to change, you can use the LwpButtonNoTextFromHost (&H800) parameter when the button is created.

Word Pro: SetPageBottomMargin method

{button ,AL('H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL('H_SETPAGEBOTTOMMARGIN_METHOD_EXSCRIPT',1)} [See example](#)

Syntax

[objectreference].SetPageBottomMargin(Amount[, RightPage])

Parameters

Amount

Data type is Long, measured in [Twips](#).

RightPage

Data type is Integer.

Return value

Integer.

Usage

Word Pro: SetPageTopMargin method

{button ,AL('H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL('H_SETPAGETOPMARGIN_METHOD_EXSCRIPT',1)} [See example](#)

Syntax

[objectreference].SetPageTopMargin(Amount[, RightPage])

Parameters

Amount

Data type is Long, measured in [Twips](#).

RightPage

Data type is Integer

Return value

Integer.

Usage

Word Pro: SetPattern method

{button ,AL('H_TABLEFILL_CLASS',0)} [See list of classes](#)

{button ,AL('H_SETPATTERN_METHOD_EXSCRIPT',1)} [See example](#)

Allows you to simultaneously set the following: the specific part of the table you want to fill, the background fill style, the background color, and the pattern color.

Syntax

[objectreference].SetPattern(TableFillStyle, BackgroundFill, BackgroundColor, PatternColor)

Parameters

TableFillStyle

Allows you to set or not to set a table fill style to specific parts of a table object. The value of this Variant parameter must be one of the strings below or its code equivalent.

<u>Value</u>	<u>Effect</u>
\$LwpTableFillStyleAll (1870)	Fills the entire table with a specific fill style.
\$LwpTableFillStyleEveryothercol (1872)	Fills every other column with a specific fill style.
\$LwpTableFillStyleEveryotherrow (1871)	Fills every other row with a specific fill style.
\$LwpTableFillStyleMixed (1873)	Uses more than one fill style for the table object.
\$LwpTableFillStyleNone (1874)	Does not set a fill style for the table object.

BackgroundFill

This Variant parameter allows you to set one of the values below as the style for the background fill in a table object. The value of this parameter must be one of the strings below or its numeric equivalent.

<u>Value</u>	<u>Effect</u>
\$LtsFillBarLeftDiag (1056964681)	
\$LtsFillBarRightDiag (1056964688)	
\$LtsFillBasket (1056964710)	
\$LtsFillBigCheck (1056964717)	
\$LtsFillBottomTopGrad (1056964729)	
\$LtsFillBrick (1056964712)	
\$LtsFillBubbles (1056964724)	
\$LtsFillChevron (1056964708)	
\$LtsFillCircles (1056964719)	
\$LtsFillClumpedNarrowDiagHatch (1056964696)	
\$LtsFillClumpedZs (1056964723)	
\$LtsFillDarkNarrowDiagHatch (1056964693)	
\$LtsFillDiagBasket (1056964711)	
\$LtsFillDiagBrick (1056964713)	
\$LtsFillDiagHatch (1056964695)	
\$LtsFillDiamonds (1056964725)	
\$LtsFillDottedDarkHash (1056964864)	
\$LtsFillDottedDarkHatch (2499)	
\$LtsFillDottedZigzag (1056964726)	
\$LtsFillDoubleLeftDiag (1056964684)	
\$LtsFillDoubleRightDiag (1056964690)	
\$LtsFillGray1 (1056964669)	
\$LtsFillGray10 (1056964678)	

\$LtsFillGray2 (1056964670)
\$LtsFillGray3 (1056964671)
\$LtsFillGray4 (1056964672)
\$LtsFillGray5 (1056964673)
\$LtsFillGray6 (1056964674)
\$LtsFillGray7 (1056964675)
\$LtsFillGray8 (1056964676)
\$LtsFillGray9 (1056964677)
\$LtsFillHoriz (1056964699)
\$LtsFillHorizBar (1056964698)
\$LtsFillHorizCheckerboard (1056964716)
\$LtsFillIrregularDiagScales (1056964721)
\$LtsFillLeftDiag (1056964682)
\$LtsFillLeftNarrowDiagHatch (1056964694)
\$LtsFillLeftRightGrad (1056964728)
\$LtsFillNarrowDoubleLeftDiag (1056964685)
\$LtsFillNarrowDoubleRightDiag (1056964863)
\$LtsFillNarrowHoriz (1056964697)
\$LtsFillNarrowVert (1056964701)
\$LtsFillNeToSwDiagStripGrad (1056964738)
\$LtsFillNeToSwGrad (1056964730)
\$LtsFillNone (1056964667)
\$LtsFillNwToSeDiagStripGrad (1056964739)
\$LtsFillNwToSeGrad (1056964731)
\$LtsFillRandomBar (1056964680)
\$LtsFillRandomSquare (1056964679)
\$LtsFillRegularCheck (1056964718)
\$LtsFillRegularHatch (1056964706)
\$LtsFillRightDiag (1056964689)
\$LtsFillRtLeftGrad (1056964744)
\$LtsFillRunningDash (1056964714)
\$LtsFillScalesDown (1056964722)
\$LtsFillScalesUp (1056964720)
\$LtsFillSolid (1056964668)
\$LtsFillSteel (1056964709)
\$LtsFillTinyHatch (1056964705)
\$LtsFillTopBottomGrad (1056964745)
\$LtsFillTripleLeftDiag (1056964686)
\$LtsFillTripleRightDiag (1056964691)
\$LtsFillVert (1056964703)
\$LtsFillVertBar (1056964702)
\$LtsFillVertCheckerboard (1056964715)
\$LtsFillWideHatch (1056964707)
\$LtsFillWideHoriz (1056964700)
\$LtsFillWideLeftDiag (1056964687)

\$LtsFillWideRightDiag (1056964692)
 \$LtsFillWideVert (1056964704)
 \$LtsIntervalYear (1056964854)
 \$LtsTimeFormatHhMm24hr (1056964853)
 \$LwpFillIndian3 (273)
 \$LwpFillPattern (2000)
 \$LwpFillPeachpie (274)

BackgroundColor

This Long parameter specifies the background color of the selected table cells.

Colors are usually represented by a combination of three separate components. The three components include a red value which can range from 0-255, a green value which can range from 0-255, and a blue value which can range from 0-255. You can combine different amounts of these three component colors to produce any other color. For example, in order to produce yellow, you can set a color object's red value to 255, green value to 255, and blue value to 0. The combination of all three component colors appears yellow.

Colors can also be represented by a single numeric value. This allows you to specify any available color in only one method parameter, as opposed to three separate parameters. In order to calculate the value that represents a specific color, use this formula:

$$(\text{RedValue} * 65536\&) + (\text{GreenValue} * 256\&) + \text{BlueValue}$$

The "&" suffix appended to the constant values above ensures that the results of the expressions are always Long values. If you do not append the "&," the result of the expression may cause an overflow error.

The table below shows some examples of this formula, the result of the formula and the color represented by the value.

Color	Formula	Result
White	$(255 * 65536\&) + (255 * 256\&) + 255$	16777215
25% Gray	$(192 * 65536\&) + (192 * 256\&) + 192$	12632256
Red	$(255 * 65536\&) + (0 * 256\&) + 0$	16711680
Yellow	$(255 * 65536\&) + (255 * 256\&) + 0$	16776960
Neon Green	$(0 * 65536\&) + (255 * 256\&) + 0$	65280
Turquoise	$(0 * 65536\&) + (255 * 256\&) + 255$	65535
Blue	$(0 * 65536\&) + (0 * 256\&) + 255$	255
Hot Pink	$(255 * 65536\&) + (0 * 256\&) + 255$	16711935
Black	$(0 * 65536\&) + (0 * 256\&) + 0$	0
50% Gray	$(128 * 65536\&) + (128 * 256\&) + 128$	8421504
Scarlet	$(128 * 65536\&) + (0 * 256\&) + 0$	8388608
Olive	$(128 * 65536\&) + (128 * 256\&) + 0$	8421376
Dark Green	$(0 * 65536\&) + (128 * 256\&) + 0$	32768
Aztec Blue	$(0 * 65536\&) + (128 * 256\&) + 128$	32896
Dark Grape	$(0 * 65536\&) + (0 * 256\&) + 97$	97
Plum Red	$(128 * 65536\&) + (0 * 256\&) + 128$	8388736

PatternColor

Specifies the pattern color of the selected table cells. This parameter is a Long value, which represents the addition of a red, a green, and a blue color component value. For information on how to calculate the value for this parameter, refer to the BackgroundColor parameter definition.

Return value

The return value for this method will always be -1 or 0. When testing the return value, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Use this method to specify a fill type, a pattern, a background color, and a pattern color for a selected range of table cells.

Word Pro: SetPopupAlignment method

{button ,AL('H_STATUSBARBUTTON_CLASS',0)} [See list of classes](#)

This method is called when responding to the StatusBarButtonFillPopupList event. Allows you to align the contents in the popup list in a status bar button.

Syntax

[objectreference].SetPopupAlignment(AlignType)

Parameters

AlignType

Data type is Variant, which allows the value of this parameter to be one of the constants listed below or its numeric equivalent (in parentheses). There is no default constant.

Value	Effect
\$LwpButtonAlignAlignCenter (89)	Specifies center alignment of the button's contents.
\$LwpButtonAlignAlignLeft (87)	Specifies left alignment of the button's contents.
\$LwpButtonAlignAlignRight (88)	Specifies right alignment of the button's contents.

Return value

Integer. Always returns True.

Usage

This method allows you to align the contents in the button's popup list.

Word Pro: SetPopupIndex method

{button ,AL(^H_STATUSBARBUTTON_CLASS',0)} [See list of classes](#)

This method is called when responding to the StatusBarButtonFillPopupList event. Allows you to select which item in the status bar button's popup list is highlighted by definition.

Syntax

[objectreference].SetPopupIndex(Index)

Parameters

Index

Data type is Integer.

Return value

True if item was selected; False if no item was selected.

Usage

Allows you to indicate which item in the popup list should be highlighted and selected by default. The index is the number corresponding to the entry you want to select in the list (zero-based). For example, 0 is the top item, 1 is the next item, 2 is the next, and so on.

If you specify an index greater than the number of items, this method returns False and nothing is selected.

Word Pro: SetPopupWidthType method

{button ,AL(^H_STATUSBARBUTTON_CLASS',0)} [See list of classes](#)

This method is called when responding to the StatusBarButtonFillPopupList event.

Syntax

[objectreference].SetPopupWidthType(WidthType)

Parameters

WidthType

Data type is Variant, which allows the value of this parameter to be one of the constants listed below or its numeric equivalent (in parentheses). There is no default constant.

<u>Value</u>	<u>Effect</u>
\$LwpButtonSizeToButton (90)	Specifies that the width of the popup list will be the width of the button.
\$LwpButtonSizeToSpecified (92)	Specifies a user-defined width for the popup list, using the SetPopupWidth method to specify the width.
\$LwpButtonSizeToText (91)	Specifies that the width of the popup list will conform to the size of the widest text in the list. For example, the value is calculated to the size of the longest text string, or, if the text is shorter than the button's width, the value is calculated to the width of the button.

Return value

Integer.

Usage

Allows you to specify how the width of the popup list is calculated.

Word Pro: SetPopupWidth method

{button ,AL(^H_STATUSBARBUTTON_CLASS',0)} [See list of classes](#)

This method is called when responding to the StatusBarButtonFillPopupList event. Allows you to set the width of the popup list in the status bar button.

Syntax

[objectreference].SetPopupWidth(Width)

Parameters

Width

Data type is Integer. Set to width in Twips. There are 1440 Twips per inch.

Return value

Integer.

Usage

Use this method in conjunction with the StatusBarButtonFillPopupList event in the StatusBar class to set the width of the popup list. The width type parameter of the popup list must be \$LwpButtonSizeToSpecified to have any effect.

Word Pro: SetRGB method

{button ,AL('H_COLOR_CLASS',0)} [See list of classes](#)

{button ,AL('H_SETRGB_METHOD_EXSCRIPT',1)} [See example](#)

Sets the RGB (red, green, and blue) values of a specific object.

Syntax

[objectreference].SetRGB(r, g, b)

Parameters

r

The red component of a color. This Integer parameter can range from 0 - 255.

g

The green component of a color. This Integer parameter can range from 0 - 255.

b

The blue component of a color. This Integer parameter can range from 0 - 255.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Use this method to set an object's red, green, and blue color components simultaneously. If you want to set the color components individually, you can use the Red, Green, and Blue properties of a specific Color object.

Word Pro: SetStorage method

{button ,AL(^H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

Syntax

[objectreference].SetStorage(plStorage, FileType)

Parameters

plStorage

Data type is Long.

FileType

Data type is String.

Return value

Usage

Word Pro: SetStyle method

```
{button ,AL(^H_WPAPPLICATION_CLASS;H_CELLCONTAINER_CLASS;H_FRAMECONTAINER_CLASS;H_NOTECONTAINER_CLASS;H_RUBYCONTAINER_CLASS;H_DROPCAPCONTAINER_CLASS;H_CLICKHERE_CLASSES;H_TEXTMARKER_CLASS;H_PAGECONTAINER_CLASS;H_SUBPAGECONTAINER_CLASS;H_SUPERPAGECONTAINER_CLASS;H_SUPERTABLECONTAINER_CLASS;H_TEXT_CLASS',0)} See list of classes
```

```
{button ,AL(^H_SETSTYLE_METHOD_EXSCRIPT',1)} See example
```

Redefines the specified style to match the object from which the method is called. If the specified style does not exist, Word Pro creates it. The type of style defined or created by Word Pro depends on the type of object from which you call this method.

Syntax

```
[Objectreference].SetStyle(StyleType, Style[, Exceptions])
```

Parameters

StyleType

Indicates the type of object for which you want to set the style. Data type is Variant, which allows the value of this parameter to be one of the constants listed below or its numeric equivalent (in parentheses). There is no default value for this parameter, but you can use \$LwpstyleTypeDefault to have Word Pro assign the default style type for the type of object from which you call this method.

Value	Effect
\$LwpstyleTypeCell (1834)	Assigns a cell style.
\$LwpstyleTypeCharacter (1830)	Assigns a character style.
\$LwpstyleTypeDefault (1828)	Assigns the default style for the type of object from which you call this method.
\$LwpstyleTypeFrame (1832)	Assigns a frame style.
\$LwpstyleTypePage (1831)	Assigns a page style.
\$LwpstyleTypeParagraph (1829)	Assigns a paragraph style.
\$LwpstyleTypeTable (1833)	Assigns a table style.

Style

A String expression that identifies the style you want to assign. If the named style does not exist, Word Pro creates a new style with that name.

Exceptions

Allows you to specify which elements of a style do not get set by this method. There are two main types of style exceptions: Layout and Text. In addition, the default exception depends on the object from which you call this method.

<u>If you are calling SetStyle from this class...</u>	<u>The legal values for the ExceptionType parameter are listed in this table...</u>	<u>Use this default value unless you specify a different one in the argument...</u>
PageContainer SuperPageContainer SubPageContainer	<u>Layout Style Exceptions</u>	LwpLayStyOverPlacement (&H2)
FrameContainer NoteContainer CellContainer RubyContainer SuperTableContainer DropCapContainer	<u>Layout Style Exceptions</u>	LwpLayStyOverSizeAndPlacement (&H3)
ClickHere Text TextMarker	<u>Text Style Exceptions</u>	LwpTextStlyeOverridesNone (&H0)
WpApplication	N/A	0

Layout Style Exceptions

<u>Value</u>	<u>Effect</u>
LwpLayStyOverBackground (&H10)	Background settings are not assigned to the style specified in the Style parameter.
LwpLayStyOverBorders (&H8)	Border settings are not assigned to the style specified in the Style parameter.
LwpLayStyOverChildren (&H10000)	Child settings are not assigned to the style specified in the Style parameter.
LwpLayStyOverColumns (&H400)	Column settings are not assigned to the style specified in the Style parameter.
LwpLayStyOverContents (&H20000)	Content settings are not assigned to the style specified in the Style parameter.
LwpLayStyOverJoins (&H20)	Join settings are not assigned to the style specified in the Style parameter.
LwpLayStyOverLeaders (&H2000)	Leader settings are not assigned to the style specified in the Style parameter.
LwpLayStyOverMargins (&H4)	Margin settings are not assigned to the style specified in the Style parameter.
LwpLayStyOverMisc (&H8000)	Miscellaneous settings are not assigned to the style specified in the Style parameter. Miscellaneous settings are those settings not included in any of the other exception categories.
LwpLayStyOverNumerics (&H200)	Numeric settings are not assigned to the style specified in the Style parameter.
LwpLayStyOverOrientation (&H4000)	Orientation settings are not assigned to the style specified in the Style parameter.
LwpLayStyOverPlacement (&H2)	Placement settings are not assigned to the style specified in the Style parameter.
LwpLayStyOverRotation (&H1000)	Rotation settings are not assigned to the style specified in the Style parameter.
LwpLayStyOverScaling (&H800)	Scaling settings are not assigned to the style specified in the Style parameter.
LwpLayStyOverScript (&H100)	Script settings are not assigned to the style specified in the Style parameter.
LwpLayStyOverShadow (&H40)	Shadow settings are not assigned to the style specified in the Style parameter.
LwpLayStyOverSize (&H1)	Size settings are not assigned to the style specified in the Style parameter.
LwpLayStyOverSizeAndPlacement (&H3)	Size and placement settings are not assigned to the style specified in the Style parameter.
LwpLayStyOverTabs (&H80)	Tab settings are not assigned to the style specified in the Style parameter.

Text Style Exceptions

<u>Value</u>	<u>Effect</u>
LwpTextStlyeOverridesNone (&H0)	
LwpTextStyleBoverridesUllet (&H200)	
LwpTextStyleloverridesNdent (&H20)	
LwpTextStyleOverridesAlignment (&H10)	
LwpTextStyleOverridesAmikake (&H4000)	
LwpTextStyleOverridesAttributes (&H4)	
LwpTextStyleOverridesBorders (&H80)	

- LwpTextStyleOverridesBreaks (&H100)
- LwpTextStyleOverridesBullet (&H200)
- LwpTextStyleOverridesCharborder (&H2000)
- LwpTextStyleOverridesFace (&H1)
- LwpTextStyleOverridesFont (&H8)
- LwpTextStyleOverridesIndent (&H20)
- LwpTextStyleOverridesKinsoku (&H1000)
- LwpTextStyleOverridesNumbering (&H400)
- LwpTextStyleOverridesSize (&H2)
- LwpTextStyleOverridesSpacing (&H40)
- LwpTextStyleOverridesTabs (&H800)
- LwpTextStyleSoverridesPacing (&H40)

The WPAApplication object does not allow for the use of these exception values. Default when called from the WPAApplication object is 0. Data type is Long.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

When you are defining a style, this method allows you to specify the hierarchy for the style definition you are creating. Any exceptions you define are equivalent to those listed in the Style Hierarchy Definitions dialog box.

Word Pro: SetUpEnvelopeMerge method

{button ,AL('H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

Sets the options for an envelope merge and opens the Merge bar.

Syntax

[objectreference].SetUpEnvelopeMerge(envWidth, envHeight, envBin, barcode, returnAddress)

Parameters

envWidth

A numeric expression of type Long, which specifies the width of the envelopes you want to merge.

envHeight

A numeric expression of type Long, which specifies the height of the envelopes you want to merge.

envBin

A String expression which specifies the name of the printer bin which contains the envelopes. Available values for this parameter are displayed in your printer's Properties dialog box in the "Paper source" option.

barcode

An Integer expression which tells Word Pro whether or not you want to include a bar code on the envelope. The legal values for this parameter are -1 and 0 but you may use the LotusScript constants True (-1) and False (0) instead of the integer values. Optional parameter. Default is 0 (False).

returnAddress

An Integer expression which tells Word Pro whether or not you want to include a return address on the envelope. The legal values for this parameter are -1 and 0 but you may use the LotusScript constants True (-1) and False (0) instead of the integer values. Optional parameter. Default is 0 (False).

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Word Pro: Shade method

{button ,AL('H_CLICKHERE_CLASS;H_TEXTMARKER_CLASS;H_TEXT_CLASS',0)} [See list of classes](#)

{button ,AL('H_SHADE_METHOD_EXSCRIPT',1)} [See example](#)

Selects text in the current context.

Syntax

[objectreference].Shade(LocationType,Unit, N)

Parameters

LocationType

Data type is Variant. The value of this parameter must be one of the strings below or its code equivalent.

\$LwpLocationTypeDocument (573)

\$LwpLocationTypeLine (571)

\$LwpLocationTypeParagraph (572)

\$LwpLocationTypeSelection (568)

\$LwpLocationTypeSentence (570)

\$LwpLocationTypeStream (574)

\$LwpLocationTypeWord (569)

Unit

Data type is Variant. The value of this parameter must be one of the strings below or its code equivalent.

\$LwpNavigateDirectionDown (1515)

\$LwpNavigateDirectionLeft (1516)

\$LwpNavigateDirectionRight (1517)

\$LwpNavigateDirectionUp (1514)

Numberof Units

Data type is Integer.

Return value

Usage

Word Pro: ShowAnyGreeting method

{button ,AL(^H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

Syntax

[objectreference].ShowAnyGreeting()

Parameters

None

Return value

Integer

Usage

Word Pro: ShowCaretAndSelection method

{button ,AL(^H_CLICKHERE_CLASS;H_TEXTMARKER_CLASS;H_TEXT_CLASS',0)} [See list of classes](#)

Displays the current selection.

Syntax

[objectreference].ShowCaretAndSelection([ForceCaretToShow], [ShowCaret] [, ShowSelection])

Parameters

ForceCaretToShow

Forces the caret to the screen if True. Optional Integer parameter. Default is False.

ShowCaret

Data type is Integer. Optional Integer parameter. Default is True.

ShowSelection

Optional Integer parameter. Default is True.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Word Pro: ShowContainers method

```
{button ,AL(^H_BASECONTAINER_CLASS;H_CELLCONTAINER_CLASS;H_FRAMECONTAINER_CLASS;H_NOTECONTAINER_CLASS;H_RUBYCONTAINER_CLASS;H_DROPDOWNCONTAINER_CLASS;H_PAGECONTAINER_CLASS;H_SUBPAGECONTAINER_CLASS;H_SUPERPAGECONTAINER_CLASS;H_ROWCONTAINER_CLASS;H_SUPERTABLECONTAINER_CLASS;H_TABLECONTAINER_CLASS;H_PARALLELCOLSCONTAINER_CLASS;H_TABLEONLYCONT_CLASS',0)} See list of classes
```

Selects a specific container in a document, if the container is selectable.

Syntax

[objectreference].ShowContainers()

Parameters

Return value

This method always returns -1.

Usage

This method selects frame container objects and table container objects if the placement is set to "On current page."

Word Pro: ShowCursor method

{button ,AL(^H_CLICKHERE_CLASS;H_TEXTMARKER_CLASS;H_TEXT_CLASS',0)} [See list of classes](#)

Scrolls the active document window so the insertion point is visible on the screen.

Syntax

[objectreference].ShowCursor()

Parameters

None.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Word Pro: ShowIconBars method

{button ,AL('H_ICONBARMANAGER_CLASS',0)} [See list of classes](#)

{button ,AL('H_SHOWICONBARS_METHOD_EXSCRIPT',1)} [See example](#)

Displays the set of SmartIcons in its default location on the screen.

Syntax

[objectreference].ShowIconBars()

Parameters**Return value**

The return value for this method will always be -1 or 0. When testing the return value, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Word Pro: ShowScrollBars method

{button ,AL('H_DOCWINDOW_CLASS',0)} [See list of classes](#)

{button ,AL('H_SHOWSCROLLBARS_METHOD_EXSCRIPT',1)} [See example](#)

Displays the scroll bars in the document window.

Syntax

[objectreference].ShowScrollBars(ScrollVert,ScrollHorz)

Parameters

ScrollVert

Data type is Boolean.

ScrollHorz

Data type is Boolean.

Return value

Usage

Word Pro: ShowStatusBar method

{button ,AL('H_STATUSBAR_CLASS',0)} [See list of classes](#)

{button ,AL('H_SHOWSTATUSBAR_METHOD_EXSCRIPT',1)} [See example](#)

Displays the status bar.

Syntax

[objectreference].ShowStatusBar()

Parameters

Return value

Integer

Usage

Word Pro: Show method

{button ,AL('H_ICONBAR_CLASS;H_DOCWINDOW_CLASS',0)} [See list of classes](#)

{button ,AL('H_SHOW_METHOD_EXSCRIPT',1)} [See example](#)

Displays an object, such as an icon bar set or a document.

Syntax

[objectreference].Show()

Parameters

Data type is Integer. The legal values for this parameter are -1 or 0 but you may use the LotusScript constants of True (-1) and False (0).

Return value

The return value for this method will always be -1 or 0. When testing the return value, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Word Pro: SimulateButtonClick method

{button ,AL(^H_STATUSBARBUTTON_CLASS',0)} [See list of classes](#)

{button ,AL(^H_SIMULATEBUTTONCLICK_METHOD_EXSCRIPT',1)} [See example](#)

Simulates a user click on the button. The StatusBarButtonClicked event will be emitted.

Syntax

[objectreference].SimulateButtonClick()

Parameters**Return value**

Integer. Always returns True.

Usage

Use this method to simulate a user click on the button. This method forces a StatusBarButtonClicked event.

Word Pro: Skip method

{button ,AL('H_CLICKHERE_CLASS;H_TEXTMARKER_CLASS;H_TEXT_CLASS',0)} [See list of classes](#)

Syntax

[objectreference].Skip(SkipType,SkipMeaning)

Parameters

SkipType

Data type is Variant. The value of this parameter must be one of the strings below or its code equivalent.

\$LwpSkipTypeFormatCheck (1767)

\$LwpSkipTypeRevision (1765)

\$LwpSkipTypeSpell (1766)

SkipMeaning

Data type is Variant. The value of this parameter must be one of the strings below or its code equivalent.

\$LwpSkipMeanRevSkipBackwordWord (1763)

\$LwpSkipMeanRevSkipForwardWord (1762)

\$LwpSkipMeanSpellMarkWord (1764)

Return value

Usage

Word Pro: SmallCaps method

{button ,AL(`H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL(`H_SMALLCAPS_METHOD_EXSCRIPT',1)} [See example](#)

Sets the small caps attribute for selected text, or all following text if no text is selected. Acts as a toggle, turning the attribute off if it is on and on if it is off. Equivalent to choosing Text - Attributes - Other, and then "Small Caps" in the Attributes box.

Syntax

[objectreference].SmallCaps()

Parameters

None.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

Word Pro: SmartSumColumn method

{button ,AL('H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL('H_SMARTSUMCOLUMN_METHOD_EXSCRIPT',1)} [See example](#)

Inserts a SmartSum formula in the currently active table cell.

Syntax

[objectreference].SmartSumColumn()

Parameters

None.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

The SmartSum formula adds the values of cells above and in the same column as the SmartSum cell, and displays the total value in the SmartSum cell. If the column contains a cell with a text value, the SmartSum formula only totals the cells between the SmartSum cell and the cell with the text.

Word Pro: SmartSumRow method

{button ,AL('H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL('H_SMARTSUMROW_METHOD_EXSCRIPT',1)} [See example](#)

Inserts a SmartSum formula in the currently active table cell.

Syntax

[objectreference].SmartSumRow()

Parameters

None.

Return value

This method returns a value of -1 (True) or 0 (False) indicating that the method succeeded or failed respectively.

Usage

The SmartSum formula adds the values of cells to the left of and in the same row as the SmartSum cell, and displays the total value in the SmartSum cell. If the row contains a cell with a text value, the SmartSum formula only totals the cells between the SmartSum cell and the cell with the text.

```
'Example: RevisionAcceptLayoutChange method
' This example accepts all revision markup changes made to the current
' layout object.
' RUNTIME DEPENDENCIES: You must have a document open for this script to work.

.Layout.RevisionAcceptLayoutChange True, "Mark Osborne"
```



```
'Example: RevisionCancelLayoutChange method
' This example cancels all revision markup changes made to the current
' layout object.
' RUNTIME DEPENDENCIES: You must have a document open for this script to work.

.Layout.RevisionCancelLayoutChange True, ""
```

'Example: RevisionMarkMode property
'This example enables revision marking, inserts some text, then disables
'revision marking.

```
.ActiveDocument.RevisionMarkMode = 1  
.Type "This text is marked for revision insertion. "  
.ActiveDocument.RevisionMarkMode = 0  
.Type "This text is not."
```

'Example: RightExternalMargin property

'This example creates a frame, then changes the padding around the frame's
'border to 1/4 inch (360 twips).

```
.NewFrame 4320, 4320, 3387, 1992, "Default Frame"
```

```
.Frame.Layout.LeftExternalMargin = 360
```

```
.Frame.Layout.TopExternalMargin = 360
```

```
.Frame.Layout.RightExternalMargin = 360
```

```
.Frame.Layout.BottomExternalMargin = 360
```

```
'Example: Right property
'This example types a paragraph of text and indents the paragraph from the right 1/2
inch.
.NewDocument
For i = 1 To 20
    .type "Indention test "
Next
.Text.Indent.Right = 720 ' right indent everything 1/2 inch.
MessageBox "Click OK to revert to the style indents.", MB_OK, "Example Script"
.Text.Indent.RevertToStyle
```

```
'Example: RunMacroOnDocEvents property
'This example asks the user whether to run scripts set to run when documents are
opened,
'and then sets the appropriate option.

stat = MessageBox ("Do you want to run scripts when opening documents?", 36, "Example
Script")
If stat = 6 Then ' user said yes
    .ApplicationWindow.UserInterfacePrefs.RunMacroOnDocEvents = True
Else
    .ApplicationWindow.UserInterfacePrefs.RunMacroOnDocEvents = False
End If
```

```
'Example: RunMacroOnLoad property
'This example asks the user whether to run application startup scripts,
'and then sets the appropriate option.

stat = MessageBox ("Do you want to run scripts when starting Word Pro?", 36, "Example
Script")
If stat = 6 Then ' user said yes
    .ApplicationWindow.UserInterfacePrefs.RunMacroOnLoad = True
Else
    .ApplicationWindow.UserInterfacePrefs.RunMacroOnLoad = False
End If
```

```
'Example: RunScript method  
' This example runs the 'TestFunction' module in the script file 'EXAMPLE.LWP'  
' RUNTIME DEPENDENCIES: You must have a script file named 'EXAMPLE.LWP' which  
' contains a function named 'TestFunction'. The script file should be  
' located in the WordPro preferences Script directory.
```

```
Dim ScriptModule as String  
Dim ScriptFunction as String
```

```
ScriptModule = "EXAMPLE.LWP"  
ScriptFunction = "TestFunction"
```

```
.RunScript ScriptModule, ScriptFunction
```

```
'Example: SaveAs method
' This example saves the current document in ASCII format. The file is saved
' in Word Pro's document directory using the file name given returned by the
' input box.
' RUNTIME DEPENDENCIES: You must have a document open for this script to work.
Dim FileName as String
Dim FilePath as String
Dim FilterType as String
FilterType = "Text"
FilePath = .ApplicationWindow.UserInterfacePrefs.DocPath
FileName = InputBox("Enter a filename to save as:", "Example Script", "")
If FileName <> "" Then
    .SaveAs FileName, FilePath , FilterType, False, True, False
End If
```


'Example: SaveDivisions property

'This example asks the user if each division in the document should
'be saved to a separate HTML file when exporting, then sets
'the appropriate option.

'

'Note that to set this option, the option to save the entire document must also
'be set to True.

```
If .ApplicationWindow.UserInterfacePrefs.HTMLOptions.EntireDocument = True Then
    stat = MessageBox ("Do you want save divisions in separate files?", 36, "Example
Script")
    If stat = 6 Then ' user said yes
        .ApplicationWindow.UserInterfacePrefs.HTMLOptions.SaveDivisions = True
    Else
        .ApplicationWindow.UserInterfacePrefs.HTMLOptions.SaveDivisions = False
    End If
Else ' we're only saving the current division, so turn the option off
    .ApplicationWindow.UserInterfacePrefs.HTMLOptions.SaveDivisions = False

End If
```

'Example: SaveDivision method

'This example examines all the divisions in the current document, looking
'for an external division. If an external division is found, the external file
'is saved.

```
Forall div In .ActiveDocument.Divisions
  If div.IsDivisionExternal Then
    div.SaveDivision
  End If
End Forall
```

```
'Example: SaveGlossary method
' This example stores a file name in the variable GlossFileName, hides the
' open documents, opens the default Word Pro glossary file, creates and saves
' a glossary file named "GLOSTST.GLS" in the User Setup glossary directory,
' then closes the glossary files and resets the default values user interface
' preferences.
' RUNTIME DEPENDENCIES: You must have create file rights in the specified
' glossary directory for this script to work.
```

```
Dim GlossFileName As String
Dim OldGlossFileName as String
OldGlossFileName = .ApplicationWindow.UserInterfacePrefs.GlossaryDataFileName
GlossFileName = "GLOSTST.GLS"
.ApplicationWindow.UserInterfacePrefs.OpenDocsVisible = False
.GlossaryOpen OldGlossFileName, "Lotus Word Pro"

.CreateGlossary

.SaveGlossary GlossFileName, "Lotus Word Pro", False
.Close
.ApplicationWindow.UserInterfacePrefs.IsReplacement = False
.ApplicationWindow.UserInterfacePrefs.OpenDocsVisible = True
.ApplicationWindow.UserInterfacePrefs.OpenReadOnly = False
```

'Example: SaveInternetFile property

'This example creates a document, then lets the user specify an FTP server

'and filename, then saves the document to the specified location.

.NewDocument

.Text.InsertText "This is a test file which will be saved to FTP.", True

Server = Inputbox ("What server and directory would you like to use?", "Example
Script")

Fname = Inputbox ("What name would you like to give the file?", "Example Script")

' this parameter should take the form of "ftp://<server>/<dir>/<Filename>"

Destination = "ftp://" & Server & "/" & fname

.SaveInternetFile Destination, "HTML"

'Example: SavePath property

'This example sets the option for exporting graphics to HTML so that all exported graphics are stored in a single directory.

```
.ApplicationWindow.UserInterfacePrefs.HTMLOptions.SavePath = 1
```

'Example: SaveSections property

'This example asks the user if each section in the document should be saved to
'a separate HTML file when exporting, then sets the appropriate option.

'Note that in order to save sections as separate files, the option to save
'divisions as separate files must also be set, or we must be saving the
'current division only.

```
If .ApplicationWindow.UserInterfacePrefs.HTMLOptions.SaveDivisions = True or _  
.ApplicationWindow.UserInterfacePrefs.HTMLOptions.EntireDocument = False Then  
    stat = MessageBox ("Do you want save sections in separate files?", 36, "Example  
Script")  
    If stat = 6 Then    ' user said yes  
        .ApplicationWindow.UserInterfacePrefs.HTMLOptions.SaveSections = True  
    Else  
        .ApplicationWindow.UserInterfacePrefs.HTMLOptions.SaveSections = False  
    End If  
Else ' we can't save separate sections, so turn the option off  
    .ApplicationWindow.UserInterfacePrefs.HTMLOptions.SaveSections = False  
  
End If
```

'Example: SaveSnapShot property

'This example script disables storing copies of graphic images with the document when it is saved.

.Application.Preferences.**SaveSnapShot** = \$LwpSnapShotNosave

'Example: SaveUserDefaults method

' This example changes the number of recent files to 5.

' RUNTIME DEPENDENCIES: You must have a document open for this script to work.

.ApplicationWindow.UserInterfacePrefs.NumOfRecentFiles = 5

.ApplicationWindow.SaveUserDefaults


```
'Example: SaveVersion method
' This example saves the selected version as a file named TESTVER.LWP.
' RUNTIME DEPENDENCIES: You must have a document open for this script to work.

Dim FileName As String
Dim Version As Long
FileName = .ApplicationWindow.UserInterfacePrefs.DocPath & "\TESTVER.LWP"
Version = 2
.SaveVersion FileName, Version
```

'Example: Save method

' This example saves the current document.

' RUNTIME DEPENDENCIES: You must have a document open for this script to work.

.Save

'Example: ScaleHeight property

'This example imports a graphic into the current document, then sizes the graphic to two inches wide by one inch high.

```
GPath = .ApplicationWindow.UserInterfacePrefs.GraphicPath
.ImportGraphic GPath & "\_wpdon.gif", ".gif", True, False, "Default Graphic/OLE"
.Layout.ScaleMode = $LwpScaleTypeCustom
.Layout.ScaleWidth = 2880 ' 2880 twips = 2 inches
.Layout.ScaleHeight = 1440 ' 1440 twips = 1 inch
```

'Example: ScaleMode property

'This example imports a graphic into the current document, then sets

'graphic scaling to original size.

```
GPath = .ApplicationWindow.UserInterfacePrefs.GraphicPath
```

```
.ImportGraphic GPath & "\_wpdon.gif", ".gif", True, False, "Default Graphic/OLE"
```

```
.Layout.ScaleMode = $LwpScaleTypeOriginalSize
```

'Example: ScalePercentage property

'This example imports a graphic into the current document, then scales the graphic to 152.5 percent of its original size.

```
GPath = .ApplicationWindow.UserInterfacePrefs.GraphicPath
.ImportGraphic GPath & "\_wpdon.gif", ".gif", True, False, "Default Graphic/OLE"
.Layout.ScaleMode = $LwpScaleTypePercentage
.Layout.ScalePercentage = 1525
```

'Example: ScaleWidth property

'This example imports a graphic into the current document, then sizes the graphic to two inches wide by one inch high.

```
GPath = .ApplicationWindow.UserInterfacePrefs.GraphicPath
.ImportGraphic GPath & "\_wpdon.gif", ".gif", True, False, "Default Graphic/OLE"
.Layout.ScaleMode = $LwpScaleTypeCustom
.Layout.ScaleWidth = 2880 ' 2880 twips = 2 inches
.Layout.ScaleHeight = 1440 ' 1440 twips = 1 inch
```

```
'Example: ScreenPositionX property
Dim CR As String*1
Dim IcnPallet As String
Dim MsgStr As String
Dim IcnMgr As IconBarManager

IcnPallet = "Comment Tools"
CR = Chr(10)
Set IcnMgr = .ApplicationWindow.IconBarManager

With IcnMgr.IconBars(IcnPallet)
    MsgStr = "Height = " & .Height & CR
    MsgStr = MsgStr & "IconBarPositionState = " & .IconBarPositionState & CR
    MsgStr = MsgStr & "PositionType = " & .PositionType & CR
    MsgStr = MsgStr & "ScreenPositionX = " & .ScreenPositionX & CR
    MsgStr = MsgStr & "ScreenPositionY = " & .ScreenPositionY
    MessageBox MsgStr, 64, "Script Example - " & .Name
End With
```

```
'Example: ScreenPositionY property
Dim CR As String*1
Dim IcnPallet As String
Dim MsgStr As String
Dim IcnMgr As IconBarManager

IcnPallet = "Comment Tools"
CR = Chr(10)
Set IcnMgr = .ApplicationWindow.IconBarManager

With IcnMgr.IconBars(IcnPallet)
    MsgStr = "Height = " & .Height & CR
    MsgStr = MsgStr & "IconBarPositionState = " & .IconBarPositionState & CR
    MsgStr = MsgStr & "PositionType = " & .PositionType & CR
    MsgStr = MsgStr & "ScreenPositionX = " & .ScreenPositionX & CR
    MsgStr = MsgStr & "ScreenPositionY = " & .ScreenPositionY
    MessageBox MsgStr, 64, "Script Example - " & .Name
End With
```


'Example: SectionName property
'This example creates a new section, then gets it's internal name.
'The section is then renamed to "Test Section".

```
Dim ThisSection as String  
.InsertSection "Default Page", True, True, $LwpStartTypeNextpage, False, True  
ThisSection = .Text.SectionName  
.Division.Foundry.Sections(ThisSection).UserName = "Test Section"
```

'Example: SectionUserName property

'This example creates a new section.

'The name of the section is then inserted into the document.

.InsertSection "Default Page", True, True, 1, 0, 1, "Example Section", 12632256

.Text.InsertText "The name of this section is " & .Text.**SectionUserName** & ". ", True

```
'Example: SelectCell method
' This example creates a table with 5 columns and 4 rows based on the
' Default Table' style. The current cell is then selected.
' RUNTIME DEPENDENCIES: You must have a document open for this script to work.

.CreateTable False, "Default Table", 5, 4
.Text.InsertText "A table cell"
.SelectCell
```

'Example: SelectColumn method

' This example creates a table with 5 columns and 4 rows based on the

' Default Table style. The first column is selected.

' RUNTIME DEPENDENCIES: You must have a document open for this script to work.

.CreateTable False, "Default Table", 5, 4

.SelectColumn

'Example: SelectDoc method

'This example inserts some text into the current document, then selects
'the entire document.

```
.Text.InsertText "This is some sample text which will be selected.", True  
.SelectDoc
```

```
'Example: SelectedPages property
'Print pages 1, 3, 4 and 5 from README95.LWP.
.OpenDocument "readme.lwp", "", "Word Pro"
' select the pages
.ActiveDocument.PrintSettings.SelectedPages = "1,3-5"
' Must set PrintRange property so that settings take effect.
.ActiveDocument.PrintSettings.PrintRange = $LtsPrintRangeSelectedPages
.Print ' and print the document.
```

```
'Example: SelectEntireCellRange method
' This example creates a table with 5 columns and 5 rows, inserts some text
' and then selects the inserted text.
' RUNTIME DEPENDENCIES: You must have a document open for this script to work.

.CreateTable False, "Default Table", 5,5
.Table.CellLayout(1,1).GotoLayout
.Type "Some text for the table cell"
.SelectEntireCellRange
```

```
'Example: SelectEntireColumn method
' This example creates a table with 5 columns and 4 rows based on the
' Default Table style and then selects the current column.
' RUNTIME DEPENDENCIES: You must have a document open for this script to work.

.CreateTable False, "Default Table", 5, 4
.Text.InsertText "A table cell"
.SelectEntireColumn
```



```
'Example: SelectEntirePColCellRange method
' This example creates a parallel column table, inserts some text and then
' selects the inserted text and cell.
' RUNTIME DEPENDENCIES: You must have a document open for this script to work.

.CreateParallelColumns 3, $LtsAlignmentHorizCenter
.Type "Some text for the cell"
.SelectEntirePColCellRange
```

'Example: SelectEntirePColColumn method

' This example creates a parallel column table with 3 columns and then selects
' the first column.

' RUNTIME DEPENDENCIES: You must have a document open for this script to work.

.CreateParallelColumns 3, \$LtsAlignmentHorizCenter

.SelectEntirePColColumn

'Example: SelectEntirePColRow method

' This example creates a parallel column table with 3 columns and then selects
' the first row.

' RUNTIME DEPENDENCIES: You must have a document open for this script to work.

.CreateParallelColumns 3, \$LtsAlignmentHorizCenter

.SelectEntirePColRow

'Example: SelectEntirePCol method

' This example creates a parallel column table with 3 columns and then selects
' the first column.

' RUNTIME DEPENDENCIES: You must have a document open for this script to work.

.CreateParallelColumns 3, \$LtsAlignmentHorizCenter

.SelectEntirePCol

```
'Example: SelectEntireRow method
' This example creates a table with 5 columns and 4 rows based on the
' Default Table' style and then selects the current row.
' RUNTIME DEPENDENCIES: You must have a document open for this script to work.

.CreateTable False, "Default Table", 5, 4
.Text.InsertText "A table cell"
.SelectEntireRow
```

```
'Example: SelectEntireTable method
' This example creates a table with 5 columns and 4 rows based on the
' Default Table' style and then selects the entire table.
' RUNTIME DEPENDENCIES: You must have a document open for this script to work.

.CreateTable False, "Default Table", 5, 4
.Text.InsertText "A table cell"
.SelectEntireTable
```

```
'Example: SelectParagraph method
' This example inserts text into the current document and selects the
' current paragraph.
' RUNTIME DEPENDENCIES: You must have a document open for this script to work.

.Text.InsertText "This is some sample text."
.SelectParagraph
```

'Example: SelectPColCell method

' This example creates a parallel column table with 3 columns and then selects
' the first cell.

' RUNTIME DEPENDENCIES: You must have a document open for this script to work.

```
.CreateParallelColumns 3, $LtsAlignmentHorizCenter  
.SelectPColCell
```


'Example: SelectRow method

' This example creates a table with 5 columns and 4 rows based on the

' Default Table style. The first row is selected.

' RUNTIME DEPENDENCIES: You must have a document open for this script to work.

.CreateTable False, "Default Table", 5, 4

.SelectRow

'Example: SelectSection method

' This example selects the contents of the current section of the document.

' RUNTIME DEPENDENCIES: You must have a document open for this script to work.

.SelectSection

```
'Example: SelectSentence method
' This example inserts some text into the current document and positions the
' cursor at the line's begininning. The entire sentence is then selected.
' RUNTIME DEPENDENCIES: You must have a document open for this script to work.

.Text.InsertText "Some sample text."
.Text.Backward $LwpNavigateObjectTypeSentence, 1

.SelectSentence
```

```
'Example: SelectTableItem method
' This example creates a table with 5 rows and 5 columns and then selects the
' entire table.
' RUNTIME DEPENDENCIES: You must have a document open for this script to work.

.CreateTable False, "Default Table", 5,5
.Table.SelectTableItem($LwpTableSelectionTable)
```

```
'Example: SelectTable method
' This example creates a table with 5 columns and 4 rows based on the
' Default Table' style and then selects the table.
' RUNTIME DEPENDENCIES: You must have a document open for this script to work.

.CreateTable False, "Default Table", 5, 4
.Text.InsertText "A table cell"
.SelectTable
```

```
'Example: SelectWord method
' This example inserts some text into the current document and positions the
' cursor at the line's begininning. The first word is then selected.
' RUNTIME DEPENDENCIES: You must have a document open for this script to work.
.Text.InsertText "Some sample text."
.Text.Backward $LwpNavigateObjectTypeSentence, 1

.SelectWord
```

'Example: Select method

' This example inserts some text into the current document which is then selected.

' RUNTIME DEPENDENCIES: You must have a document open for this script to work.

.Text.InsertText "This is some text."

.Select \$LwpSelectObjectTypeParagraph

Word Pro: ShowMarks property

{button ,AL('H_CLICKHERE_CLASS',0)} [See list of classes](#)

{button ,AL('H_SHOWMARKS_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write)

This language element has not yet been defined. You can check for a newer Help file on the Lotus web site.

Data Type

Integer

Syntax

showmarksvalue = [objectreference].ShowMarks

[objectreference].ShowMarks = showmarksvalue

Legal values

The legal values for this property are -1 and 0 but you may use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Word Pro: ShowMenuCustomization property

{button ,AL('H_USERINTERFACEPREFS_CLASS','0)} [See list of classes](#)

{button ,AL('H_SHOWMENUCUSTOMIZATION_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write)

This language element has not yet been defined. You can check for a newer Help file on the Lotus web site.

Data Type

Integer

Syntax

showmenucustomizationvalue = [objectreference].ShowMenuCustomization

[objectreference].ShowMenuCustomization = showmenucustomizationvalue

Legal values

The legal values for this property are -1 and 0 but you may use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Word Pro: ShowMenuWarningMessage property

{button ,AL('H_USERINTERFACEPREFS_CLASS',0)} [See list of classes](#)

(Read-write)

This language element has not yet been defined. You can check for a newer Help file on the Lotus web site.

Data Type

Integer

Syntax

showmenuwarningmessagevalue = [objectreference].ShowMenuWarningMessage

[objectreference].ShowMenuWarningMessage = showmenuwarningmessagevalue

Legal values

The legal values for this property are -1 and 0 but you may use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Word Pro: ShowNoWelcomeBox property

{button ,AL('H_USERINTERFACEPREFS_CLASS',0)} [See list of classes](#)

{button ,AL('H_SHOWNOWELCOMEBOX_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Determines whether the Welcome dialog box is displayed.

Data Type

[Integer](#)

Syntax

shownowelcomeboxvalue = [objectreference].ShowNoWelcomeBox

[objectreference].ShowNoWelcomeBox = shownowelcomeboxvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values. Default is False (0).

Usage

Equivalent to the "Welcome Dialog" option in the "Disable" field on the General panel of the Word Pro Preferences dialog box. If this property is set to True (-1), Word Pro does not display the Welcome dialog box. If set to False (0), Word Pro displays the Welcome dialog box.

Word Pro: ShowStatistics property

{button ,AL('H_GRAMMAR_CLASS',0)} [See list of classes](#)

{button ,AL('H_SHOWSTATISTICS_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) A flag that determines the compilation of document statistics that displays when Grammar Check finishes on or off.

Data Type

Integer

Syntax

showstatisticsvalue = [objectreference].ShowStatistics

[objectreference].ShowStatistics = showstatisticsvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Sets the flag on or off that displays the document statistics when Grammar Check completes. This is automatically set to not display.

Word Pro: ShowTabs property

{button ,AL(^H_DIVISIONINFO_CLASS;H_SECTION_CLASS;H_INDEXSECTION_CLASS',0)} [See list of classes](#)

{button ,AL(^H_SHOWTABS_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write)

Data Type

[Integer](#)

Syntax

showtabsvalue = [objectreference].ShowTabs

[objectreference].ShowTabs = showtabsvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

{button ,AL(^H_SHOWDIVISIONTABS_PROPERTY_MEMDEF',0)} [See related topics](#)

Word Pro: SilverBullets property

{button ,AL('H_FOUNDRY_CLASS',0)} [See list of classes](#)

(Read-only) An object created from the SilverBulletCollection class. This object provides access to SilverBullet objects.

Data Type

[SilverBulletCollection](#)

Syntax

silverbulletsvalue = [objectreference].SilverBullets

Legal values

Always contains an instance of the SilverBulletCollection class.

Usage

When accessed through the Foundry property on a Division object, the collection object in this property provides access to all the SilverBullet objects contained in that Division object.

When accessed through the AppFoundry property on the WPAApplication object, this collection object provides access to all the SilverBullet objects contained in the Word Pro Clipboard.

When accessed through the TempFoundry property on the WPAApplication object, this collection object provides access to all the SilverBullet objects placed in TempFoundry by WordPro or a script.

When accessed through the Foundry property on the WPAApplication object, this collection object provides access to all the SilverBullet objects contained in the currently active Division object.

This property is not used in the Foundry property on the TextDocument object.

For more information about collection classes, see [Overview: Word Pro LotusScript Collection Classes](#).

Word Pro: SilverBullet property

{button ,AL(^H_BULLET_CLASS',0)} [See list of classes](#)

(Read-only)

Data Type

[SilverBullet](#)

Syntax

silverbulletvalue = [objectreference].SilverBullet

Legal values

Always contains an instance of the SilverBullet class.

Usage

Word Pro: SingleCellSelected property

{button ,AL(^H_BASSETABLE_CLASS;H_FOOTNOTETABLE_CLASS;H_PARALLELCOLUMNS_CLASS;H_GLOSSARY_CLASS;H_TABLEHEADING_CLASS;H_TABLE_CLASS',0)} [See list of classes](#)

(Read-only) Indicates whether a single cell is currently selected.

Data Type

Integer

Syntax

singlecellselectedvalue = [objectreference].SingleCellSelected

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

A value of True in this property indicates that a single cell is selected, or that the cursor is currently within a cell. A value of False indicates that multiple cells are currently selected.

Word Pro: SizeStyleName property

{button ,AL('H_CHARACTERSTYLE_CLASS;H_PARAGRAPHSTYLE_CLASS',0)} [See list of classes](#)

(Read-write) The size information for a font.

Data Type

String

Syntax

sizestylevalue = [objectreference].SizeStyleName

[objectreference].SizeStyleName = sizestylevalue

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Word Pro: Size property

{button ,AL('H_FONT_CLASS',0)} [See list of classes](#)

{button ,AL('H_SIZE_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write)

Data Type

[Font]

[Single](#)

[Index]

[Long](#)

Syntax

sizevalue = [objectreference].Size

[objectreference].Size = sizevalue

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Word Pro: SizingUnitName property

{button ,AL('H_USERINTERFACEPREFS_CLASS',0)} [See list of classes](#)

(Read-only) The name and abbreviation of the name of the units in the SizingUnits property.

Data Type

String

Syntax

sizingunitnamevalue = [objectreference].SizingUnitName

Legal values

The legal values for this property are listed below under Usage.

Usage

This property corresponds to the [SizingUnits property](#).

<u>SizingUnits property value</u>	<u>SizingUnitName property value</u>
\$LtsScaleModeCentimeter (1056964840)	Centimeters (cm)
\$LtsScaleModeInch (1056964838)	Inches (in)
\$LtsScaleModePoint (1056964837)	Points (pts)
\$LtsScalePica (1728)	Picas (pi)

Word Pro: SizingUnits property

{button ,AL('H_USERINTERFACEPREFS_CLASS',0)} [See list of classes](#)

{button ,AL('H_SIZINGUNITS_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) The default scaling unit chosen by the user in the Word Pro Preferences dialog box.

Data Type

Data type is [Variant](#) which allows the value of this property to be one of the constants listed below or its numeric equivalent (in parentheses).

Syntax

sizingunitsvalue = [objectreference].SizingUnits

[objectreference].SizingUnits = sizingunitsvalue

Legal values

<u>Value</u>	<u>Effect</u>
\$LtsScaleModeCentimeter (1056964840)	Default scaling units are centimeters.
\$LtsScaleModeInch (1056964838)	Default scaling units are inches.
\$LtsScaleModePoint (1056964837)	Default scaling units are points.
\$LwpScaleModePica (1728)	Default scaling units are picas.

Usage

Equivalent to the options in the "Measure in" list box on the General panel of the Word Pro Preferences dialog box.

Word Pro: Skipped property

{button ,AL('H_BULLET_CLASS',0)} [See list of classes](#)

{button ,AL('H_SKIPPED_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write)

Data Type

Variant (Enumerated)

CommandState

Syntax

skippedvalue = [objectreference].Skipped

[objectreference].Skipped = skippedvalue

Legal values

\$LwpCommandStateOff (151)

\$LwpCommandStateOn (152)

\$LwpCommandStateStyle (153)

Usage

Word Pro: SkipWordMode property

{button ,AL('H_ATTRIBUTES_CLASS',0)} [See list of classes](#)

(Read-write)

Data Type

Integer

Syntax

skipwordmodevalue = [objectreference].SkipWordMode

[objectreference].SkipWordMode = skipwordmodevalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Word Pro: SmallCaps property

{button ,AL('H_FONT_CLASS;H_FONTMETRICS_CLASS',0)} [See list of classes](#)

{button ,AL('H_SMALLCAPS_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write)

Data Type

[Integer](#)

Syntax

smallcapsvalue = [objectreference].SmallCaps

[objectreference].SmallCaps = smallcapsvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Word Pro: SmallFileFormat property

{button ,AL('H_PREFERENCES_CLASS',0)} [See list of classes](#)

{button ,AL('H_SMALLFILEFORMAT_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write)

Data Type

[Integer](#)

Syntax

smallfileformatvalue = [objectreference].SmallFileFormat

[objectreference].SmallFileFormat = smallfileformatvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Word Pro: SmartCorrects property

{button ,AL(^H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL(^H_SMARTCORRECTS_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-only) An instance of the SmartCorrectCollection class.

Data Type

[SmartCorrectCollection](#)

Syntax

smartcorrectsvalue = [objectreference].SmartCorrects

Legal values

Always contains an instance of the SmartCorrectCollection class.

Usage

Word Pro: SmartCorrect property

{button ,AL('H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL('H_SMARTCORRECT_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-only) The SmartCorrect object for the currently active session of Word Pro.

Data Type

[SmartCorrect](#)

Syntax

smartcorrectvalue = [objectreference].SmartCorrect

Legal values

Always contains an instance of the SmartCorrect class.

Usage

The properties in this SmartCorrect object contain the SmartCorrect settings for Word Pro.

{button ,AL('H_ISSMARTCORRECTENABLED_PROPERTY_MEMDEF',0)} [See related topics](#)

Word Pro: SmartFill property

{button ,AL('H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

{button ,AL('H_SMARTFILL_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) An instance of the SmartFillCollection class. Use this object to access the SmartFill object for the current session of Word Pro.

Data Type

[SmartFillCollection](#)

Syntax

[objectreference].SmartFill = smartfillvalue

smartfillvalue = [objectreference].SmartFill

Legal values

Always contains an instance of the SmartFillCollection class.

Usage

The SmartFill property contains an instance of the SmartFill collection class. The SmartFill collection represents one SmartFill object for each language Word Pro supports. You iterate through the collection members by specifying the language number you want to manipulate. A list of language numbers is in the description of the KeyboardLanguage property in LotusScript Help.

Word Pro: SmartLabelsEnabled property

{button ,AL('H_PREFERENCES_CLASS;',0)} [See list of classes](#)

{button ,AL('H_SMARTLABELSENABLED_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write)

This language element has not yet been defined. You can check for a newer Help file on the Lotus web site.

Data Type

Integer

Syntax

smartlabelsenabledvalue = [objectreference].SmartLabelsEnabled

[objectreference].SmartLabelsEnabled = smartlabelsenabledvalue

Legal values

The legal values for this property are -1 and 0 but you may use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Word Pro: SmartLevel property

{button ,AL(^H_NUMBERING_CLASS',0)} [See list of classes](#)

(Read-write)

Data Type

Variant (Enumerated)

CommandState

Syntax

smartlevelvalue = [objectreference].SmartLevel

[objectreference].SmartLevel = smartlevelvalue

Legal values

\$LwpCommandStateOff (151)

\$LwpCommandStateOn (152)

\$LwpCommandStateStyle (153)

Usage

Word Pro: SnapShotSaveOptions property

{button ,AL(^H_PREFERENCES_CLASS',0)} [See list of classes](#)

(Read-only)

Data Type

[StringCollection](#)

Syntax

snapshotsaveoptionsvalue = [objectreference].SnapShotSaveOptions

Legal values

Always contains an instance of the StringCollection class.

Usage

Word Pro: SortLevel1 property

{button ,AL('H_SORTOPTIONS_CLASS',0)} [See list of classes](#)

{button ,AL('H_SORTLEVEL1_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-only) Allows you to access options for the primary sort order of a multi-level sort.

Data Type

[SortKey](#)

Syntax

[objectreference].SortLevel1 = sortlevel1value

sortlevel1value = [objectreference].SortLevel1

Legal values

Always contains an instance of the SortKey class.

Usage

Word Pro: SortLevel2 property

{button ,AL('H_SORTOPTIONS_CLASS',0)} [See list of classes](#)

{button ,AL('H_SORTLEVEL2_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Allows you to access options for the secondary sort order of a multi-level sort.

Data Type

[SortKey](#)

Syntax

[objectreference].SortLevel2 = sortlevel2value

sortlevel2value = [objectreference].SortLevel2

Legal values

Always contains an instance of the SortKey class.

Usage

Word Pro: SortLevel3 property

{button ,AL(^H_SORTOPTIONS_CLASS',0)} [See list of classes](#)

(Read-write) Allows you to access options for the tertiary sort order of a multi-level sort.

Data Type

[SortKey](#)

Syntax

[objectreference].SortLevel3 = sortlevel3value

sortlevel3value = [objectreference].SortLevel3

Legal values

Always contains an instance of the SortKey class.

Usage

Word Pro: SortNumbers property

{button ,AL(^H_SORTOPTIONS_CLASS',0)} [See list of classes](#)

(Read-write) Specifies whether words or numbers are positioned first or last when doing an alphabetical sort.

Data Type

Variant (Enumerated)

SortNumberOrder

Syntax

sortnumbersvalue = [objectreference].SortNumbers

[objectreference].SortNumbers = sortnumbersvalue

Legal values

<u>Value</u>	<u>Effect</u>
\$LwpSortNumberOrderFirst (1770)	Specifies if a word or number is positioned first in a row.
\$LwpSortNumberOrderLast (1771)	Specifies if a word or number is positioned last in a row

Usage

This property specifies whether words or numbers are positioned first or last in a row. For example, if you select "First" in a Last-Name First-Name "Name" field, Word Pro lists the names alphabetically by last name. If you select "Last" in a Last-Name First-Name "Name" field, Word Pro lists the names alphabetically by first name.

Equivalent to choosing Text - Sort and selecting an option for "Sort numbers."

Word Pro: SortOptions property

{button ,AL(^H_DIVISION_CLASS;H_TEXTDOCUMENT_CLASS',0)} [See list of classes](#)

(Read-only)

Data Type

[SortOptions](#)

Syntax

sortoptionsvalue = [objectreference].SortOptions

Legal values

Always contains an instance of the SortOptions class.

Usage

Word Pro: SortOrder property

{button ,AL(^H_SORTKEY_CLASS',0)} [See list of classes](#)

(Read-write) Allows you to specify an ascending (A - Z or 0 - 9), or a descending (Z - A or 9 - 0) sort order.

Data Type

Data type is Variant which allows the value of this property to be one of the constants listed below or its numeric equivalent (in parentheses).

Syntax

sortordervalue = [objectreference].SortOrder

[objectreference].SortOrder = sortordervalue

Legal values

<u>Value</u>	<u>Effect</u>
\$LtsSortAscending (1056964770)	Sorts the data from A to Z or 0 to 9.
\$LtsSortDescending (1056964772)	Sorts the data from Z to A or 9 to 0.

Usage

Equivalent to choosing Text - Sort and selecting "Ascending" or "Descending" in the desired sort level "Order" box.

Word Pro: SortType property

{button ,AL(^H_SORTKEY_CLASS',0)} [See list of classes](#)

(Read-write) Specifies whether the sorting for a level is done numerically or alphanumerically.

Data Type

Data type is Variant which allows the value of this property to be one of the constants listed below or its numeric equivalent (in parentheses).

Syntax

sorttypevalue = [objectreference].SortType

[objectreference].SortType = sorttypevalue

Legal values

<u>Value</u>	<u>Effect</u>
\$LwpSortTypeAlphanumeric (1772)	Indicates that the sort will be done alphanumerically.
\$LwpSortTypeNumeric (1773)	Indicates that the sort will be done numerically.

Usage

Equivalent to choosing Text - Sort and selecting either "Alphanumeric" or "Numeric" in the "Type" box for a specific sort level.

Word Pro: SortWordOption property

{button ,AL(`H_SORTKEY_CLASS',0)} [See list of classes](#)

{button ,AL(`H_SORTWORDOPTION_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Specifies whether to sort text by all words, the first or second word, or by any word you select.

Data Type

Data type is [Variant](#) which allows the value of this property to be one of the constants listed below or its numeric equivalent (in parentheses).

Syntax

sortwordoptionvalue = [objectreference].SortWordOption

[objectreference].SortWordOption = sortwordoptionvalue

Legal values

<u>Value</u>	<u>Effect</u>
\$LwpSortWhichWordAllwords (1776)	Sorts on every character in the chosen field.
\$LwpSortWhichWordFirstword (1774)	Sorts on the first word of the chosen field.
\$LwpSortWhichWordLastword (1775)	Sorts on the last word of the chosen field.
\$LwpSortWhichWordOther (1777)	Sorts on the word selected in the "Sort on a Word" dialog box.

Usage

Equivalent to choosing Text - Sort and selecting "First," "Last," "All," or "Other" in the desired sort level "Word" box.

Word Pro: SortWord property

{button ,AL(^H_SORTKEY_CLASS',0)} [See list of classes](#)

(Read-write) Allows you to sort on a specific word within a field.

Data Type

Integer

Syntax

sortwordvalue = [objectreference].SortWord

[objectreference].SortWord = sortwordvalue

Legal values

Any integer from 1 to 32767.

Usage

Equivalent to choosing Text - Sort, selecting "Other" from one of the desired "Word" boxes, and selecting a number from the "Word on which to sort" box.

If "From end of field" is selected, Word Pro will sort on the word number that you specified, counting backwards from the last word of the sort field.

Word Pro: SpaceAbove property

{button ,AL(^H_FOOTNOTESEPOPT_CLASS;H_FOOTNOTECONTSEP_CLASS;H_FOOTNOTESEPARATOR_CLASS',0)} [See list of classes](#)

(Read-write) Determines the space above a footnote separator.

Data Type

Long

Syntax

spaceabovevalue = [objectreference].SpaceAbove

[objectreference].SpaceAbove = spaceabovevalue

Legal values

Data type is Long but the unit of measurement used for this property is Twips. There are 1440 Twips per inch.

Usage

Equivalent to the "Space above" option on the Separators panel of the Footnote and Endnote Options dialog box.

Word Pro: SpaceBelow property

{button ,AL(^H_FOOTNOTESEPOPT_CLASS;H_FOOTNOTECONTSEP_CLASS;H_FOOTNOTESEPARATOR_CLASS',0)} [See list of classes](#)

(Read-write) Determines the space below a footnote separator.

Data Type

Long

Syntax

spacebelowvalue = [objectreference].SpaceBelow

[objectreference].SpaceBelow = spacebelowvalue

Legal values

Data type is Long but the unit of measurement used for this property is Twips. There are 1440 Twips per inch.

Usage

Equivalent to the "Space below" option on the Separators panel of the Footnote and Endnote Options dialog box.

Word Pro: SpacesBetweenSentences property

{button ,AL('H_GRAMMAR_CLASS',0)} [See list of classes](#)

{button ,AL('H_SPACESBETWEENSENTENCES_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Indicates the number of spaces between sentences that will be allowed in a sentence in Grammar Check.

Data Type

[Integer](#)

Syntax

spacesbetweensentencesvalue = [objectreference].SpacesBetweenSentences

[objectreference].SpacesBetweenSentences = spacesbetweensentencesvalue

Legal values

The legal values for this property are 1 or 2.

Usage

Use this property to set the number of spaces between sentences, either 1 or 2. Equivalent to choosing Edit - Check Grammar, clicking Options, and typing 1 or 2 in the "Number of spaces between sentences" field on the Grammatical Style panel.

Word Pro: SpacingStyleName property

{button ,AL('H_PARAGRAPHSTYLE_CLASS',0)} [See list of classes](#)

(Read-write)

Data Type

String

Syntax

spacingstylevalue = [objectreference].SpacingStyleName

[objectreference].SpacingStyleName = spacingstylevalue

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Word Pro: SpacingUnitName property

{button ,AL(^H_USERINTERFACEPREFS_CLASS,0)} [See list of classes](#)

(Read-only) The abbreviation for the name of the units in the SpacingUnits property.

Data Type

String

Syntax

spacingunitnamevalue = [objectreference].SpacingUnitName

Legal values

The legal values for this property are listed below under Usage.

Usage

This property corresponds to the [SpacingUnits property](#).

<u>SpacingUnits property value</u>	<u>SpacingUnitName property value</u>
\$LtsScaleModeCentimeter (1056964840)	"Centimeters (cm)"
\$LtsScaleModeInch (1056964838)	"Inches (in)"
\$LtsScaleModePoint (1056964837)	"Point (pts)"
\$LtsScalePica (1728)	"Picas (pi)"

Word Pro: SpacingUnits property

{button ,AL('H_USERINTERFACEPREFS_CLASS',0)} [See list of classes](#)

(Read-write) The scaling unit used for displaying spacing measurements in Word Pro.

Data Type

Variant (Enumerated)

Syntax

spacingunitsvalue = [objectreference].SpacingUnits

[objectreference].SpacingUnits = spacingunitsvalue

Legal values

<u>Value</u>	<u>Effect</u>
\$LtsScaleModeCentimeter (1056964840)	Default custom spacing units are centimeters.
\$LtsScaleModeInch (1056964838)	Default custom spacing units are inches.
\$LtsScaleModePoint (1056964837)	Default custom spacing units are points.
\$LwpScaleModePica (1728)	Default custom spacing units are picas.

Usage

Equivalent to the "Spacing units" box in the Spacing Custom dialog boxes for line spacing and paragraph spacing.

Word Pro: Spacing property

{button ,AL(^H_CLICKHERE_CLASS;H_TEXTMARKER_CLASS;H_PARAGRAPHSTYLE_CLASS;H_TEXT_CLASS',
0)} [See list of classes](#)

(Read-only)

Data Type

[Spacing](#)

Syntax

spacingvalue = [objectreference].Spacing

Legal values

Always contains an instance of the Spacing class.

Usage

Word Pro: Span property

```
{button ,AL('H_LAYOUT_CLASS;H_CELLGROUPLAYOUT_CLASS;H_CELLAYOUT_CLASS;H_CONNECTEDLAY  
OUT_CLASS;H_COLUMNGROUPLAYOUT_CLASS;H_FOOTERLAYOUT_CLASS;H_FRAMELAYOUT_CLASS;H  
_FRAMEGROUPLAYOUT_CLASS;H_DROPDOWNLAYOUT_CLASS;H_GROUPLAYOUT_CLASS;H_HEADERLAY  
OUT_CLASS;H_NOTELAYOUT_CLASS;H_PAGELAYOUT_CLASS;H_ROWGROUPLAYOUT_CLASS;H_ROWLA  
YOUT_CLASS;H_RUBYLAYOUT_CLASS;H_SUPERTABLEGROUPLAYOUT_CLASS;H_SUPERTABLELAYOUT  
_CLASS;H_TABLELAYOUT_CLASS;H_ENDNOTELAYOUT_CLASS;H_FOOTNOTELAYOUT_CLASS;H_TABLE  
HEADINGLAYOUT_CLASS;H_TOCSUPERTABLELAYOUT_CLASS',0)} See list of classes
```

(Read-write) Indicates whether or not a row layout object will span across multiple pages.

Data Type

Integer

Syntax

spanvalue = [objectreference].Span

[objectreference].Span = spanvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Equivalent to the "Row spans pages" option on the Table Cell Size & Margins panel of the InfoBox for cell layout objects. This property is used only with cell layout objects.

Word Pro: SpellCheckInitialCaps property

{button ,AL(^H_USERINTERFACEPREFS_CLASS,0)} [See list of classes](#)

(Read-write) Determines whether Spell Check will check words that begin with a capital letter.

Data Type

Integer

Syntax

spellcheckinitialcapsvalue = [objectreference].SpellCheckInitialCaps

[objectreference].SpellCheckInitialCaps = spellcheckinitialcapsvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values. Default is True (-1).

Usage

Equivalent to the "Check words with initial caps" option in the Spell Check Options dialog box. If this property is set to False (0), Spell Check skips words with initial caps. If set to True (-1), Spell Check checks words that have initial caps.

Word Pro: SpellCheckRepeatedWords property

{button ,AL('H_USERINTERFACEPREFS_CLASS',0)} [See list of classes](#)

(Read-write) Determines whether Spell Check will check for repeated words.

Data Type

Integer

Syntax

spellcheckrepeatedwordsvvalue = [objectreference].SpellCheckRepeatedWords

[objectreference].SpellCheckRepeatedWords = spellcheckrepeatedwordsvvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values. Default is True (-1).

Usage

Equivalent to the "Check for repeated words" option in the Spell Check Options dialog box. If this property is set to False (0), Spell Check skips repeated words if they are spelled correctly. If set to True (-1), Spell Check detects and flags repeated words.

Word Pro: SpellCheckUserDictAlternatives property

{button ,AL('H_USERINTERFACEPREFS_CLASS',0)} [See list of classes](#)

(Read-write) Determines whether Spell Check will search the user dictionary for spelling alternatives.

Data Type

Integer

Syntax

spellcheckuserdictalternativesvalue = [objectreference].SpellCheckUserDictAlternatives

[objectreference].SpellCheckUserDictAlternatives = spellcheckuserdictalternativesvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values. Default is False (0).

Usage

Equivalent to the "Include user dictionary alternatives" option in the Spell Check Options dialog box. If this property is set to True (-1), Spell Check searches the user dictionary for spelling alternatives. If set to False (0), Spell Check searches only the current language dictionary for spelling alternatives.

Word Pro: SpellCheckWordsWithNums property

{button ,AL(^H_USERINTERFACEPREFS_CLASS,0)} [See list of classes](#)

(Read-write) Determines whether Spell Check checks words with numbers in them.

Data Type

Integer

Syntax

spellcheckwordswithnumsvvalue = [objectreference].SpellCheckWordsWithNums

[objectreference].SpellCheckWordsWithNums = spellcheckwordswithnumsvvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values. Default is True (-1).

Usage

Equivalent to the "Check words with numbers" box in the Spell Check Options dialog box. If this property is set to True (-1), Spell Check checks words that contain numbers (for example, "Lotus123"). If set to False (0), Spell Check skips words that contain numbers.

Word Pro: SpellColor property

{button ,AL(^H_APPVIEWPREFS_CLASS',0)} [See list of classes](#)

(Read-only) Stores the color that highlights spelling and grammar errors.

Data Type

[Color](#)

Syntax

spellcolorvalue = [objectreference].SpellColor

Legal values

Always contains an instance of the Color class.

Usage

Equivalent to the "Color for unrecognized words" option in the Spell Check Options dialog box.

Word Pro: SpellFocusedColor property

{button ,AL(^H_APPVIEWPREFS_CLASS',0)} [See list of classes](#)

(Read-only) The color used to highlight spelling and grammar errors.

Data Type

[Color](#)

Syntax

spellfocusedcolorvalue = [objectreference].SpellFocusesColor

Legal values

Always contains an instance of the Color class.

Usage

This color is always the inverse of the color that is stored in the SpellColor property. You cannot change the red, green, or blue properties of the SpellFocusedColor object.

Word Pro: SplitPercentage property

{button ,AL(^H_USERINTERFACEPREFS_CLASS',0)} [See list of classes](#)

(Read-write) Sets or returns the percentage of the MDI window that will be assigned to a document that is opened or created in a split window.

Data Type

Integer

Syntax

splitpercentagevalue = [objectreference].SplitPercentage

[objectreference].SplitPercentage = splitpercentagevalue

Legal values

Value can range from 1 to 99. Default is 50.

Usage

Use this property to determine the percentage of the MDI window that will be allocated to a new split window. See the `HorizontalSplitWindow` and `VerticalSplitWindow` properties for more information on splitting the MDI window.

Some actions cause the `SplitPercentage` property to be reset to its default of 50. For example, calling the `SplitWindow` method resets the `SplitPercentage` to 50. Choosing `View - Split Left-Right`, or `View Split Top-Bottom` also resets this property to 50.

Word Pro: StartColumns property

{button ,AL(^H_MARKER_CLASS;H_POWERFIELD_CLASS;H_CLICKHERE_CLASS;H_TEXTMARKER_CLASS;H_RUBYMARKER_CLASS;H_TABLEMARKER_CLASS',0)} [See list of classes](#)

(Read-only)

Data Type

Integer

Syntax

startcolumnsvalue = [objectreference].StartColumns

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Word Pro: StartingColOfSelection property

{button ,AL(^H_BASETABLE_CLASS;H_FOOTNOTETABLE_CLASS;H_PARALLELCOLUMNS_CLASS;H_GLOSSARY_CLASS;H_TABLEHEADING_CLASS;H_TABLE_CLASS',0)} [See list of classes](#)

(Read-only) Returns the ID number of the first column included in a selection of table cells.

Data Type

Integer

Syntax

startingcolofselectionvalue = [objectreference].StartingColOfSelection

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

The column ID is a zero based value, which means that the first column in a table has an ID value of zero.

Word Pro: StartingColStringOfSelection property

{button ,AL(^H_BASETABLE_CLASS;H_FOOTNOTETABLE_CLASS;H_PARALLELCOLUMNS_CLASS;H_GLOSSARY_CLASS;H_TABLEHEADING_CLASS;H_TABLE_CLASS',0)} [See list of classes](#)

(Read-only) Returns a string ID value which represents the first column contained in a selection of cells.

Data Type

String

Syntax

startingcolstringofselectionvalue = [objectreference].StartingColStringOfSelection

Legal values

The value of this property cannot be set by a script.

Usage

String column ID values are calculated alphabetically from left to right. The first 26 columns correspond to the letters of the alphabet, starting with "A" and continuing through "Z." Column 27 has a value of "AA," and column 52 has a value of "AZ." Column 53 has a value of "BA," column 79 has a value of "CA," and so on.

Word Pro: StartingNumber property

{button ,AL(^H_FOOTNOTENUMOPT_CLASS;H_ENDNOTEDIVISIONGROUPNUM_CLASS;H_ENDNOTEDIVISIONNUM_CLASS;H_ENDNOTEDOCNUM_CLASS;H_FOOTNOTENUMBERING_CLASS',0)} [See list of classes](#)

{button ,AL(^H_STARTINGNUMBER_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) The first number used to reference a footnote or endnote in a document.

Data Type

[Integer](#)

Syntax

startingnumbervalue = [objectreference].StartingNumber

[objectreference].StartingNumber = startingnumbervalue

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Equivalent to choosing Create - Footnote/Endnote, clicking Options, and selecting a number in the "Starting at" box on the Numbering panel.

Word Pro: StartingRowOfSelection property

{button ,AL(^H_BASETABLE_CLASS;H_FOOTNOTETABLE_CLASS;H_PARALLELCOLUMNS_CLASS;H_GLOSSARY_CLASS;H_TABLEHEADING_CLASS;H_TABLE_CLASS',0)} [See list of classes](#)

(Read-only) Returns the number of the first row included in a selection of table cells.

Data Type

Integer

Syntax

startingrowofselectionvalue = [objectreference].StartingRowOfSelection

Legal values

This property is read-only and cannot be set by a script.

Usage

The row ID is a zero based value, which means that the first row in a table has an ID value of zero.

Word Pro: StartRow property

{button ,AL(^H_MARKER_CLASS;H_POWERFIELD_CLASS;H_CLICKHERE_CLASS;H_TEXTMARKER_CLASS;H_RUBYMARKER_CLASS;H_TABLEMARKER_CLASS',0)} [See list of classes](#)

(Read-only)

Data Type

Integer

Syntax

startrowvalue = [objectreference].StartRow

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Word Pro: Start property

{button ,AL('H_USEWHEN_CLASS',0)} [See list of classes](#)

(Read-write)

Data Type

Variant (Enumerated)

StartType

Syntax

startvalue = [objectreference].Start

[objectreference].Start = startvalue

Legal values

\$LwpStartTypeNextevenpage (1827)

\$LwpStartTypeNextoddpager (1826)

\$LwpStartTypeNextpage (1824)

\$LwpStartTypeThispage (1825)

Usage

Word Pro: StateID property

```
{button ,AL(^H_DIVISION_CLASS;H_MARKER_CLASS;H_POWERFIELD_CLASS;H_CLICKHERE_CLASS;H_TEXT  
MARKER_CLASS;H_RUBYMARKER_CLASS;H_TABLEMARKER_CLASS;H_TEXTDOCUMENT_CLASS',0)}
```

[See list of classes](#)

(Read-only)

Data Type

Long

Syntax

stateidvalue = [objectreference].StateID

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Word Pro: StatusBarButtons property

{button ,AL(^H_STATUSBAR_CLASS',0)} [See list of classes](#)

(Read-only) This property provides a list of all buttons in the status bar, by name.

Data Type

[StatusBarButtonCollection](#)

Syntax

statusbarbuttonvalue = [objectreference].StatusBarButtons

Legal values

Always contains an instance of the StatusBarButtonCollection class.

Usage

If you add a new status button to the status bar in LotusScript, this property returns an assigned number, such as CUSTOM1.

Word Pro: StatusBarVisible property

{button ,AL('H_APPLICATIONWINDOW_CLASS',0)} [See list of classes](#)

(Read-only) A flag that tells you whether or not the StatusBar is turned on.

Data Type

Integer

Syntax

statusbarvisiblevalue = [objectreference].StatusBarVisible

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values. Default value is True (-1).

Usage

If you are in clean screen mode, the value is False (0).

Word Pro: StatusBar property

{button ,AL('H_APPLICATIONWINDOW_CLASS',0)} [See list of classes](#)

(Read-only) An instance of the StatusBar class.

Data Type

[StatusBar](#)

Syntax

statusbarvalue = [objectreference].StatusBar

Legal values

Always contains an instance of the StatusBar class.

Usage

Use this property to access the StatusBar object when a document is not open.

Word Pro: StatusSpellReplaceAll property

{button ,AL(^H_USERINTERFACEPREFS_CLASS',0)} [See list of classes](#)

(Read-write) Sets or returns whether the Spell Check button on the status bar replaces the current misspelling or all similar misspellings.

Data Type

[Integer \(Bool\)](#)

Syntax

statusspellreplaceallvalue = [objectreference].StatusSpellReplaceAll

[objectreference].StatusSpellReplaceAll = statusspellreplaceallvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values. Default is False (0).

Usage

If the value is False (0), the Spell Check button on the status bar replaces only the currently highlighted word. If the value is True (-1), the Spell Check button on the status bar replaces all instances of this error with the selected word.

Word Pro: StrikeThrough property

{button ,AL('H_FONT_CLASS',0)} [See list of classes](#)

(Read-write) Determines whether or not a font uses the strikethrough attribute.

Data Type

Integer

Syntax

strikethroughvalue = [objectreference].StrikeThrough

[objectreference].StrikeThrough = strikethroughvalue

Legal values**Usage**

For the font object in the Text class, this property is equivalent to the "Strikethrough" attribute on the Text Font properties panel of the Text InfoBox.

Word Pro: StyleExceptions property

{button ,AL(^H_LAYOUT_CLASS;H_CELLGROUPLAYOUT_CLASS;H_CELLLAYOUT_CLASS;H_CONNECTEDLAYOUT_CLASS;H_COLUMNGROUPLAYOUT_CLASS;H_FOOTERLAYOUT_CLASS;H_FRAMELAYOUT_CLASS;H_FRAMEGROUPLAYOUT_CLASS;H_DROPDOWNLAYOUT_CLASS;H_GROUPLAYOUT_CLASS;H_HEADERLAYOUT_CLASS;H_NOTELAYOUT_CLASS;H_PAGELAYOUT_CLASS;H_ROWGROUPLAYOUT_CLASS;H_ROWLAYOOUT_CLASS;H_RUBYLAYOUT_CLASS;H_SUPERTABLEGROUPLAYOUT_CLASS;H_SUPERTABLELAYOUT_CLASS;H_TABLELAYOUT_CLASS;H_ENDNOTELAYOUT_CLASS;H_FOOTNOTELAYOUT_CLASS;H_TABLEHEADINGLAYOUT_CLASS;H_TOCSUPERTABLELAYOUT_CLASS;H_CLICKHERE_CLASS;H_TEXTMARKER_CLASS;H_TEXT_CLASS',0)} [See list of classes](#)

(Read-only) Provides a Long numeric value that indicates which style attributes of an object have been overridden.

Data Type

Data type is Long, which allows the value of this parameter to be one of the constants listed below or its hexadecimal equivalent (in parentheses). You can combine these values when you want Word Pro to combine the features listed below. Use the OR operator to combine values.

Syntax

styleexceptionsvalue = [objectreference].StyleExceptions

Legal values

[Layout]

<u>Value</u>	<u>Effect</u>
LwpLayStyOverSize (&H1)	The size settings of the layout object are overridden.
LwpLayStyOverPlacement (&H2)	The placement settings of the layout object are overridden.
LwpLayStyOverSizeAndPlacement (&H3)	The size and placement settings of the layout object are overridden.
LwpLayStyOverMargins (&H4)	The margin settings of the layout object are overridden.
LwpLayStyOverBorders (&H8)	The border settings of the layout object are overridden.
LwpLayStyOverBackground (&H10)	The background settings of the layout object are overridden.
LwpLayStyOverJoins (&H20)	The join settings of the layout object are overridden.
LwpLayStyOverShadow (&H40)	The shadow settings of the layout object are overridden.
LwpLayStyOverTabs (&H80)	The tab settings of the layout object are overridden.
LwpLayStyOverScript (&H100)	The script settings of the layout object are overridden.
LwpLayStyOverNumerics (&H200)	The numeric settings of the layout object are overridden.
LwpLayStyOverColumns (&H400)	The column settings of the layout object are overridden.
LwpLayStyOverScaling (&H800)	The scaling settings of the layout object are overridden.
LwpLayStyOverRotation (&H1000)	The rotation settings of the layout object are overridden.
LwpLayStyOverLeaders (&H2000)	The leader settings of the layout object are overridden.
LwpLayStyOverOrientation (&H4000)	The orientation settings of the layout object are overridden.

LwpLayStyOverMisc (&H8000)

Miscellaneous settings of the layout object are overridden. Miscellaneous settings are those settings that don't appear in any of the other categories.

LwpLayStyOverChildren (&H10000)

The child layout settings of the layout object are overridden.

LwpLayStyOverContents (&H20000)

The content settings of the layout object are overridden.

[Text]

None = &H0

Face = &H1

Size = &H2

Attributes = &H4

bold italics strikethru super sub smallcaps underline single and double underline upper lower

Font = &H8

everything else hidden protected color

Alignment = &H10

Indent = &H20

Spacing = &H40

Paragraph and line

Borders = &H80

Breaks = &H100

Bullet = &H200

Numbering = &H400

Tabs = &H800

Kinsoku = &H1000

Charborder = &H2000

Amikake = &H4000

Usage

[Layout]

Layout objects, such as frames, table cells, and pages are based on styles. For example, when you create a frame in Word Pro, it is usually based on a style called Default Frame. When you create a table in Word Pro, the table is usually based on a style called Default Table, and the table cells are usually based on a style called Default Cell.

The StyleExceptions property of a layout object represents all of the layout properties which were modified from the original style properties. For example, if the Default Frame style specifies a margin value of 1/10 inch, and you've modified the margins of frame A to 1/2 inch, the StyleExceptions property of frame A will include the value &H4 in its bitmask. This indicates that the margin settings of frame A were modified and do not coincide with the margin settings that were provided by the Default Frame style.

Word Pro: StyleName property

{button ,AL(^H_CHARACTERSTYLE_CLASS',0)} [See list of classes](#)

(Read-only)

Data Type

String

Syntax

stylenamevalue = [objectreference].StyleName

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Word Pro: StylePaths property

{button ,AL('H_USERINTERFACEPREFS_CLASS',0)} [See list of classes](#)

(Read-only) Stores multiple paths (drive and directories) for SmartMaster files.

Data Type

[StringCollection](#)

Syntax

stylepathsvalue = [objectreference].StylePaths

Legal values

Always contains an instance of the StringCollection class.

Usage

Equivalent to the "SmartMaster" value on the Locations panel of the Word Pro Preferences dialog box. "SmartMaster" can contain multiple paths. You can use this property to read these multiple paths, including the default or first path that is also stored in the StylePath property on UserInterfacePrefs.

Word Pro: StylePath property

{button ,AL(^H_USERINTERFACEPREFS_CLASS',0)} [See list of classes](#)

(Read-write) Sets or returns the default path (drive and directory) for Word Pro SmartMasters.

Data Type

String

Syntax

stylepathvalue = [objectreference].StylePath

[objectreference].StylePath = stylepathvalue

Legal values

A valid path including drive and directory.

Usage

Equivalent to the "SmartMaster" field on the Locations panel of the Word Pro Preferences dialog box. In Word Pro, the "SmartMaster" field can contain multiple paths. This property contains the first path listed in the "SmartMaster" field.

Word Pro: StyleSheetFullPath property

{button ,AL('H_TEXTDOCUMENT_CLASS',0)} [See list of classes](#)

(Read-write)

Data Type

String

Syntax

stylesheetfullpathvalue = [objectreference].StyleSheetFullPath

[objectreference].StyleSheetFullPath = stylesheetfullpathvalue

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Word Pro: StyleSheetName property

{button ,AL('H_TEXTDOCUMENT_CLASS;H_USERINTERFACEPREFS_CLASS',0)} [See list of classes](#)

[UserInterfacePrefs]

(Read-write) Sets or returns the name of the default SmartMaster to be used for new documents.

[TextDocument]

(Read-only) Indicates the name of the SmartMaster template that was used to create the document.

Data Type

String

Syntax

stylesheetnamevalue = [objectreference].StyleSheetName

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

[UserInterfacePrefs]

Equivalent to the "Plain Document SmartMaster" value on the Default files panel of the Word Pro Preferences dialog box.

Word Pro: StyleSheetPath property

{button ,AL('H_TEXTDOCUMENT_CLASS',0)} [See list of classes](#)

(Read-only)

Data Type

String

Syntax

stylesheetpathvalue = [objectreference].StyleSheetPath

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Word Pro: Style property

{button ,AL(^H_LAYOUT_CLASS;H_CELLGROUPLAYOUT_CLASS;H_CELLLAYOUT_CLASS;H_CONNECTEDLAYOUT_CLASS;H_COLUMNGROUPLAYOUT_CLASS;H_FOOTERLAYOUT_CLASS;H_FRAMELAYOUT_CLASS;H_FRAMEGROUPLAYOUT_CLASS;H_DROPDOWNLAYOUT_CLASS;H_GROUPLAYOUT_CLASS;H_HEADERLAYOUT_CLASS;H_NOTELAYOUT_CLASS;H_PAGELAYOUT_CLASS;H_ROWGROUPLAYOUT_CLASS;H_ROWLAYOOUT_CLASS;H_RUBYLAYOUT_CLASS;H_SUPERTABLEGROUPLAYOUT_CLASS;H_SUPERTABLELAYOUT_CLASS;H_TABLELAYOUT_CLASS;H_ENDNOTELAYOUT_CLASS;H_FOOTNOTELAYOUT_CLASS;H_TABLEHEADINGLAYOUT_CLASS;H_TOCSUPERTABLELAYOUT_CLASS;H_BORDER_CLASS',0)} [See list of classes](#)

(Read-only) Indicates the specific layout object from which the current layout object inherited its style settings.

Data Type

[Layout](#)

Syntax

stylevalue = [objectreference].Style

Legal values

Always contains an instance of the Layout class.

Usage

Use this property to access the default property values that were assigned to a layout object when it was created. For example, you can create a frame based on the Default Frame style and then change the number of columns within that frame. To access the current number of columns for the frame, you can use the following statement:

```
[framelayoutobject].numcols
```

To check the number of columns a frame was assigned when it was initially created, you can use the following statement:

```
[framelayoutobject].style.numcols
```

Word Pro: Subscript property

{button ,AL('H_FONT_CLASS',0)} [See list of classes](#)

{button ,AL('H_SUBSCRIPT_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write)

Data Type

Integer

Syntax

subscriptvalue = [objectreference].Subscript

[objectreference].Subscript = subscriptvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Word Pro: Suffix property

{button ,AL('H_NUMERICFORMATSUBSET_CLASS',0)} [See list of classes](#)

{button ,AL('H_SUFFIX_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Allows you to include suffix text with a specific number format.

Data Type

String

Syntax

suffixvalue = [objectreference].Suffix

[objectreference].Suffix = suffixvalue

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Equivalent to the "Text after" option in the Edit Format dialog box. You can access this dialog box by clicking the Format Options button on the Number Format panel of the InfoBox for cell layout objects.

Word Pro: Suggestions property

{button ,AL('H_EDITOR_CLASS',0)} [See list of classes](#)

{button ,AL('H_SUGGESTIONS_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) The suggestions that appear for an editor when a document is opened.

Data Type

Data type is [Variant](#) which allows the value of this property to be one of the constants listed below or its hexadecimal equivalent (in parentheses). You can combine these constants when you want Word Pro to combine the features listed below. Use the OR operator to combine constants.

Syntax

suggestionsvalue = [objectreference].Suggestions

[objectreference].Suggestions = suggestionsvalue

Legal values

<u>Value</u>	<u>Effect</u>
LwpEditSuggEditingInNewVersion (&H4)	Suggests that an editor create a new version in which to work; only an option when edits are in the current or new version.
LwpEditSuggNoSuggestions (&H0)	No suggestions display.
LwpEditSuggOnlyMarkupEditsAllowed (&H1)	Suggests that edits appear as markups.
LwpEditSuggRevAndCommentIconbar (&H2)	Suggests that the editor use the Review & Comment Tools icon bar to insert comments.

Usage

Use one of the values above to determine which suggestion displays when the editor opens the document. For more information on combining these values, called bitmasks, see the topic [Overview: Word Pro LotusScript Enumerated Values](#).

Word Pro: Superscript property

{button ,AL('H_FONT_CLASS',0)} [See list of classes](#)

{button ,AL('H_SUPERSCRIPT_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write)

Data Type

[Integer](#)

Syntax

superscriptvalue = [objectreference].Superscript

[objectreference].Superscript = superscriptvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Word Pro: SuperTableContainer property

{button ,AL(^H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

(Read-only) An instance of the SuperTableContainer class. This is a current context property that only contains an object when the focus of Word Pro includes a table. If there is no table in the focus, this property is empty.

Data Type

[SuperTableContainer](#)

Syntax

supertablecontainervalue = [objectreference].SuperTableContainer

Legal values

An instance of the SuperTableContainer class.

Usage

When the focus includes a table, this property contains the SuperTableContainer object which groups together the objects that comprise the SuperTable object in the focus. You can use this property to access the Layout or other objects related to that super table.

Word Pro: SuperTableLayouts property

{button ,AL('H_FOUNDRY_CLASS',0)} [See list of classes](#)

(Read-only) An object created from the SuperTableLayoutCollection class. This object provides access to SuperTableLayout objects.

Data Type

[SuperTableLayoutCollection](#)

Syntax

supertablelayoutsvalue = [objectreference].SuperTableLayouts

Legal values

Always contains an instance of the SuperTableLayoutCollection class.

Usage

When accessed through the Foundry property on a Division object, the collection object in this property provides access to all the SuperTableLayout objects contained in that Division object.

When accessed through the AppFoundry property on the WPAApplication object, this collection object provides access to all the SuperTableLayout objects contained in the Word Pro Clipboard.

When accessed through the TempFoundry property on the WPAApplication object, this collection object provides access to all the SuperTableLayout objects placed in TempFoundry by WordPro or a script.

When accessed through the Foundry property on the WPAApplication object, this collection object provides access to all the SuperTableLayout objects contained in the currently active Division object.

This property is not used in the Foundry property on the TextDocument object.

For more information about collection classes, see [Overview: Word Pro LotusScript Collection Classes](#).

Word Pro: SuperTables property

{button ,AL('H_FOUNDRY_CLASS',0)} [See list of classes](#)

(Read-only) An object created from the SuperTableCollection class. This object provides access to SuperTable objects.

Data Type

[SuperTableCollection](#)

Syntax

supertablesvalue = [objectreference].SuperTables

Legal values

Always contains an instance of the SuperTableCollection class.

Usage

When accessed through the Foundry property on a Division object, the collection object in this property provides access to all the SuperTable objects contained in that Division object.

When accessed through the AppFoundry property on the WPAApplication object, this collection object provides access to all the SuperTable objects contained in the Word Pro Clipboard.

When accessed through the TempFoundry property on the WPAApplication object, this collection object provides access to all the SuperTable objects placed in TempFoundry by WordPro or a script.

When accessed through the Foundry property on the WPAApplication object, this collection object provides access to all the SuperTable objects contained in the currently active Division object.

This property is not used in the Foundry property on the TextDocument object.

For more information about collection classes, see [Overview: Word Pro LotusScript Collection Classes](#).

Word Pro: SuppressHeaders property

{button ,AL('H_DIVISIONINFO_CLASS',0)} [See list of classes](#)

{button ,AL('H_SUPPRESSHEADERS_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Specifies whether headers will be suppressed on filler pages.

Data Type

[Integer](#)

Syntax

suppressheadersvalue = [objectreference].SuppressHeaders

[objectreference].SuppressHeaders = suppressheadersvalue

Legal values

<u>Value</u>	<u>Effect</u>
0	Word Pro does not suppress headers on filler pages.
1	Word Pro suppresses headers on filler pages.

Usage

Equivalent to "Suppress headers/footers on filler pages" on the Options panel of the Document Properties dialog box.

Word Pro: TabExits property

{button ,AL('H_CLICKHERE_CLASS',0)} [See list of classes](#)

{button ,AL('H_TABEXITS_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Indicates whether pressing the Tab key while in a particular ClickHere object will exit the ClickHere or insert a normal tab mark.

Data Type

[Integer](#)

Syntax

tabexitsvalue = [objectreference].TabExits

[objectreference].TabExits = tabexitsvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Word Pro: TableContainer property

{button ,AL(^H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

(Read-only) An instance of the ParallelColsContainer class or the TableOnlyCont class. This is a current context property that only contains an object when the focus of Word Pro includes a page with parallel columns or a table. If neither parallel columns nor a table are in the focus, this property is empty.

Data Type

TableContainer

Syntax

tablecontainervalue = [objectreference].TableContainer

Legal values

An instance of the ParallelColsContainer class or the TableOnlyCont class.

Usage

When the focus includes a page with parallel columns, this property contains the ParallelColsContainer object, which groups together the objects that comprise the parallel columns in the focus. You can use this property to access the Layout object related to those parallel columns.

When the focus includes a table, this property contains the TableOnlyCont object that groups together the objects that comprise the table in the focus. You can use this property to access the TableLayout or other objects related to that table.

Word Pro: TableExports property

{button ,AL(^H_FILTER_CLASS',0)} [See list of classes](#)

(Read-only)

Data Type

[StringCollection](#)

Syntax

tableexportsvalue = [objectreference].TableExports

Legal values

Always contains an instance of the StringCollection class.

Usage

Word Pro: TableFill property

{button ,AL(^H_BASSETABLE_CLASS;H_FOOTNOTETABLE_CLASS;H_PARALLELCOLUMNS_CLASS;H_GLOSSARY_CLASS;H_TABLEHEADING_CLASS;H_TABLE_CLASS',0)} [See list of classes](#)

(Read-only) Contains an instance of the TableFill class, which allows you to modify the background and pattern information of table cells.

Data Type

[TableFill](#)

Syntax

tablefillvalue = [objectreference].TableFill

Legal values

Always contains an instance of the TableFill class.

Usage

Word Pro: TableHeadingLayouts property

{button ,AL('H_FOUNDRY_CLASS',0)} [See list of classes](#)

(Read-only) An object created from the TableHeadingLayoutCollection class. This object provides access to TableHeadingLayout objects.

Data Type

[TableHeadingLayoutCollection](#)

Syntax

tableheadinglayoutsvalue = [objectreference].TableHeadingLayouts

Legal values

Always contains an instance of the TableHeadingLayoutCollection class.

Usage

When accessed through the Foundry property on a Division object, the collection object in this property provides access to all the TableHeadingLayout objects contained in that Division object.

When accessed through the AppFoundry property on the WPAApplication object, this collection object provides access to all the TableHeadingLayout objects contained in the Word Pro Clipboard.

When accessed through the TempFoundry property on the WPAApplication object, this collection object provides access to all the TableHeadingLayout objects placed in TempFoundry by WordPro or a script.

When accessed through the Foundry property on the WPAApplication object, this collection object provides access to all the TableHeadingLayout objects contained in the currently active Division object.

This property is not used in the Foundry property on the TextDocument object.

For more information about collection classes, see [Overview: Word Pro LotusScript Collection Classes](#).

Word Pro: TableHeadings property

{button ,AL('H_FOUNDRY_CLASS',0)} [See list of classes](#)

(Read-only) An object created from the TableHeadingCollection class. This object provides access to TableHeading objects.

Data Type

[TableHeadingCollection](#)

Syntax

tableheadingsvalue = [objectreference].TableHeadings

Legal values

Always contains an instance of the TableHeadingCollection class.

Usage

When accessed through the Foundry property on a Division object, the collection object in this property provides access to all the TableHeading objects contained in that Division object.

When accessed through the AppFoundry property on the WPAApplication object, this collection object provides access to all the TableHeading objects contained in the Word Pro Clipboard.

When accessed through the TempFoundry property on the WPAApplication object, this collection object provides access to all the TableHeading objects placed in TempFoundry by WordPro or a script.

When accessed through the Foundry property on the WPAApplication object, this collection object provides access to all the TableHeading objects contained in the currently active Division object.

This property is not used in the Foundry property on the TextDocument object.

For more information about collection classes, see [Overview: Word Pro LotusScript Collection Classes](#).

Word Pro: TableImports property

{button ,AL(^H_FILTER_CLASS',0)} [See list of classes](#)

(Read-only)

Data Type

[StringCollection](#)

Syntax

tableimportsvalue = [objectreference].TableImports

Legal values

Always contains an instance of the StringCollection class.

Usage

Word Pro: TableLayouts property

{button ,AL('H_FOUNDRY_CLASS',0)} [See list of classes](#)

(Read-only) An object created from the TableLayoutCollection class. This object provides access to TableLayout objects.

Data Type

[TableLayoutCollection](#)

Syntax

tablelayoutvalue = [objectreference].TableLayouts

Legal values

Always contains an instance of the TableLayoutCollection class.

Usage

When accessed through the Foundry property on a Division object, the collection object in this property provides access to all the TableLayout objects contained in that Division object.

When accessed through the AppFoundry property on the WPAApplication object, this collection object provides access to all the TableLayout objects contained in the Word Pro Clipboard.

When accessed through the TempFoundry property on the WPAApplication object, this collection object provides access to all the TableLayout objects placed in TempFoundry by WordPro or a script.

When accessed through the Foundry property on the WPAApplication object, this collection object provides access to all the TableLayout objects contained in the currently active Division object.

This property is not used in the Foundry property on the TextDocument object.

For more information about collection classes, see [Overview: Word Pro LotusScript Collection Classes](#).

Word Pro: TableLine property

{button ,AL(^H_BASETABLE_CLASS;H_FOOTNOTETABLE_CLASS;H_PARALLELCOLUMNS_CLASS;H_GLOSSARY_CLASS;H_TABLEHEADING_CLASS;H_TABLE_CLASS',0)} [See list of classes](#)

(Read-only) Contains an instance of the TableLine class, which allows you to modify border line styles around table objects.

Data Type

TableLine

Syntax

tablelinevalue = [objectreference].TableLine

Legal values

Always contains an instance of the TableLine class.

Usage

Word Pro: TableMarkers property

{button ,AL('H_FOUNDRY_CLASS',0)} [See list of classes](#)

(Read-only) An object created from the TableMarkerCollection class. This object provides access to TableMarker objects.

Data Type

[TableMarkerCollection](#)

Syntax

tablemarkersvalue = [objectreference].TableMarkers

Legal values

Always contains an instance of the TableMarkerCollection class.

Usage

When accessed through the Foundry property on a Division object, the collection object in this property provides access to all the TableMarker objects contained in that Division object.

When accessed through the AppFoundry property on the WPAApplication object, this collection object provides access to all the TableMarker objects contained in the Word Pro Clipboard.

When accessed through the TempFoundry property on the WPAApplication object, this collection object provides access to all the TableMarker objects placed in TempFoundry by WordPro or a script.

When accessed through the Foundry property on the WPAApplication object, this collection object provides access to all the TableMarker objects contained in the currently active Division object.

This property is not used in the Foundry property on the TextDocument object.

For more information about collection classes, see [Overview: Word Pro LotusScript Collection Classes](#).

Word Pro: TableOfContents property

{button ,AL('H_HTMLOPTIONS_CLASS',0)} [See list of classes](#)

{button ,AL('H_TABLEOFCONTENTS_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write)

This language element has not yet been defined. You can check for a newer Help file on the Lotus web site.

Data Type

Integer

Syntax

tableofcontentsvalue = [objectreference].TableOfContents

[objectreference].TableOfContents = tableofcontentsvalue

Legal values

The legal values for this property are -1 and 0 but you may use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Word Pro: TableOnlyContainer property

{button ,AL(^H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

(Read-only) An instance of the TableOnlyCont class. This is a current context property that only contains an object when the focus of Word Pro includes a table. If there is no table in the focus, this property is empty.

Data Type

[TableOnlyCont](#)

Syntax

tableonlycontainervalue = [objectreference].TableOnlyContainer

Legal values

An instance of the TableOnlyCont class.

Usage

When the focus includes a table, this property contains the TableOnlyCont object, which groups together the objects that comprise the table in the focus. You can use this property to access the TableLayout or other objects related to that table.

Word Pro: TableStyleName property

{button ,AL('H_PREFERENCES_CLASS',0)} [See list of classes](#)

(Read-write)

Data Type

String

Syntax

tablestylevalue = [objectreference].TableStyleName

[objectreference].TableStyleName = tablestylevalue

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Word Pro: TableStyles property

{button ,AL('H_FOUNDRY_CLASS',0)} [See list of classes](#)

(Read-only) An object created from the TableLayoutCollection class. This object provides access to TableLayout objects which are used as table styles.

Data Type

[TableLayoutCollection](#)

Syntax

tablestylesvalue = [objectreference].TableStyles

Legal values

Always contains an instance of the TableLayoutCollection class.

Usage

When accessed through the Foundry property on a Division object, the collection object in this property provides access to all the TableLayout objects which are used as table styles and contained in that Division object.

When accessed through the AppFoundry property on the WPAApplication object, this collection object provides access to all the TableLayout objects which are used as table styles and contained in the Word Pro Clipboard.

When accessed through the TempFoundry property on the WPAApplication object, this collection object provides access to all the TableLayout objects which are used as table styles and placed in TempFoundry by WordPro or a script.

When accessed through the Foundry property on the WPAApplication object, this collection object provides access to all the TableLayout objects which are used as table styles and contained in the currently active Division object.

This property is not used in the Foundry property on the TextDocument object.

For more information about collection classes, see [Overview: Word Pro LotusScript Collection Classes](#).

Word Pro: Tables property

{button ,AL('H_FOUNDRY_CLASS',0)} [See list of classes](#)

(Read-only) An object created from the TableOnlyCollection class. This object provides access to Table objects.

Data Type

[TableOnlyCollection](#)

Syntax

tablesvalue = [objectreference].Tables

Legal values

Always contains an instance of the TableOnlyCollection class.

Usage

When accessed through the Foundry property on a Division object, the collection object in this property provides access to all the Table objects contained in that Division object.

When accessed through the AppFoundry property on the WPAApplication object, this collection object provides access to all the Table objects contained in the Word Pro Clipboard.

When accessed through the TempFoundry property on the WPAApplication object, this collection object provides access to all the Table objects placed in TempFoundry by WordPro or a script.

When accessed through the Foundry property on the WPAApplication object, this collection object provides access to all the Table objects contained in the currently active Division object.

This property is not used in the Foundry property on the TextDocument object.

For more information about collection classes, see [Overview: Word Pro LotusScript Collection Classes](#).

Word Pro: Table property

{button ,AL(^H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

(Read-only) The Table object which is uppermost in the focus when this property is called.

Data Type

[Table](#)

Syntax

tablevalue = [objectreference].Table

Legal values

Always contains an instance of the Table class.

Usage

Word Pro: TabOrder property

{button ,AL('H_CLICKHERE_CLASS',0)} [See list of classes](#)

{button ,AL('H_TABORDER_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) The tab order for a ClickHere object.

Data Type

Long

Syntax

tabordervalue = [objectreference].TabOrder

[objectreference].TabOrder = tabordervalue

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Word Pro: TabRack property

{button ,AL('H_LAYOUT_CLASS;H_CELLGROUPLAYOUT_CLASS;H_CELLLAYOUT_CLASS;H_CONNECTEDLAY
OUT_CLASS;H_COLUMNGROUPLAYOUT_CLASS;H_FOOTERLAYOUT_CLASS;H_FRAMELAYOUT_CLASS;H
_FRAMEGROUPLAYOUT_CLASS;H_DROPDOWNLAYOUT_CLASS;H_GROUPLAYOUT_CLASS;H_HEADERLAY
OUT_CLASS;H_NOTELAYOUT_CLASS;H_PAGELAYOUT_CLASS;H_ROWGROUPLAYOUT_CLASS;H_ROWLA
YOUT_CLASS;H_RUBYLAYOUT_CLASS;H_SUPERTABLEGROUPLAYOUT_CLASS;H_SUPERTABLELAYOUT
_CLASS;H_TABLELAYOUT_CLASS;H_ENDNOTELAYOUT_CLASS;H_FOOTNOTELAYOUT_CLASS;H_TABLE
HEADINGLAYOUT_CLASS;H_TOCSUPERTABLELAYOUT_CLASS;H_CLICKHERE_CLASS;H_TEXTMARKER_
CLASS;H_PARAGRAPHSTYLE_CLASS;H_TEXT_CLASS',0)} [See list of classes](#)

{button ,AL('H_TABRACK_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-only) Returns the TabRack object for a layout object.

Data Type

[TabRack](#)

Syntax

tabrackvalue = [objectreference].TabRack

Legal values

Always contains an instance of the TabRack class.

Usage

Word Pro: TabRelativeTo property

{button ,AL('H_USERINTERFACEPREFS_CLASS',0)} [See list of classes](#)

(Read-write) The tab position on the ruler.

Data Type

Integer

Syntax

tabrelativetovalue = [objectreference].TabRelativeTo

[objectreference].TabRelativeTo = tabrelativetovalue

Legal values

<u>Value</u>	<u>Effect</u>
0	Tab position on ruler is relative to the left margin.
1	Tab position on ruler is relative to the right margin.
2	Tab position on ruler is centered between the left and right margins.

Usage

Equivalent to the "Tab position on ruler" field in the Set Tabs on Ruler dialog box. Although this box contains four choices, the legal values for this property only encompass three of those choices. The fourth choice, "Evenly spaced every," cannot be set as a value for this property.

Word Pro: TabSpacing property

{button ,AL(^H_DIVISION_CLASS;H_TEXTDOCUMENT_CLASS',0)} [See list of classes](#)

{button ,AL(^H_TABSPACING_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) The default tab spacing for a division or document.

Data Type

Long

Syntax

tabspacingvalue = [objectreference].TabSpacing

[objectreference].TabSpacing = tabspacingvalue

Legal values

Data type is Long but the unit of measurement used for this property is Twips. There are 1440 Twips per inch.

Usage

Setting this property is equivalent to setting a value in the Set Default Tabs dialog box.

Word Pro: TabSymbolChar property

{button ,AL(^H_CHARACTERSET_CLASS',0)} [See list of classes](#)

(Read-only)

Data Type

Integer

Syntax

tabsymbolcharvalue = [objectreference].TabSymbolChar

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Word Pro: TabType property

{button ,AL(^H_USERINTERFACEPREFS_CLASS',0)} [See list of classes](#)

(Read-write) Sets or returns the type of tab to be placed on the ruler.

Data Type

Data type is [Variant](#) which allows the value of this property to be one of the constants listed below or its numeric equivalent (in parentheses).

Syntax

tabtypevalue = [objectreference].TabType

[objectreference].TabType = tabtypevalue

Legal values

<u>Value</u>	<u>Effect</u>
\$LwpTabTypeCenter (1864)	Sets a center aligned tab on the ruler.
\$LwpTabTypeLeft (1863)	Sets a left aligned tab on the ruler.
\$LwpTabTypeNumeric (1866)	Sets a numeric tab on the ruler.
\$LwpTabTypeRight (1865)	Sets a right aligned tab on the ruler.

Usage

Equivalent to the "Tab type" field in the Set Tabs on Ruler dialog box.

Word Pro: TempFindAndReplace property

{button ,AL(^H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

(Read-only)

Data Type

[FindAndReplace](#)

Syntax

tempfindandreplacevalue = [objectreference].TempFindAndReplace

Legal values

Always contains an instance of the FindAndReplace class.

Usage

Word Pro: TempFoundry property

{button ,AL(^H_WPAPPLICATION_CLASS',0)} [See list of classes](#)

(Read-only) A Foundry object used by Word Pro to temporarily store objects during drag and drop operations. You can use the collections in this property's Foundry object to temporarily store objects while your script is running. You must clear the TempFoundry contents after each use.

Data Type

Foundry

Syntax

tempfoundryvalue = [objectreference].TempFoundry

Legal values

Always contains an instance of the Foundry class.

Usage

TempFoundry is a property in WordPro. TempFoundry contains another Foundry object. Word Pro uses TempFoundry to temporarily store objects which are part of a drag and drop, or other internal operation. You can use TempFoundry in much the same way.

Like its counterpart, AppFoundry, TempFoundry contains a Foundry object. You can use the collection objects on this Foundry object as a staging area for any Word Pro LotusScript objects you create and manipulate. For example, when you want to move an object or objects from one document to another, you can store those objects temporarily in the TempFoundry collection objects. The TempFoundry property is always available, regardless of which document is active, so you always have access to the contents of its collections. This makes it an ideal place for temporarily storing items which you want to use or move.

Note You must clear TempFoundry after each use. Any objects left in any of TempFoundry's collections can reappear during drag and drop and other operations, and result in unpredictable behavior.

Word Pro: TextAfter property

{button ,AL('H_FRAMECAPTIONOPTIONS_CLASS',0)} [See list of classes](#)

{button ,AL('H_TEXTAFTER_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) The text that appears after the sequence number in a frame caption.

Data Type

String.

Syntax

textaftervalue = [objectreference].TextAfter

[objectreference].TextAfter = textaftervalue

Legal values

String.

Usage

Equivalent to selecting the frame in the document, choosing Frame - New Caption, and typing text in the "Text after" box.

Word Pro: TextandTableExports property

{button ,AL(^H_FILTER_CLASS',0)} [See list of classes](#)

(Read-only)

Data Type

[StringCollection](#)

Syntax

textandtableexportsvalue = [objectreference].TextandTableExports

Legal values

Always contains an instance of the StringCollection class.

Usage

Word Pro: TextandTableImports property

{button ,AL(^H_FILTER_CLASS',0)} [See list of classes](#)

(Read-only)

Data Type

[StringCollection](#)

Syntax

textandtableimportsvalue = [objectreference].TextandTableImports

Legal values

Always contains an instance of the StringCollection class.

Usage

Word Pro: TextAttributes property

{button ,AL(^H_CHARACTERSTYLE_CLASS;H_EDITOR_CLASS;H_REVISIONDISPLAY_CLASS;H_PARAGRAPHS_TYLE_CLASS',0)} [See list of classes](#)

[Editor]

(Read-only) Comprised of other attributes that are not normally associated as part of the font, such as highlight, hidden, or protected. This property only applies to text characteristics of specific deleted text.

Data Type

[Attributes](#)

Syntax

textattributesvalue = [objectreference].TextAttributes

[objectreference].TextAttributes = textattributesvalue

Legal values

Always contains an instance of the Attributes class.

Usage

[Editor]

In a Word Pro document, the font describes the typeface, point size, color, and attributes (such as bold, italic, and so on) of any inserted text. However, the TextAttributes property describes other text characteristics, such as whether the text is highlighted, hidden, or protected, and applies only to deleted text.

For example, when an editor uses the markup option that hides deleted text, the TextAttributes property, as part of the Editor class, captures all the characteristics of the deleted text. That editor can then use the TextAttributes property to determine the text characteristics of specific deleted text.

Word Pro: TextBefore property

{button ,AL('H_FRAMECAPTIONOPTIONS_CLASS',0)} [See list of classes](#)

{button ,AL('H_TEXTBEFORE_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) The text that should appear before the sequence number in a frame caption.

Data Type

String.

Syntax

textbeforevalue = [objectreference].TextBefore

[objectreference].TextBefore = textbeforevalue

Legal values

String.

Usage

Equivalent to selecting the frame in a document, choosing Frame - New Caption, and typing text in the "Text before" box.

Word Pro: TextMarkersInSelection property

{button ,AL('H_CLICKHERE_CLASS;H_TEXTMARKER_CLASS;H_TEXT_CLASS',0)} [See list of classes](#)

{button ,AL('H_TEXTMARKERSINSELECTION_EXSCRIPT',1)} [See example](#)

(Read-only)

This language element has not yet been defined. Lotus provides updated Help files on its web site.

Data Type

StringCollection

Syntax

textmarkersinselectionvalue = [objectreference].TextMarkersInSelection

Legal values

Always contains an instance of the StringCollection class.

Usage

{button ,AL('H_TEXTMARKERS_PROPERTY_MEMDEF',0)} [See related topics](#)

Word Pro: TextMarkers property

{button ,AL('H_FOUNDRY_CLASS',0)} [See list of classes](#)

(Read-only) An object created from the TextMarkerCollection class. This object provides access to TextMarker objects.

Data Type

[TextMarkerCollection](#)

Syntax

textmarkersvalue = [objectreference].TextMarkers

Legal values

Always contains an instance of the TextMarkerCollection class.

Usage

When accessed through the Foundry property on a Division object, the collection object in this property provides access to all the TextMarker objects contained in that Division object.

When accessed through the AppFoundry property on the WPAApplication object, this collection object provides access to all the TextMarker objects contained in the Word Pro Clipboard.

When accessed through the TempFoundry property on the WPAApplication object, this collection object provides access to all the TextMarker objects placed in TempFoundry by WordPro or a script.

When accessed through the Foundry property on the WPAApplication object, this collection object provides access to all the TextMarker objects contained in the currently active Division object.

This property is not used in the Foundry property on the TextDocument object.

For more information about collection classes, see [Overview: Word Pro LotusScript Collection Classes](#).

{button ,AL('H_TEXTMARKERS_IN_SELECTION_PROPERTY_MEMDEF',0)} [See related topics](#)

Word Pro: TextOrientation property

```
{button ,AL(^H_BASECONTAINER_CLASS;H_CELLCONTAINER_CLASS;H_FRAMECONTAINER_CLASS;H_NOTECONTAINER_CLASS;H_RUBYCONTAINER_CLASS;H_DROPDOWNCONTAINER_CLASS;H_PAGECONTAINER_CLASS;H_SUBPAGECONTAINER_CLASS;H_SUPERPAGECONTAINER_CLASS;H_ROWCONTAINER_CLASS;H_SUPERTABLECONTAINER_CLASS;H_TABLECONTAINER_CLASS;H_PARALLELCOLSCONTAINER_CLASS;H_TABLEONLYCONT_CLASS',0)} See list of classes
```

(Read-only) Returns the orientation of text in relation to its container.

Data Type

Data type is Variant, which allows the value of this property to be one of the string constants below or its numeric equivalent (in parentheses).

Syntax

textorientationvalue = [objectreference].TextOrientation

Legal values

The legal values for this property are listed below:

<u>Value</u>	<u>Effect</u>
0	The TextOrient property of the Layout object is set to \$LwpTextOrientLefttorightToptobottom.
1	The TextOrient property of the Layout object is set to \$LwpTextOrientToptobottomRighttoleft.
2	The TextOrient property of the Layout object is set to \$LwpTextOrientRighttoleftBottomtotop.
3	The TextOrient property of the Layout object is set to \$LwpTextOrientBottomtotopLefttoright.
4	The TextOrient property of the Layout object is set to \$LwpTextOrientRighttoleftToptobottom.
5	The TextOrient property of the Layout object is set to \$LwpTextOrientToptobottomLefttoright.
6	The TextOrient property of the Layout object is set to \$LwpTextOrientLefttorightBottomtotop.
7	The TextOrient property of the Layout object is set to \$LwpTextOrientBottomtotopRighttoLeft.

Usage

The TextOrientation property takes into account the orientation of text in relation only to the parent container. For example, frame A is oriented from right to left and bottom to top. Frame A contains table cell B, which is also oriented from right to left and bottom to top. Text contained in table cell B appears on the screen from left to right and top to bottom. However, the TextOrientation property of cell B contains a value of 2. This indicates that the text is oriented from right to left and bottom to top in relation to its parent container, which is cell B.

Word Pro: TextOrient property

```
{button ,AL(^H_LAYOUT_CLASS;H_CELLGROUPLAYOUT_CLASS;H_CELLLAYOUT_CLASS;H_CONNECTEDLAY  
OUT_CLASS;H_COLUMNGROUPLAYOUT_CLASS;H_FOOTERLAYOUT_CLASS;H_FRAMELAYOUT_CLASS;H  
_FRAMEGROUPLAYOUT_CLASS;H_DROP_CAP_LAYOUT_CLASS;H_GROUPLAYOUT_CLASS;H_HEADERLAY  
OUT_CLASS;H_NOTELAYOUT_CLASS;H_PAGELAYOUT_CLASS;H_ROWGROUPLAYOUT_CLASS;H_ROWLA  
YOUT_CLASS;H_RUBY_LAYOUT_CLASS;H_SUPERTABLEGROUPLAYOUT_CLASS;H_SUPERTABLELAYOUT  
_CLASS;H_TABLELAYOUT_CLASS;H_ENDNOTE_LAYOUT_CLASS;H_FOOTNOTE_LAYOUT_CLASS;H_TABLE  
HEADING_LAYOUT_CLASS;H_TOCSUPERTABLELAYOUT_CLASS',0)} See list of classes
```

(Read-write) The orientation of text in a layout.

Data Type

Data type is Variant, which allows the value of this property to be one of the constants listed below or its numeric equivalent (in parentheses).

Syntax

textorientvalue = [objectreference].TextOrient

[objectreference].TextOrient = textorientvalue

Legal values

<u>Value</u>	<u>Effect</u>
\$LwpTextOrientBottomtotopLefttoright (1913)	Starts text in a layout object from the bottom to the top and then flows the text from left to right.
\$LwpTextOrientBottomtotopRighttoleft (1917)	Starts text in a layout object from the top to the bottom and then flows the text from right to left.
\$LwpTextOrientLefttorightBottomtotop (1916)	Starts text in a layout object from the left to the right and then flows the text from bottom to top.
\$LwpTextOrientLefttorightToptobottom (1910)	Starts text in a layout object from the left to the right and then flows the text from top to bottom.
\$LwpTextOrientRighttoleftBottomtotop (1912)	Starts text in a layout object from the right to the left and then flows the text from bottom to top.
\$LwpTextOrientRighttoleftToptobottom (1914)	Starts text in a layout object from the right to the left and then flows the text from top to bottom.
\$LwpTextOrientToptobottomLefttoright (1915)	Starts text in a layout object from the top to the bottom and then flows the text from left to right.
\$LwpTextOrientToptobottomRighttoleft (1911)	Starts text in a layout object from the top to the bottom and then flows the text from right to left.

Usage

Equivalent to the "Text direction" setting on the Misc panel of the InfoBox for certain layout objects.

Word Pro: TextStyleName property

{button ,AL('H_PREFERENCES_CLASS',0)} [See list of classes](#)

(Read-write)

Data Type

String

Syntax

textstylevalue = [objectreference].TextStyleName

[objectreference].TextStyleName = textstylevalue

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Word Pro: TextStyles property

{button ,AL('H_FOUNDRY_CLASS',0)} [See list of classes](#)

(Read-only) An object created from the TextStyleCollection class. This object provides access to both CharacterStyle and ParagraphStyle objects.

Data Type

[TextStyleCollection](#)

Syntax

textstylesvalue = [objectreference].TextStyles

Legal values

Always contains an instance of the TextStyleCollection class.

Usage

When accessed through the Foundry property on a Division object, the collection object in this property provides access to all the CharacterStyle and ParagraphStyle objects contained in that Division object.

When accessed through the AppFoundry property on the WPAApplication object, this collection object provides access to all the CharacterStyle and ParagraphStyle objects contained in the Word Pro Clipboard.

When accessed through the TempFoundry property on the WPAApplication object, this collection object provides access to all the CharacterStyle and ParagraphStyle objects placed in TempFoundry by WordPro or a script.

When accessed through the Foundry property on the WPAApplication object, this collection object provides access to all the CharacterStyle and ParagraphStyle objects contained in the currently active Division object.

This property is not used in the Foundry property on the TextDocument object.

For more information about collection classes, see [Overview: Word Pro LotusScript Collection Classes](#).

Word Pro: Texts property

{button ,AL('H_FOUNDRY_CLASS',0)} [See list of classes](#)

(Read-only) An object created from the TextCollection class. This object provides access to Text objects.

Data Type

[TextCollection](#)

Syntax

textsvalue = [objectreference].Texts

Legal values

Always contains an instance of the TextCollection class.

Usage

When accessed through the Foundry property on a Division object, the collection object in this property provides access to all the Text objects contained in that Division object.

When accessed through the AppFoundry property on the WPApplication object, this collection object provides access to all the Text objects contained in the Word Pro Clipboard.

When accessed through the TempFoundry property on the WPApplication object, this collection object provides access to all the Text objects placed in TempFoundry by WordPro or a script.

When accessed through the Foundry property on the WPApplication object, this collection object provides access to all the Text objects contained in the currently active Division object.

This property is not used in the Foundry property on the TextDocument object.

For more information about collection classes, see [Overview: Word Pro LotusScript Collection Classes](#).

Word Pro: TextTightness property

{button ,AL('H_FONT_CLASS',0)} [See list of classes](#)

{button ,AL('H_TEXTTIGHTNESS_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Allows you to adjust the spacing between characters.

Data Type

[Integer](#)

Syntax

texttightnessvalue = [objectreference].TextTightness

[objectreference].TextTightness = texttightnessvalue

Legal values

The legal values for this property are from 0 - 100.

Usage

Equivalent to the "Kerning" value located in the Misc panel of the InfoBox for text.

Word Pro: TextViewAttributes property

{button ,AL(^H_CLICKHERE_CLASS;H_TEXTMARKER_CLASS;H_TEXT_CLASS',0)} [See list of classes](#)

(Read-write) Indicates which of three special attributes has been set for viewing the current text stream.

Data Type

Variant (Enumerated Bitmask)

Syntax

textviewattributesvalue = [objectreference].TextViewAttributes

[objectreference].TextViewAttributes = textviewattributesvalue

Legal values

<u>Value</u>	<u>Effect</u>
LwpTextViewCaretVisible (&H1)	Sets the cursor blinking at the current text stream.
LwpTextViewHiddenStoryptr (&H4)	Keeps the window from scrolling when the cursor moves off the screen.
LwpTextViewSelectionVisible (&H2)	The shading is not visible.

Usage

Word Pro: Text property

{button ,AL('H_WPAPPLICATION_CLASS;H_BULLET_CLASS;H_GRAPHICOLEBJECT_CLASS;H_GRAPHIC_CLASS;H_OLEOBJECT_CLASS;H_NOTELAYOUT_CLASS;H_SILVERBULLET_CLASS',0)} [See list of classes](#)

{button ,AL('H_TEXT_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-only) An object created from the Text class.

Data Type

Text

Syntax

textvalue = [objectreference].Text

Legal values

Always contains an instance of the Text class.

Usage

The Text object you retrieve when you call this property is determined by the object from which you call the property.

CurrentApplication.Text

When you call this property from the WPAApplication object, the property returns the Text object which is uppermost in the focus when this property is called.

Note The remainder of this topic is not yet complete.

<Bulletobject>.Text

When you call this property from a Bullet object, the property returns the Text object which

<GraphicOleObjectobject>.Text

When you call this property from a GraphicOleObject object, the property returns the Text object which

<Graphicobject>.Text

When you call this property from a Graphic object, the property returns the Text object which

<NoteLayoutobject>.Text

When you call this property from a NoteLayout object, the property returns the Text object which

<OleObjectobject>.Text

When you call this property from a OleObject object, the property returns the Text object which

<SilverBulletobject>.Text

When you call this property from a SilverBullet object, the property returns the Text object which

Word Pro: ThesMaxNumSynonymsReturned property

{button ,AL(^H_USERINTERFACEPREFS_CLASS',0)} [See list of classes](#)

(Read-write) The maximum number of synonyms returned by the thesaurus.

Data Type

[Integer](#)

Syntax

thesmaxnumsynonymsreturnedvalue = [objectreference].ThesMaxNumSynonymsReturned

[objectreference].ThesMaxNumSynonymsReturned = thesmaxnumsynonymsreturnedvalue

Legal values

Use zero (0) to specify the default of 25 synonyms. Use any other positive integer to specify the corresponding number of synonyms. For example, specify 5 for a maximum of 5, 118 for a maximum of 118, and so on.

Usage

Word Pro: TileWindow property

{button ,AL('H_DOCWINDOW_CLASS',0)} [See list of classes](#)

{button ,AL('H_TILEWINDOW_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write)

Data Type

Variant (Enumerated)

TileType

Syntax

tilewindowvalue = [objectreference].TileWindow

[objectreference].TileWindow = tilewindowvalue

Legal values

\$LwpTileTypeCascade (1946)

\$LwpTileTypeHorz (1948)

\$LwpTileTypeVert (1947)

Usage

Word Pro: Tile property

```
{button ,AL('H_LAYOUT_CLASS;H_CELLGROUPLAYOUT_CLASS;H_CELLLAYOUT_CLASS;H_CONNECTEDLAYOUT_CLASS;H_COLUMNGROUPLAYOUT_CLASS;H_FOOTERLAYOUT_CLASS;H_FRAMELAYOUT_CLASS;H_FRAMEGROUPLAYOUT_CLASS;H_DROPDOWNLAYOUT_CLASS;H_GROUPLAYOUT_CLASS;H_HEADERLAYOUT_CLASS;H_NOTELAYOUT_CLASS;H_PAGELAYOUT_CLASS;H_ROWGROUPLAYOUT_CLASS;H_ROWLAYO  
UT_CLASS;H_RUBYLAYOUT_CLASS;H_SUPERTABLEGROUPLAYOUT_CLASS;H_SUPERTABLELAYOUT  
_CLASS;H_TABLELAYOUT_CLASS;H_ENDNOTELAYOUT_CLASS;H_FOOTNOTELAYOUT_CLASS;H_TABLE  
HEADINGLAYOUT_CLASS;H_TOCSUPERTABLELAYOUT_CLASS',0)} See list of classes
```

```
{button ,AL('H_TILE_PROPERTY_EXSCRIPT',1)} See example
```

(Read-write) Determines whether or not a layout object's graphic content will repeat horizontally and vertically.

Data Type

[Integer](#)

Syntax

tilevalue = [objectreference].Tile

[objectreference].Tile = tilevalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

This property can be used on the Layer property of a layout object to tile a watermark graphic. For example, to tile the watermark graphic on a page layout, you could use the following code:

```
.Page.Layout.Layer.Tile = True
```

Setting the Tile property to True is equivalent to setting the "Placement" option to "Tiled" on the Image properties panel of the InfoBox.

Word Pro: TimedSaveFileExtension property

{button ,AL(^H_USERINTERFACEPREFS_CLASS,0)} [See list of classes](#)

(Read-write) Sets or returns the default file extension for time-saved files.

Data Type

String

Syntax

timedsavefileextensionvalue = [objectreference].TimedSaveFileExtension

[objectreference].TimedSaveFileExtension = timedsavefileextensionvalue

Legal values

This property must contain valid file extension characters, including the leading period. Word Pro uses ".~TS" as a default value for this property.

Usage

For more information on timed saves, see the Word Pro Help topics that relate to automatically saving documents.

For more information on timed saves in LotusScript, see the [TimedSave](#) method topic.

Word Pro: Time property

{button ,AL('H_NOTELAYOUT_CLASS',0)} [See list of classes](#)

(Read-write) The time that a specific note layout was created.

Data Type

Long

Syntax

timevalue = [objectreference].Time

[objectreference].Time = timevalue

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

The value returned by this property represents the number of seconds elapsed since midnight on January 1, 1970.

Word Pro: TitleBarDocNumber property

{button ,AL('H_TEXTDOCUMENT_CLASS',0)} [See list of classes](#)

(Read-write)

Data Type

Integer

Syntax

titlebardocnumbervalue = [objectreference].TitleBarDocNumber

[objectreference].TitleBarDocNumber = titlebardocnumbervalue

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Word Pro: TitleBarVisible property

{button ,AL('H_APPLICATIONWINDOW_CLASS',0)} [See list of classes](#)

(Read-only) A flag that tells you whether or not the TitleBar is turned on.

Data Type

Integer

Syntax

titlebarvisiblevalue = [objectreference].TitleBarVisible

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values. There is no default value.

Usage

If you are in Clean Screen mode, the value is False (0).

Word Pro: Title property

{button ,AL('H_PREFERENCES_CLASS',0)} [See list of classes](#)

(Read-write)

Data Type

String

Syntax

titlevalue = [objectreference].Title

[objectreference].Title = titlevalue

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Word Pro: TOCRange property

{button ,AL('H_TOCSUPERTABLELAYOUT_CLASS',0)} [See list of classes](#)

(Read-write)

Data Type

String

Syntax

toorangevalue = [objectreference].TOCRange

[objectreference].TOCRange = toorangevalue

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Word Pro: TOCSource property

{button ,AL(^H_TOCSUPERTABLELAYOUT_CLASS',0)} [See list of classes](#)

(Read-write) Indicates which portion of a document is included when the TOC is generated.

Data Type

Data type is [Variant](#) which allows the value of this property to be one of the constants listed below or its numeric equivalent (in parentheses).

Syntax

tocsourcevalue = [objectreference].TOCSource

[objectreference].TOCSource = tocsourcevalue

Legal values

<u>Value</u>	<u>Effect</u>
\$LwpGenerateFromCurrentdivision (362)	TOC will be generated from the current division.
\$LwpGenerateFromCurrentleveldivision (361)	TOC will be generated from the current group of divisions.
\$LwpGenerateFromCurrentsection (363)	TOC will be generated from the current section.
\$LwpGenerateFromEntiredocument (360)	TOC will be generated from the entire document.
\$LwpGenerateFromMarker (364)	TOC will be generated from a marker object.

Usage

Corresponds to the "Generate table of contents across" box on the Scope and Placement panel of the Table of Contents Assistant dialog box.

Word Pro: TopBorder property

{button ,AL(^H_BORDERLINES_CLASS;H_GUTTER_CLASS;H_BETWEENLINES_CLASS',0)} [See list of classes](#)

(Read-only) Allows you to access an object's top border object.

Data Type

[Border](#)

Syntax

topbordervalue = [objectreference].TopBorder

Legal values

Always contains an instance of the Border class.

Usage

You can also use the AllBorders property to simultaneously access an object's BottomBorder, LeftBorder, RightBorder, and TopBorder objects.

Word Pro: TopExternalMargin property

```
{button ,AL('H_LAYOUT_CLASS;H_CELLGROUPLAYOUT_CLASS;H_CELLLAYOUT_CLASS;H_CONNECTEDLAY  
OUT_CLASS;H_COLUMNGROUPLAYOUT_CLASS;H_FOOTERLAYOUT_CLASS;H_FRAMELAYOUT_CLASS;H  
_FRAMEGROUPLAYOUT_CLASS;H_DROP_CAP_LAYOUT_CLASS;H_GROUPLAYOUT_CLASS;H_HEADERLAY  
OUT_CLASS;H_NOTELAYOUT_CLASS;H_PAGELAYOUT_CLASS;H_ROWGROUPLAYOUT_CLASS;H_ROWLA  
YOUT_CLASS;H_RUBY_LAYOUT_CLASS;H_SUPERTABLEGROUPLAYOUT_CLASS;H_SUPERTABLELAYOUT  
_CLASS;H_TABLELAYOUT_CLASS;H_ENDNOTELAYOUT_CLASS;H_FOOTNOTELAYOUT_CLASS;H_TABLE  
HEADINGLAYOUT_CLASS;H_TOCSUPERTABLELAYOUT_CLASS',0)} See list of classes
```

```
{button ,AL('H_TOPEXTERNALMARGIN_PROPERTY_EXSCRIPT',1)} See example
```

(Read-write) Allows you to set the amount of margin space that is present above a layout object.

Data Type

Long

Syntax

topexternalmarginvalue = [objectreference].TopExternalMargin

[objectreference].TopExternalMargin = topexternalmarginvalue

Legal values

Data type is Long but the unit of measurement used for this property is Twips. There are 1440 Twips per inch.

Usage

This property can't be set individually for FrameLayout objects within Word Pro. It is combined with all external margin values in the "Padding around border" setting on the Size & Margins panel of the InfoBox.

Word Pro: TopLeftCellRowId property

```
{button ,AL(^H_LAYOUT_CLASS;H_CELLGROUPLAYOUT_CLASS;H_CELLLAYOUT_CLASS;H_CONNECTEDLAY  
OUT_CLASS;H_COLUMNGROUPLAYOUT_CLASS;H_FOOTERLAYOUT_CLASS;H_FRAMELAYOUT_CLASS;H  
_FRAMEGROUPLAYOUT_CLASS;H_DROP_CAPLAYOUT_CLASS;H_GROUPLAYOUT_CLASS;H_HEADERLAY  
OUT_CLASS;H_NOTELAYOUT_CLASS;H_PAGELAYOUT_CLASS;H_ROWGROUPLAYOUT_CLASS;H_ROWLA  
YOUT_CLASS;H_RUBYLAYOUT_CLASS;H_SUPERTABLEGROUPLAYOUT_CLASS;H_SUPERTABLELAYOUT  
_CLASS;H_TABLELAYOUT_CLASS;H_ENDNOTELAYOUT_CLASS;H_FOOTNOTELAYOUT_CLASS;H_TABLE  
HEADINGLAYOUT_CLASS;H_TOCSUPERTABLELAYOUT_CLASS',0)} See list of classes
```

(Read-only) Indicates the row ID of the top left cell in a selection of table cells.

Data Type

Integer

Syntax

toleftcellrowidvalue = [objectreference].TopLeftCellRowId

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, See the overview on [scalar datatypes](#).

Usage

This property is valid only when called from a CellGroupLayout object. You can access the CellGroupLayout object for the current selection with this statement:

```
.Cell.Layout
```

Just add the ToLeftCellRowID property to get the row ID for the top left cell.

```
.Cell.Layout.TopLeftCellRowID
```

Note The row ID is a zero based value, which means that the first row in a table is zero.

Word Pro: TopLevelMenuText property

{button ,AL(^H_OLEOBJECT_CLASS',0)} [See list of classes](#)

(Read-only)

The text displayed in the top menu bar when an OLEObject is selected.

Data Type

String

Syntax

toplevelmenutextvalue = [objectreference].TopLevelMenuText

Legal values

This property is read-only. The value of this property cannot be set by a script.

Usage

This can be useful for double checking the type of an OLEObject.

To get an idea of what this text represents, insert several different types of OLE objects into a Word Pro document, then click on each one in turn. Word Pro changes the menu text to reflect the type of each selected OLE object. The name of each object's menu is the value of this property when called from that object.

Word Pro: Top property

{button ,AL(^H_WINDOW_CLASS;H_APPLICATIONWINDOW_CLASS;H_STATUSBAR_CLASS;H_DOCWINDOW_CLASS',0)} [See list of classes](#)

(Read-write)

[StatusBar]

The top position of the status bar on the workspace.

[ApplicationWindow]

The top position of the application window on the desktop.

Data Type

Long

Syntax

topvalue = [objectreference].Top

[objectreference].Top = topvalue

Legal values

Data type is Long but the unit of measurement used for this property is Twips. There are 1440 Twips per inch.

Usage

[StatusBar]

Use this property to obtain the top position of the status bar on the workspace.

[ApplicationWindow]

Use this property to manipulate the top position of the application window on the desktop.

Word Pro: TotalEditingTime property

{button ,AL('H_DOCINFO_CLASS',0)} [See list of classes](#)

{button ,AL('H_TOTALEDITINGTIME_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read only) The total number of minutes that a document has been open for editing.

Data Type

Long

Syntax

totaleditingtimevalue = [objectreference].TotalEditingTime

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Word Pro: TrailingText property

{button ,AL(`H_FOOTNOTENUMOPT_CLASS;H_ENDNOTEDIVISIONGROUPNUM_CLASS;H_ENDNOTEDIVISIONNUM_CLASS;H_ENDNOTEDOCNUM_CLASS;H_FOOTNOTENUMBERING_CLASS',0)} [See list of classes](#)

{button ,AL(`H_TRAILINGTEXT_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Adds a string of characters after a footnote or endnote reference.

Data Type

String

Syntax

trailingtextvalue = [objectreference].TrailingText

[objectreference].TrailingText = trailingtextvalue

Legal values

The legal values for this property are determined by its data type. For more information about standard LotusScript data types, choose Edit - Scripts & Macros. Choose Show Script Editor. In the Script Editor, choose Help - Lotus Script.

Usage

Equivalent to choosing Create - Footnote/Endnote, clicking Options, selecting "Enclosed by," and inserting text in the "text after" box located on the Numbering panel.

{button ,AL(`H_LEADINGTEXT_PROPERTY_MEMDEF',0)} [See related topics](#)

Word Pro: TypeAboveLine property

{button ,AL(^H_SPACING_CLASS',0)} [See list of classes](#)

(Read-write)

Data Type

Variant

Syntax

typeabovelinevalue = [objectreference].TypeAboveLine

[objectreference].TypeAboveLine = typeabovelinevalue

Legal values**Usage**

Word Pro: TypeAbove property

{button ,AL('H_PARAGRAPHBORDER_CLASS;H_SPACING_CLASS',0)} [See list of classes](#)

{button ,AL('H_TYPEABOVE_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) In the Spacing class, represents the type of line spacing above a paragraph. In the Border class, represents the width of the line above an object.

Data Type

Variant (Enumerated)

Syntax

typeabovevalue = [objectreference].TypeAbove

[objectreference].TypeAbove = typeabovevalue

Legal values

When called from the Border object, the legal values are:

<u>Value</u>	<u>Effect</u>
\$LwpParaBorderWidthMargin (2052)	The border line spans to the margins of the object's container.
\$LwpParaBorderWidthNone (1624)	There is no border line.
\$LwpParaBorderWidthOther (2053)	The length of the border line is user-specified.
\$LwpParaBorderWidthText (2051)	The length is the longest line of text in the paragraph

When called from the Spacing object, the legal values are:

<u>Value</u>	<u>Effect</u>
\$LwpSpacingTypeCustom (1791)	The spacing is user-defined.
\$LwpSpacingTypeDynamic (1789)	The spacing is determined by the largest font on the line.
\$LwpSpacingTypeLeading (1790)	Uses fixed leading for spacing.
\$LwpSpacingTypeNone (1792)	There is no spacing.

Usage

Word Pro: TypeBelow property

{button ,AL('H_PARAGRAPHBORDER_CLASS;H_SPACING_CLASS',0)} [See list of classes](#)

{button ,AL('H_TYPEBELOW_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) In the Spacing class, represents the type of line spacing below a paragraph. In the Border class, represents the width of the line below an object.

Data Type

Variant (Enumerated)

Syntax

typebelowvalue = [objectreference].TypeBelow

[objectreference].TypeBelow = typebelowvalue

Legal values

When called from the Border object, the legal values are:

<u>Value</u>	<u>Effect</u>
\$LwpParaBorderWidthMargin (2052)	The border line spans to the margins of the object's container.
\$LwpParaBorderWidthNone (1624)	There is no border line.
\$LwpParaBorderWidthOther (2053)	The length of the border line is user-specified.
\$LwpParaBorderWidthText (2051)	The length is the longest line of text in the paragraph

When called from the Spacing object, the legal values are:

<u>Value</u>	<u>Effect</u>
\$LwpSpacingTypeCustom (1791)	The spacing is user-defined.
\$LwpSpacingTypeDynamic (1789)	The spacing is determined by the largest font on the line.
\$LwpSpacingTypeLeading (1790)	Uses fixed leading for spacing.
\$LwpSpacingTypeNone (1792)	There is no spacing.

Usage

Word Pro: TypeBetween property

{button ,AL('H_PARAGRAPHBORDER_CLASS','0')} [See list of classes](#)

{button ,AL('H_TYPEBETWEEN_PROPERTY_EXSCRIPT','1')} [See example](#)

(Read-write) The line type of a line between paragraphs.

Data Type

The data type for this property is Variant.

Syntax

typebetweenvalue = [objectreference].TypeBetween

[objectreference].TypeBetween = typebetweenvalue

Legal values

The legal values for this property are listed below. You can use the constant or the numeric value when setting this property. Word Pro always returns the numeric value. You can specify only one of these values at any time.

<u>Value</u>	<u>Effect</u>
\$LwpParaBorderWidthMargin (2052)	The length of the line will adhere to the width of the margins.
\$LwpParaBorderWidthNone (1624)	There will be no line.
\$LwpParaBorderWidthOther (2053)	The line will be of a specific width. Use this parameter in conjunction with the WidthBetween property.
\$LwpParaBorderWidthText (2051)	The length of the line will adhere to the width of the text.

Usage

[ParagraphBorder]

This property is equivalent to the "Line length/position" for the "Between" line options located in the Lines & Shadow Options dialog box.

Word Pro: TypeRight property

{button ,AL('H_PARAGRAPHBORDER_CLASS',0)} [See list of classes](#)

(Read-write)

Data Type

Data type is Variant (Enumerated).

ParaBorderWidth

Syntax

typerightvalue = [objectreference].TypeRight

[objectreference].TypeRight = typerightvalue

Legal values

\$LwpParaBorderWidthMargin (2052)

\$LwpParaBorderWidthNone (1624)

\$LwpParaBorderWidthOther (2053)

\$LwpParaBorderWidthText (2051)

Usage

Word Pro: Type property

{button ,AL('H_CHARACTERSTYLE_CLASS;H_POWERFIELD_CLASS;H_FOOTNOTE_CLASS;H_SPACING_CLASS;H_PARAGRAPHSTYLE_CLASS',0)} [See list of classes](#)

[CharacterStyle]

(Read-only) Indicates the type of a specific object. See the Legal values and Usage sections for information on how this property applies to all classes in which it appears.

[Footnote]

(Read-write) Indicates the type of a specific object. See the Legal values and Usage sections for information on how this property applies to all classes in which it appears.

Data Type

Data type is [Variant](#) which allows the value of this property to be one of the constants listed below or its numeric equivalent (in parentheses).

Syntax

typevalue = [objectreference].Type

Legal values

[CharacterStyle]

<u>Value</u>	<u>Effect</u>
\$LwpCharStyleTypeCharacter (140)	

[Footnote]

<u>Value</u>	<u>Effect</u>
\$LwpFnTypeAnyposition (289)	Used internally by Word Pro.
\$LwpFnTypeAtBottomOfPage (290)	Footnote will be positioned at the bottom of the page.
\$LwpFnTypeAtEndOfDiv (293)	Footnote will be positioned at the end of the division.
\$LwpFnTypeAtEndOfDivGroupSepDiv (296)	Footnote will be positioned at the end of the division group in a separate division.
\$LwpFnTypeAtEndOfDivisionGroup (295)	Footnote will be positioned at the end of the division group.
\$LwpFnTypeAtEndOfDivisionSepDiv (294)	Footnote will be positioned at the end of the division in a separate division.
\$LwpFnTypeAtEndOfDoc (291)	Footnote will be positioned at the end of the document.
\$LwpFnTypeAtEndOfDocSepDiv (292)	Footnote will be positioned at the end of the document in a separate division.

Usage

[Footnote]

Equivalent to a combination of the "Place footnote at" and "Place in separate division" options on the Footnotes dialog box.

Word Pro: Typos property

{button ,AL('H_FORMATCHECKPREF_CLASS',0)} [See list of classes](#)

(Read-write)

Data Type

Data type is Variant (Enumerated Bitmask).

TypoChoices

Syntax

typosvalue = [objectreference].Typos

[objectreference].Typos = typosvalue

Legal values

LwpTypoChoicesListbox1 (&H1)

LwpTypoChoicesListbox2 (&H2)

LwpTypoChoicesListbox3 (&H4)

LwpTypoChoicesListbox4 (&H8)

LwpTypoChoicesListboxall (&HF)

Usage

Word Pro: UIActive property

{button ,AL('H_OLEOBJECT_CLASS',0)} [See list of classes](#)

(Read-only)

Indicates whether or not the user interface for an OLEObject is active at the time this property is called.

Data Type

Integer

Syntax

uiactivevalue = [objectreference].UIActive

Legal values

A value of -1 indicates that the user interface is active. A value of 0 indicates that the user interface is not active. This property is read-only. The value of this property cannot be set by a script.

Usage

The user interface (UI) for an OLE object is comprised of the tool bars and menus which appear when you activate that OLE object. This UI varies from one object to another. For example, an embedded bitmap has a different UI from an embedded 123 spreadsheet.

When you click or double-click an OLE object, you activate its UI. If the UI appears but leaves the Word Pro window in place, the OLE object supports in-place editing. However, if the source application for the OLE object launches and obscures the Word Pro window, then the OLE object you activated **does not** support in-place editing.

Caution Only OLE objects which support in-place editing will return TRUE when their UI is activated.

OLE controls will return TRUE in both design and run mode. In design mode, you activate an OLE control's UI by double-clicking the control. In run mode, you activate an OLE control's UI by using the control (i.e. typing in an edit box or selecting a value from a list).

Word Pro: Underline property

{button ,AL('H_FONT_CLASS',0)} [See list of classes](#)

{button ,AL('H_UNDERLINE_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write)

Data Type

Integer

Syntax

underlinevalue = [objectreference].Underline

[objectreference].Underline = underlinevalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Word Pro: UndoLevels property

{button ,AL(^H_REVISIONDISPLAY_CLASS',0)} [See list of classes](#)

(Read-only)

Data Type

[StringCollection](#)

Syntax

undolevelsvalue = [objectreference].UndoLevels

Legal values

Usage

Word Pro: Units property

{button ,AL(^H_USERINTERFACEPREFS_CLASS',0)} [See list of classes](#)

(Read-only) A collection of all the units of measurement available in Word Pro.

Data Type

[StringCollection](#)

Syntax

unitsvalue = [objectreference].Units

Legal values

Always contains an instance of the StringCollection class with the following members: Inches (in), Centimeters (cm), Points (pts), and Picas (pi).

Usage

This property is used to populate the "Measure in" box on the General panel of the Word Pro Preferences dialog box.

Word Pro: UnknownTagsInCommentNotes property

{button ,AL('H_HTMLOPTIONS_CLASS',0)} [See list of classes](#)

{button ,AL('H_UNKNOWNTAGSINCOMMENTNOTES_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write)

This language element has not yet been defined. You can check for a newer Help file on the Lotus web site.

Data Type

Integer

Syntax

unknowntagsincommentnotesvalue = [objectreference].UnknownTagsInCommentNotes

[objectreference].UnknownTagsInCommentNotes = unknowntagsincommentnotesvalue

Legal values

The legal values for this property are -1 and 0 but you may use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Word Pro: UpdateFields property

{button ,AL('H_PRINTSETTINGS_CLASS',0)} [See list of classes](#)

{button ,AL('H_UPDATEFIELDS_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write) Updates any power fields before the document prints.

Data Type

Integer

Syntax

updatefieldsvalue = [objectreference].UpdateFields

[objectreference].UpdateFields = updatefieldsvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Equivalent to choosing File - Print, clicking Options, and selecting "Fields" in the Update section of the Print Options dialog box.

Word Pro: UpdateOnLoadImmediate property

{button ,AL(^H_POWERFIELD_CLASS',0)} [See list of classes](#)

(Writeonly)

Data Type

Integer

Syntax

[objectreference].UpdateOnLoadImmediate = updateonloadimmediatevalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Word Pro: UpdateOnLoad property

{button ,AL(^H_POWERFIELD_CLASS',0)} [See list of classes](#)

(Read-write)

Data Type

Integer

Syntax

updateonloadvalue = [objectreference].UpdateOnLoad

[objectreference].UpdateOnLoad = updateonloadvalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Word Pro: UpperCase property

{button ,AL('H_FONT_CLASS',0)} [See list of classes](#)

{button ,AL('H_UPPERCASE_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write)

Data Type

[Integer](#)

Syntax

uppercasevalue = [objectreference].UpperCase

[objectreference].UpperCase = uppercasevalue

Legal values

The legal values for this property are -1 and 0. If you prefer, you can use the LotusScript constants of True (-1) and False (0) instead of the integer values.

Usage

Word Pro: URLAddress property

{button ,AL('H_HTMLOPTIONS_CLASS',0)} [See list of classes](#)

{button ,AL('H_URLADDRESS_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write)

This language element has not yet been defined. You can check for a newer Help file on the Lotus web site.

Data Type

String

Syntax

urladdressvalue = [objectreference].URLAddress

[objectreference].URLAddress = urladdressvalue

Legal values

Any value of type String.

Usage

Word Pro: URLDescription property

{button ,AL('H_HTMLOPTIONS_CLASS',0)} [See list of classes](#)

{button ,AL('H_URLDESCRIPTION_PROPERTY_EXSCRIPT',1)} [See example](#)

(Read-write)

This language element has not yet been defined. You can check for a newer Help file on the Lotus web site.

Data Type

String

Syntax

urldescriptionvalue = [objectreference].URLDescription

[objectreference].URLDescription = urldescriptionvalue

Legal values

Any value of type String.

Usage

'Example: ToggleCleanScreen method

'This example enables clean screen, then disables clean screen.

.ToggleCleanScreen

MessageBox "Click OK to toggle Clean Screen mode.", MB_OK, "Example Script"

.ToggleCleanScreen

'Example: ToggleIconBar method

' This example toggles whether or not the SmartIcon bar is displayed.

.ToggleIconBar

'Example: TopExternalMargin property

'This example creates a frame, then changes the padding around the frame's
'border to 1/4 inch (360 twips).

```
.NewFrame 4320, 4320, 3387, 1992, "Default Frame"
```

```
.Frame.Layout.LeftExternalMargin = 360
```

```
.Frame.Layout.TopExternalMargin = 360
```

```
.Frame.Layout.RightExternalMargin = 360
```

```
.Frame.Layout.BottomExternalMargin = 360
```

```
'Example: TotalEditingTime property
'This example prints the date and time the document was modified,
'and the total editing time to the LotusScript output panel.
MDate = .ActiveDocument.Docinfo.ModifiedDateString
MTime = .ActiveDocument.Docinfo.ModifiedTimeString
ETime = .ActiveDocument.Docinfo.TotalEditingTime
Print "The current document was modified on " & MDate & " at " & MTime & ". " & ETime
& " minutes have been spent editing the document."
```

'Example: TrailingText property

'This example sets the trailing text option for footnotes and endnotes
'to the period character.

.Division.FootnoteOptions.FootnoteNumbering.**TrailingText** = "."

.Division.FootnoteOptions.EndnoteDivisionNum.**TrailingText** = "."

.Division.FootnoteOptions.EndnoteDivisionGroupNum.**TrailingText** = "."

.Division.FootnoteOptions.EndnoteDocNum.**TrailingText** = "."

'Example: TypeAbove property

'This example sets the spacing above the current paragraph to a custom value
'of 12 points.

.Text.Spacing.**TypeAbove** = \$LwpSpacingTypeCustom

.Text.Spacing.AmountOfSpaceAbove = 240 ' 240 twips = 12 points

'Example: TypeBelow property

'This example sets the spacing below the current paragraph to a custom value
'of 12 points.

.Text.Spacing.**TypeBelow** = \$LwpSpacingTypeCustom

.Text.Spacing.AmountOfSpaceBelow = 240 ' 240 twips = 12 points

```
'Example: TypeBetween property
'This example inserts some text into the current document, then places
'a line the width of the text between the paragraphs.

' Insert some text, then select it.
.Text.InsertText "This is an example paragraph.", True
.Text.InsertText "This is another example paragraph.", True
.Text.Backward $LwpNavigateObjectTypeParagraph , 2
' Set up the borders
.Text.ParagraphBorder.Betweenlines.LinePlacement = &Hf
.Text.ParagraphBorder.Betweenlines.AllBorders.Pattern = $LtsBorderPatternSolid
.Text.ParagraphBorder.Betweenlines.AllBorders.WidthInTwips = 20 ' 1 point line
' make the line red
.Text.ParagraphBorder.Betweenlines.AllBorders.Color.Red = 255
.Text.ParagraphBorder.Betweenlines.AllBorders.Color.Blue = 0
.Text.ParagraphBorder.Betweenlines.AllBorders.Color.Green = 0
.Text.ParagraphBorder.Betweenlines.AllBorders.Color.Override = $LwpColorOverrideRgb
'Put 1/10 inch between the text and the line
.Text.ParagraphBorder.MarginBetween = 144
'Make the line as long as the paragraph
.Text.ParagraphBorder.TypeBetween = $LwpParaBorderWidthText
```

```
'Example: Type method
' This example uses the Type method to issues keystrokes which insert two
' lines of text into the current document and then select both lines.
' RUNTIME DEPENDENCIES: You must have a document open for this script to work.

.BeginChange ' use this method with multiple Type commands in a sub or function
.Type("Some sample [TAB] text.")
.Type("[ENTER]")
.Type("More sample [TAB] text.")
.Type("[ENTER]")
.Type("[SHIFTUP][SHIFTUP]")
.EndChange
```

'Example: Underline method

' This example first inserts sample text in the current document and selects
' the paragraph. The script then uses the Underline method to toggle the
' underline attribute.

' RUNTIME DEPENDENCIES: You must have a document open with selected text
' for this script to work.

.Text.InsertText "This is some sample text."

.SelectParagraph

.Underline

'Example: Underline property

'This example enables underlining, types some text, then disables the attribute.

```
.Text.Font.Underline = True
```

```
.Text.InsertText "This is underlined text."
```

```
.Text.Font.Underline = False
```

'Example: Undo method

' This example types some text into the current document which is then undone.

' RUNTIME DEPENDENCIES: You must have a document open for this script to work.

.Type "Typing this text wil be be undone"

.Undo

```
'Example: UnknownTagsInCommentNotes property
'This example asks the user whether to place unknown tags in comment notes during HTML
import.

stat = MessageBox ("Do you want to import unknown HTML tags in comments?", 36,
"Example Script")
If stat = 6 Then   ' user said yes
    .ApplicationWindow.UserInterfacePrefs.HTMLOptions.UnknownTagsInCommentNotes = True
Else
    .ApplicationWindow.UserInterfacePrefs.HTMLOptions.UnknownTagsInCommentNotes =
False
End If
```

```
'Example: UnlinkFrameContents method
'This example creates two frames, and then links their contents.
'After a message box is displayed, the frames are unlinked.

'Create the first frame, name it, and insert some text.
.NewFrame 1386, 906, 1842, 2007, "Default Frame"
.Frame.Layout.Name = "FirstFrame"
For i = 1 To 5
    .Text.InsertText "This is some text inside the frame.", False
Next
'Create second frame and name it.
.NewFrame 1536, 891, 4227, 2067, "Default Frame"
.Frame.Layout.Name = "SecondFrame"
'Link the two frames
.Foundry.Frames.Item("FirstFrame").LinkFrame = "SecondFrame"
MessageBox "Click OK to unlink the frames.", MB_OK, "Example Script"
'Go to the 'parent' frame to unlink
.GoToObject "Frame", True
.UnlinkFrameContents
```

```
'Example: UnregisterWPDataSet method
' This example creates a dataset named 'PhoneNumbers' off of the application
' object. Two dataset items are added and then printed to the Script Editor
' Output panel. The dataset is then removed.
' RUNTIME DEPENDENCIES: You must have a document open for this script to work.
```

```
Dim AppDataSet As WPDataSet
Set AppDataSet = .RegisterWPDataSet("PhoneNumbers")
AppDataSet.SetData "Mark", "555-1234"
AppDataSet.SetData "Peyton", "555-5678"
Print AppDataSet.GetData("Mark", " ")
Print AppDataSet.GetData("Peyton", " ")
```

```
.UnRegisterWPDataSet "PhoneNumbers"
```

'Example: UpdateFields property

'This example asks the user whether to update Power Fields before printing,

'sets the appropriate option, and prints the document.

```
stat = MessageBox ("Do you want to update Power Fields before printing?", 36, "Example  
Script")
```

```
If stat = 6 Then ' user said yes
```

```
    .ActiveDocument.PrintSettings.UpdateFields = True
```

```
Else
```

```
    .ActiveDocument.PrintSettings.UpdateFields = False
```

```
End If
```

```
.Print
```

'Example: UpdateFootersText method

'This example creates a new division, then inserts some text in the footer.

'The UpdateFootersText method copies the footer to the original division as well.

```
.ActiveDocWindow.WinViewPrefs.IsViewSectionTabs = True
```

```
.ApplicationWindow.SectionTabs.AddNewSectionTabs
```

```
.GoToObject "Footer", True
```

```
.Text.InsertText "This is a footer for the document"
```

```
.UpdateFootersText
```

```
.Container.Start $LwpDocumentObjectTypeDocument
```

'Example: UpdateHeadersText method

'This example creates a new division, then inserts some text in the header.

'The UpdateHeadersText method copies the header to the original division as well.

```
.ActiveDocWindow.WinViewPrefs.IsViewSectionTabs = True
```

```
.ApplicationWindow.SectionTabs.AddNewSectionTabs
```

```
.GoToObject "Header", True
```

```
.Text.InsertText "This is a header for the document"
```

```
.UpdateHeadersText
```

```
.Container.Start $LwpDocumentObjectTypeDocument
```


'Example: UpdatePageSizeChange method

' This example updates any changes to the size of the paper used to print the document.

' RUNTIME DEPENDENCIES: You must have a document open for this script to work.

.ActiveDocument.PrintManager.UpdatePageSizeChange

```
'Example: UpdatePowerFields method  
'This example inserts some text into the document, then the number of words  
'power field, and finally more text. The UpdatePowerFields method is  
'then called to update the word count.
```

```
.Text.InsertText "This is some sample text which will be used to test the  
UpdatePowerFields method. "  
.InsertField "NumWords", False, False, False  
.Text.InsertText " This is some more text. "  
MessageBox "Click OK to update Power Fields", MB_OK, "Example Script"  
.ActiveDocument.UpdatePowerFields
```

```
'Example: UpdateSelectedFields method
'This example inserts some text, a document field, and some more text. The document
fields are updated, and the result is shown on the screen.

.Text.InsertText "This is some sample text which will be used to test the update
fields method. "
.InsertDocInfo $LwpDocVarNumwords
.Text.InsertText " This is some more text. "
MessageBox "Click OK to update the word count.", MB_OK, "Example Script"
.ActiveDocument.DocInfo.UpdateSelectedFields
.ActiveDocWindow.Update
```

'Example: UpperCase method

' This example toggles the uppercase attribute of the selected text, displays
' a message box, then toggles the uppercase again.

' RUNTIME DEPENDENCIES: You must have a document open and some text selected
' for this script to work.

.UpperCase

MessageBox "Click OK undo uppercase change.",MB_OK,"Example Script"

.UpperCase

'Example: UpperCase property

'This example enables uppercase, types some text, then disables the attribute.

```
.Text.Font.UpperCase = True
```

```
.Text.InsertText "This is capitalized text."
```

```
.Text.Font.UpperCase = False
```

'Example: URLAddress property

'This example causes Word Pro to automatically create a link at the bottom of
'documents exported to HTML. The link is to the URL "www.lotus.com"

' First, tell Word Pro to create the link...

.ApplicationWindow.UserInterfacePrefs.HTMLOptions.IncludeURL = True

' ... then provide the address and description for the link

.ApplicationWindow.UserInterfacePrefs.HTMLOptions.**URLAddress** = "www.lotus.com"

.ApplicationWindow.UserInterfacePrefs.HTMLOptions.URLDescription = "Lotus Home Page"

'Example: URLDescription property

'This example causes Word Pro to automatically create a link at the bottom of
'documents exported to HTML. The link is to the URL "www.lotus.com"

' First, tell Word Pro to create the link...

.ApplicationWindow.UserInterfacePrefs.HTMLOptions.IncludeURL = True

' ... then provide the address and description for the link

.ApplicationWindow.UserInterfacePrefs.HTMLOptions.URLAddress = "www.lotus.com"

.ApplicationWindow.UserInterfacePrefs.HTMLOptions.**URLDescription** = "Lotus Home Page"

```
'Example: UseConsistentSpaceBetweenSentences property
'This example asks the user whether to check for consistent spacing between
'sentences in Format Check, and then sets the appropriate option.

stat = MessageBox ("Do you want to check sentence spacing errors?", 36, "Example
Script")
If stat = 6 Then ' user said yes
    .Application.FormatCheckPreferences.UseConsistentSpaceBetweenSentences = True
Else
    .Application.FormatCheckPreferences.UseConsistentSpaceBetweenSentences = False
End If
```



```
'Example: UseCycleKeys property
'This example asks the user whether enable CycleKeys instead of using
'function keys to select styles, and then sets the appropriate option.

stat = MessageBox ("Do you want to enable CycleKeys?", 36, "Example Script")
If stat = 6 Then ' user said yes
    .ApplicationWindow.UserInterfacePrefs.UseCycleKeys = True
Else
    .ApplicationWindow.UserInterfacePrefs.UseCycleKeys = False
End If
```

'Example: UseGreeting property

'This example creates a greeting for the current document, and enables
'the greeting to be displayed.

```
.ActiveDocument.DocControl.UseGreeting = True
```

```
.ActiveDocument.DocControl.Greeting = "Please review and return with your comments  
ASAP."
```

```
'Example: UseNextStyle property
'This example types some text into the current document.
'The option to use a specific style following the current paragraph is set and
demonstrated.
.Type "This is some text using the Default Text Style.[Enter]"
.Text.Breaks.NextStyleName = "Bullet 1" ' specify the style name to use for the next
paragraph.
.Text.Breaks.UseNextStyle = True ' set the option to use the style name.
.Type "The next paragraph will be Bullet 1[Enter]"
.Type "This paragraph uses Bullet 1"
```

'Example: UsePrinterSettings property

'This example asks the user whether to use printer settings to determine page
'size and orientation, and then sets the appropriate option.

```
stat = MessageBox ("Do you want to base page settings on printer settings?", 36,  
"Example Script")
```

```
If stat = 6 Then ' user said yes
```

```
    .Page.Layout.RightPage.UsePrinterSettings = True
```

```
Else
```

```
    .Page.Layout.RightPage.UsePrinterSettings = False
```

```
End If
```

'Example: UseRelative property

'This example sets up text indented 1/4 inch from the indent level of the
'previous higher-level paragraph.

.Text.InsertText "Now is the time for all good men to come to the aid of their
country. "

.Text.Indent.All = 360

.Text.Indent.**UseRelative** = True

'Example: UserInitials property

'This example reads the current settings for the user's initials, and displays
'an input box, letting the user change them. The changes are then saved.

```
CurrentValue = .Preferences.UserInitials
```

```
NewValue = InputBox ("Type the new initials", "Example Script", CurrentValue)
```

```
.Preferences.UserInitials = NewValue
```

'Example: UserName property
'This example creates a new section, then gets it's internal name.
'The section is then renamed to "Test Section".

```
Dim ThisSection as String  
.InsertSection "Default Page", True, True, $LwpStartTypeNextpage, False, True  
ThisSection = .Text.SectionName  
.Division.Foundry.Sections(ThisSection).UserName = "Test Section"
```

```
'Example: UseSeparatorLine property
'This example asks the user whether to use a separator line above footnotes,
'and continued footnotes, then sets the appropriate option.

stat = MessageBox ("Do you want a separator line above footnotes?", 36, "Example
Script")
If stat = 6 Then ' user said yes
'set option for line above regular notes
    .Division.FootnoteOptions.FootnoteSeparator.UseSeparatorLine = True
'set option for line above continued notes
    .Division.FootnoteOptions.FootnoteContSep.UseSeparatorLine = True
Else
    .Division.FootnoteOptions.FootnoteSeparator.UseSeparatorLine = False
    .Division.FootnoteOptions.FootnoteContSep.UseSeparatorLine = True
End If
```


'Example: UsesHelp property

'This example creates a Click Here block, and sets the Help text for the block.

```
Dim ClickHereName as String
```

```
ClickHereName = .InsertClickHere()
```

'First enable the Help text

```
.Division.Foundry.ClickHeres(ClickHereName).UsesHelp = True
```

'And set the Help text contents

```
.Division.Foundry.ClickHeres(ClickHereName).HelpText = "Click here to type Text"
```

'Example: UseSuperscriptReferenceNum property

'This example sets the option for superscripting the footnote reference number
'to on for footnotes, and off for endnotes at the end of the document.

.Division.FootnoteOptions.FootnoteNumbering.**UseSuperscriptReferenceNum** = True

.Division.FootnoteOptions.EndnoteDocNum.**UseSuperscriptReferenceNum** = False

```
'Example: UseTwoSpacesBetweenSentences property
'This example asks the user whether to check for two spaces between sentences,
'and then sets the appropriate option.

stat = MessageBox ("Do you want two spaces between sentences?", 6, "Example Script")
If stat = 6 Then ' user said yes
    .Application.FormatCheckPreferences.UseTwoSpacesBetweenSentences = True
Else
    .Application.FormatCheckPreferences.UseTwoSpacesBetweenSentences = False
End If
```

'Example: VersionManager property

'This example creates a new version in the current document, then adds a remark to the new version.

```
EdInitials = .EditorManager.CurrentEditor.EditorInitials
```

```
.ActiveDocument.VersionManager.CreateVersion "New Version"
```

```
.ActiveDocument.VersionManager.CurrentVersion.CreateRemark "This is a new remark",  
872846578, EdInitials
```

'Example: VersionName property
'This example prints the name of all the versions in the current document
'to the LotusScript output panel.

```
Forall Version In .ActiveDocument.VersionManager.Versions  
    Print Version.VersionName  
End Forall
```

'Example: Versions property

'This example prints the name of all the versions in the current document
'to the LotusScript output panel.

```
Forall Version In .ActiveDocument.VersionManager.Versions  
    Print Version.VersionName  
End Forall
```

```
'Example: VerticalSplitWindow property
' create a new document, then split the document's window vertically

.NewDocument
' set the option for a vertical window
.ApplicationWindow.UserInterfacePrefs.VerticalSplitWindow = true
' the NewWindow method actually executes the split
.NewWindow
```

'Example: VertRuler property
'This example displays the vertical ruler if it is not displayed,
'and hides the ruler if it is.

```
If Not .ApplicationWindow.VertRuler.IsShowing Then
    .ActiveDocWindow.WinViewPrefs.IsViewVertRuler = True
Else
    .ActiveDocWindow.WinViewPrefs.IsViewVertRuler = False
End If
```


'Example: ViewLevel property

'This example sets several view levels for the current document.

```
MessageBox "Click OK to display the current document at 75% magnification", MB_OK,  
"Example Script"
```

```
.ActiveDocWindow.ViewLevel = 75
```

```
MessageBox "Click OK to display the current document at 100% magnification", MB_OK,  
"Example Script"
```

```
.ActiveDocWindow.ViewLevel = 100
```

'Example: ViewType property

'This example sets the custom view level to 91% and displays the custom view.

%INCLUDE "WPBITMSK.LSS" ' needed to set value for LwpViewsCustom.

.ActiveDocWindow.WinViewPrefs.CustomViewLevel = 91

.ActiveDocWindow.WinViewPrefs.**ViewType** = LwpViewsCustom

```
'Example: Visible property
' Toggle the status bar
If .ApplicationWindow.StatusBar.Visible = True Then
    if it's visible, hide the bar
    .ApplicationWindow.StatusBar.HideStatusBar
Else    ' display the bar
    .ApplicationWindow.StatusBar.ShowStatusBar
End If
```

'Example: WasDeletedInRevMarkMode property

'This example creates a frame, then asks the user whether it should be

'marked as revision deleted. It then performs the action.

.CreateFrame

stat = MessageBox ("Do you want to mark this frame as revision deleted?", 36, "Example Script")

If stat = 6 Then ' user said yes

 .Frame.Layout.**WasDeletedInRevMarkMode** = True

 .Frame.Layout.WasInsertedInRevMarkMode = False

Else

 .Frame.Layout.**WasDeletedInRevMarkMode** = False

End If

'Example: WasInsertedInRevMarkMode property
'This example creates a frame, then asks the user whether it should be
'marked as revision inserted. It then performs the action.

```
.CreateFrame
stat = MessageBox ("Do you want to mark this frame as revision inserted?", 36,
"Example Script")
If stat = 6 Then ' user said yes
    .Frame.Layout.WasInsertedInRevMarkMode = True
    .Frame.Layout.WasDeletedInRevMarkMode = False 'can't be both
Else
    .Frame.Layout.WasInsertedInRevMarkMode = False
End If
```

'Example: Where property

'This example creates a frame, then sets the layout to "On current page".

'Finally the frame is anchored to the lower right of the page edge.

```
.NewFrame 3656, 1173, 3389, 2155, "Default Frame"
```

```
.Frame.Anchor $LwpAnchorWhereLayout, $LwpConditionTypeOnlyspecificpage,  
$LwpRelativeTypeLytParent
```

```
.Frame.Layout.Where = $LwpWhereTypeLowerRight
```

'Example: WidowOrphan property

'This example asks the user whether to allow the last line of a paragraph
'at the beginning of a page, or the first line of a paragraph at the end of
'a page within the current document, and then sets the appropriate option.

```
stat = MessageBox ("Do you want to allow widow and orphan lines?", 36, "Example  
Script")
```

```
If stat = 6 Then ' user said yes
```

```
    .ActiveDocument.DocOptions.WidowOrphan = True
```

```
Else
```

```
    .ActiveDocument.DocOptions.WidowOrphan = False
```

```
End If
```

'Example: WidthAbove property

'This example creates a line above the current paragraph, and

'sets its width to 3 inches.

```
.Text.ParagraphBorder.BorderLines.LinePlacement = &H4 ' put border on top  
.Text.ParagraphBorder.BorderLines.TopBorder.Pattern = $LtsBorderPatternSolid  
.Text.ParagraphBorder.BorderLines.TopBorder.WidthInTwips = 20  
.Text.ParagraphBorder.MarginTop = 144  
.Text.ParagraphBorder.TypeAbove = $LwpParaBorderWidthOther  
.Text.ParagraphBorder.WidthAbove = 4320 ' 3 inches
```


'Example: WidthBelow property

'This example creates a line below the current paragraph, and

'sets its width to four inches.

```
.Text.ParagraphBorder.BorderLines.LinePlacement = &H8 ' bottom of object
```

```
.Text.ParagraphBorder.BorderLines.BottomBorder.Pattern = $LtsBorderPatternSolid
```

```
.Text.ParagraphBorder.BorderLines.BottomBorder.WidthInTwips = 20
```

```
.Text.ParagraphBorder.MarginBottom = 144
```

```
.Text.ParagraphBorder.TypeBelow = $LwpParaBorderWidthOther
```

```
.Text.ParagraphBorder.WidthBelow = 5760 ' set the width to 4 inches
```

'Example: WidthBetween property

'This example inserts some text into the current document, then places
'a five inch line between the paragraphs.

' Insert some text, then select it.

.Text.InsertText "This is an example paragraph.", True

.Text.InsertText "This is another example paragraph.", True

.Text.Backward \$LwpNavigateObjectTypeParagraph , 2

' Set up the borders

.Text.ParagraphBorder.Betweenlines.LinePlacement = &Hf

.Text.ParagraphBorder.Betweenlines.AllBorders.Pattern = \$LtsBorderPatternSolid

.Text.ParagraphBorder.Betweenlines.AllBorders.WidthInTwips = 20 ' 1 point line

' make the line red

.Text.ParagraphBorder.Betweenlines.AllBorders.Color.Red = 255

.Text.ParagraphBorder.Betweenlines.AllBorders.Color.Blue = 0

.Text.ParagraphBorder.Betweenlines.AllBorders.Color.Green = 0

.Text.ParagraphBorder.Betweenlines.AllBorders.Color.Override = \$LwpColorOverrideRgb

'Put 1/10 inch between the text and the line

.Text.ParagraphBorder.MarginBetween = 144

'Make the line as long as the paragraph

.Text.ParagraphBorder.TypeBetween = \$LwpParaBorderWidthOther

.Text.ParagraphBorder.**WidthBetween** = 7200 ' 5 inches

'Example: WidthInTwips property

'This example creates a frame, then sets double lines with a width of 40 twips

'around all borders of the frame.

.CreateFrame

.Frame.Layout.BorderLines.AllBorders.Pattern = \$LtsBorderPatternDouble

.Frame.Layout.BorderLines.AllBorders.**WidthInTwips** = 40

```
'Example: Width property
' This example creates a table, then prints the width of the active cell
' to the LotusScript output panel.

.CreateTable
TwipCellWid = .Cell.Width
Print "The cell width is " & TwipCellWid/1440 & " inches."
```

'Example: WindowsName property

'This example inserts and selects some text, then sets the font to "Swiss".

'The actual name and windows name of the font is then printed to the LotusScript output panel.

'Since most computers don't have the Swiss font installed, the ActualName property will be set to Arial, which is very close to Swiss.

```
.Text.InsertText "This is a font test."  
.SelectSentence  
.Text.Font.FontName = "Swiss"  
Print .Text.Font.ActualName ' This is the actual font being used.  
Print .Text.Font.WindowsName ' This is the font the user selected.  
.Text.Font.FontName = .Text.Font.ActualName
```

```
'Example: WinViewPrefs property
'This example displays the current document in draft view, then in the previous view.
.ActiveDocWindow.WinViewPrefs.IsInDraft = True
Msgbox "Click OK to turn off draft view." , MB_OK, "Example Script"
.ActiveDocWindow.WinViewPrefs.IsInDraft = False
```

'Example: WithComments property

'This example asks the user whether to print comment notes, then sets the
'appropriate option, and prints the document.

```
stat = MessageBox ("Do you want to print comment notes?", 36, "Example Script")
```

```
If stat = 6 Then ' user said yes
```

```
    .ActiveDocument.PrintSettings.WithComments = True
```

```
Else
```

```
    .ActiveDocument.PrintSettings.WithComments = False
```

```
End If
```

```
.Print
```

```
'Example: WmCommand method
' This example display the Word Pro Welcome dialog using the WmCommand
' function. The list of defined values for the WmCommand function are
' located in the "wpbitmsk.lss"file.
' RUNTIME DEPENDENCIES: You must have a document open for this script to work.
Const LwpMenuMfWelcome = &H7AD
.WmCommand(LwpMenuMfWelcome)
```


'Example: WordDoubleUnderline property

'This example enables word double underlining, types some text, then disables the attribute.

```
.Text.Font.WordDoubleUnderline = True
```

```
.Text.InsertText "This is double underlined text."
```

```
.Text.Font.WordDoubleUnderline = False
```

```
'Example: WordUnderline method
' This example toggles the word underline attribute of the selected text.
' RUNTIME DEPENDENCIES: You must have a document open and some text selected ' for
this script to work.

.WordUnderline
MessageBox "Click OK undo word underline change.",MB_OK,"Example Script"
.WordUnderline
```

'Example: WordUnderline property

'This example enables word underlining, types some text, then disables the attribute.

```
.Text.Font.WordUnderline = True
```

```
.Text.InsertText "This is word underlined text."
```

```
.Text.Font.WordUnderline = False
```

```
'Example: WorkingType property
'This example asks the user whether to use the previously selected file type
'when showing the Browse dialog box, and then sets the appropriate option.

stat = MessageBox ("Do you want to use the working file type?", 36, "Example Script")
If stat = 6 Then ' user said yes
    .ApplicationWindow.UserInterfacePrefs.WorkingType = True
Else
    .ApplicationWindow.UserInterfacePrefs.WorkingType = False
End If
```

```

'Example: WPDataSets property
' This example creates a dataset named 'ExampleDataSet' off of the active
' document. The 'FirstName' and 'LastName' items are created and filled with data.
' Finally the values for the dataset items are printed to the Script Editor
' Output panel. Since no dataset item named 'Address' was defined, the default
' dataset value will be printed in the last statement.
' RUNTIME DEPENDENCIES: You must have a document open for this script to work.

Dim DataSetName as String
Dim DefValue as String
' establish a variable representing all data sets in this document
Dim DocDataSets As WPDataSetCollection
Set DocDataSets = .ActiveDocument.WPDataSets

' give a new name to the data set I'm creating
DataSetName = "ExampleDataSet"
DefValue = "Default"

' assign values to new set
DocDataSets(DataSetName).SetData "FirstName","John"
DocDataSets(DataSetName).SetData "LastName","Doe"

' retrieve values from set
Print DocDataSets(DataSetName).GetData("FirstName",DefValue)
Print DocDataSets(DataSetName).GetData("LastName",DefValue)
' this will display default value, since address not defined.
Print DocDataSets(DataSetName).GetData("Address",DefValue)

```

```
'Example: WrapType property
'This example types some text in the current document, then creates a frame
'on top of the text. The frame's placement is set to No Wrap Beside, so that
'the text flows above and below the frame.

For i = 1 To 20
    .Text.InsertText "Now is the time for all good men to come to the aid of their
country.  "
Next
.NewFrame 1506, 1989, 2505, 2572, "Default Frame"
.Frame.Layout.WrapType = $LwpWrapTypeLayoutNoWrapBeside
```

```
'Example: Write method
' This example creates a bag in the active division and then writes some data
' to the bag. The data from the created bag is read and printed to the Lotus
' Script Output panel. Next, data from all bags in the Bag Collection is
' printed.
' RUNTIME DEPENDENCIES: You must have a document open for this script to work.
```

```
Dim BagName As String
Dim MyBag As Bag
Dim BagData As String
```

```
BagData = "This is data for the bag."
LenBagData = Len(BagData)
```

```
BagName = .Division.Foundry.Create($LwpFoundryCreateTypeBag)
Set MyBag = .Division.Foundry.Bags.Item(BagName)
```

```
Stat = MyBag.Write(BagData, LenBagData)
```

```
If Stat = True Then
    Print "BagData= " & MyBag.Read(LenBagData)
End If
```

```
Forall ThisBag In .Division.Foundry.Bags
    ThisBag.Reset
    Print "Name = " ThisBag.Name
    Print "Length = " ThisBag.Length
    Print ThisBag.Read(ThisBag.Length)
End Forall
```

'Example: XOffset property

'This example imports a graphic into the current document, then positions the image 1/2 inch from the left edge of the frame.

```
GPath = .ApplicationWindow.UserInterfacePrefs.GraphicPath
.ImportGraphic GPath & "\_wpdon.gif", ".gif", False, False, "Default Graphic/OLE"
.Frame.Layout.XOffset = 720 ' offset 1/2 inch from left
```


'Example: XPosition property
'This example creates a frame, and places a shadow on the bottom
'and right sides of the frame.

```
.CreateFrame  
.Frame.Layout.Shadow.XPosition = 101  
.Frame.Layout.Shadow.YPosition = 101
```

'Example: YOffset property

'This example imports a graphic into the current document, then positions the image 1/4 inch from the top edge of the frame.

```
GPath = .ApplicationWindow.UserInterfacePrefs.GraphicPath
.ImportGraphic GPath & "\_wpdon.gif", ".gif", False, False, "Default Graphic/OLE"
.Frame.Layout.YOffset = 360 ' offset 1/4 inch from top
```

'Example: YPosition property
'This example creates a frame, and places a shadow on the bottom
'and right sides of the frame.

```
.CreateFrame  
.Frame.Layout.Shadow.XPosition = 101  
.Frame.Layout.Shadow.YPosition = 101
```

'Example: Zero property

'This example creates a table, and inserts a zero in the first cell. The cell
'is then formatted to display in blue if its value is zero.

```
.CreateTable False, "Default Table", 2, 3  
.Type "0"  
.Table.CurrentCell.NumericFormat.Zero.Prefix = ""  
.Table.CurrentCell.NumericFormat.Zero.Suffix = ""  
.Table.CurrentCell.NumericFormat.Zero.ColorOverride = True  
.Table.CurrentCell.NumericFormat.Zero.Color.Red = 0  
.Table.CurrentCell.NumericFormat.Zero.Color.Blue = 255  
.Table.CurrentCell.NumericFormat.Zero.Color.Green = 0  
.Table.CurrentCell.NumericFormat.Zero.Color.Override = $LwpColorOverrideRgb
```

'Example: ZipCode property
'This example gets the current value of the ZIP code, and uses it as the default
'for an input asking the user to enter a new code. The ZipCode property is then
'reset with the new value.

```
CurrentValue = .Preferences.ZipCode  
NewValue = Inputbox ("What Zip Code do you want to use?", "Example Script",  
CurrentValue)  
.Preferences.ZipCode = NewValue
```

Word Pro LotusScript A-Z

A
B
C
D
E
F
G
H
I
J
K
L
M
N
O
P
Q
R
S
T
U
V
W
X
Y
Z

A

[Abandon method](#)
[Abilities property](#)
[AbortURLDownload method](#)
[AboutToReturnToOriginator property](#)
[AbsoluteOn property](#)
[AbsoluteTextOrientation property](#)
[AbsoluteXPos property](#)
[AbsoluteYPos property](#)
[Accelerator property](#)
[Accelerators property](#)
[Accelerators class](#)
[AccessibilityOn property](#)
[AcquireTWAImage method](#)
[Action property](#)
[Activate method](#)
[ActivateAs method](#)
[Active property](#)
[ActiveDocument property](#)
[ActiveDocWindow property](#)
[ActualEnumName property](#)
[ActualName property](#)
[Add method](#)
[AddAccelerators method](#)
[AddACondition method](#)
[AddBookmark method](#)
[AddChildToLayout method](#)
[AddContainer method](#)
[AddDdeLink method](#)
[AddDivision method](#)

[AddDivisionToPrint method](#)
[AddEditorManager method](#)
[AddField method](#)
[AddIcon method](#)
[AddIndexAllEntry method](#)
[AddIndexEntry method](#)
[AddLayoutOverride method](#)
[AddNewSectionTabs method](#)
[AddOutlineSequenceItem method](#)
[AddPopupGraphicItem method](#)
[AddPopupPointSizeItem method](#)
[AddPopupTextItem method](#)
[Address1 property](#)
[Address2 property](#)
[AddSectionTabs method](#)
[AddSmartCorrect method](#)
[AddStringToList method](#)
[AddTOCEntry method](#)
[AddVerbMenu method](#)
[AdjustShade method](#)
[Adopt method](#)
[AdviseOnRename method](#)
[AdviseOnSave method](#)
[Afid property](#)
[AfidClassName property](#)
[Align property](#)
[Alignment property](#)
[Alignment class](#)
[AlignmentChar property](#)
[AlignmentType property](#)
[AlignStyleName property](#)
[All property](#)
[AllBorders property](#)
[AllowAlternateVerification property](#)
[AllowListEdit property](#)
[AllowListMultiValues property](#)
[AllowResizeWhenCrop property](#)
[AlternateName property](#)
[Always property](#)
[Amikake property](#)
[Amikake class](#)
[AmikakeName property](#)
[AmikakeType property](#)
[Amount property](#)
[AmountOfSpaceAbove property](#)
[AmountOfSpaceAboveLine property](#)
[AmountOfSpaceBelow property](#)
[AmtTether property](#)
[AmtToRotateContent property](#)
[AmtToTetherFrom property](#)
[Anchor method](#)
[Animated property](#)
[AnswerMsgBox method](#)
[AnyEdits method](#)
[AnyNumber property](#)
[AnyOLEDELinks property](#)
[AppendMacro method](#)
[AppFoundry property](#)

Application class
Application property
ApplicationWindow property
ApplicationWindow class
ApplyAdjectivePos property
ApplyAdjectNounPart property
ApplyAgreementWithHereThere property
ApplyAnglicisms property
ApplyArchaicExpressions property
ApplyArticleAgreement property
ApplyBadComparatives property
ApplyBadInflection property
ApplyBadNoun property
ApplyBadNounGender property
ApplyBadPlural property
ApplyBadPrepositions property
ApplyBelgianExpression property
ApplyBorrowedForeign property
ApplyBureuaJargon property
ApplyCalque property
ApplyCapitalizationCheck property
ApplyClauseErrors property
ApplyCliches property
ApplyColloquialExpression property
ApplyCommonlyConfusedWords property
ApplyCommonMisspell property
ApplyComplexWords property
ApplyConfusedEasy property
ApplyConfusedEnglish property
ApplyConfusedHard property
ApplyConfusedMedium property
ApplyConfusedVerb property
ApplyConsecutiveNouns property
ApplyContractions property
ApplyDerogatory property
ApplyDifferentPrep property
ApplyDoubleNegative property
ApplyDoublePlural property
ApplyDoubleWordCheck property
ApplyElision property
ApplyEnglishDerived property
ApplyEnglishWords property
ApplyExotic property
ApplyExtraPrepositionCheck property
ApplyFalseFriend property
ApplyFemaleOccupation property
ApplyFixedExpression property
ApplyForeignWord property
ApplyFormalTerms property
ApplyFormatErrors property
ApplyGallicisms property
ApplyGenderExpressions property
ApplyGermanisms property
ApplyHomoGraphs property
ApplyHomonyms property
ApplyHomonymsEasy property
ApplyHomonymsHard property
ApplyHomoPhone1 property

[ApplyHomoPhone2 property](#)
[ApplyHomoPhones property](#)
[ApplyIncorrectPlural property](#)
[ApplyInformalExpressions property](#)
[ApplyJargonWords property](#)
[ApplykSplitInfinitives property](#)
[ApplyLowercaseAdjective property](#)
[ApplyLowercaseColor property](#)
[ApplyLowercaseNumbers property](#)
[ApplyLowercasePhrases property](#)
[ApplyLowercasePronouns property](#)
[ApplyMassVsCount property](#)
[ApplyMisspelledExpressions property](#)
[ApplyMisspelledForeignExpressions property](#)
[ApplyMisspelledItalian property](#)
[ApplyMisspelledWords property](#)
[ApplyMisusedWords property](#)
[ApplyNonStandardExpression property](#)
[ApplyNonStandardModifiers property](#)
[ApplyNoudModifierOrderCheck property](#)
[ApplyNounConsistency property](#)
[ApplyNounPhraseAgree property](#)
[ApplyNSAdjective property](#)
[ApplyNSClause property](#)
[ApplyNSCompare property](#)
[ApplyNSContract property](#)
[ApplyNSGeography property](#)
[ApplyNSInflection property](#)
[ApplyNSNegation property](#)
[ApplyNSPrep property](#)
[ApplyNSPronoun property](#)
[ApplyNSSpell property](#)
[ApplyNSUsage property](#)
[ApplyNSVerbForm property](#)
[ApplyOpenClosedSpelling property](#)
[ApplyOpenUsage property](#)
[ApplyOverusedPhrases property](#)
[ApplyPassiveVerbErrors property](#)
[ApplyPostClitAgree property](#)
[ApplyPrepExpression property](#)
[ApplyPrepositionalPhrases property](#)
[ApplyPretentiousWords property](#)
[ApplyPronounErrors property](#)
[ApplyPunctuationErrors property](#)
[ApplyRedundantExpressions property](#)
[ApplyRegionalExpression property](#)
[ApplyRelatedWord property](#)
[ApplySensitiveExp property](#)
[ApplySexistExpressions property](#)
[ApplySpellStandard property](#)
[ApplyStockPhrase property](#)
[ApplyStyleParameters property](#)
[ApplySubjectVerbAgreement property](#)
[ApplySwedishGender property](#)
[ApplySwedishNegation property](#)
[ApplySwedishUsage property](#)
[ApplyTrite property](#)
[ApplyTwoGender property](#)

[ApplyTypicalMisspell property](#)
[ApplyUnGrammaticalExpressions property](#)
[ApplyVagueQuantifiers property](#)
[ApplyVerbGroupConsistency property](#)
[ApplyWeakModifiers property](#)
[ApplyWordChoice property](#)
[ApplyWordCompoundingCheck property](#)
[ApplyWordConfusion property](#)
[ApplyWordForm property](#)
[ApplyWordGender property](#)
[ApplyWordParts property](#)
[ApplyWordyPhraseCheck property](#)
[AppViewPrefs property](#)
[AppViewPrefs class](#)
[AreDisabledIconsGrayed property](#)
[Ascent property](#)
[AsciiCodePage property](#)
[AtBeginning property](#)
[AtBeginningOfLine property](#)
[AtBeginningOfObject property](#)
[AtBeginningOfParagraph property](#)
[AtBeginningOfStream property](#)
[AtBeginningOfWord property](#)
[AtEnd property](#)
[AtEndOfLine property](#)
[AtEndOfObject property](#)
[AtEndOfParagraph property](#)
[AtEndOfStream property](#)
[AtEndOfWord property](#)
[Attributes property](#)
[Attributes class](#)
[AttrStyleName property](#)
[AuthorName property](#)
[AutoBackup property](#)
[AutoHyphenate property](#)
[AutomaticLink property](#)
[AutoRunMacro property](#)
[AutoRunMacro class](#)
[AutoSave property](#)
[AutoSaveMinutes property](#)
[AutoVersion property](#)

B

[BackColor property](#)
[BackColorIndex property](#)
[Background property](#)
[Background class](#)
[BackgroundPrintingOn property](#)
[BackgroundSpellingOn property](#)
[Backspace method](#)
[BackupPath property](#)
[BackupPaths property](#)
[Backward method](#)
[Bag class](#)
[BagCollection class](#)
[Bags property](#)
[BaseCollection class](#)

[BaseContainer class](#)
[BaseLineOffset property](#)
[BaseObject class](#)
[BaseTable property](#)
[BaseTable class](#)
[BaseTables property](#)
[BaseURL property](#)
[BeginChange method](#)
[BeginCustomLines method](#)
[BetweenLines class](#)
[BetweenLines property](#)
[BinName property](#)
[BinNameFromNumber method](#)
[BinNames property](#)
[BlockPaint property](#)
[Blue property](#)
[BodyOnly property](#)
[Bold property](#)
[Bold method](#)
[BookletPrinting property](#)
[Bookmark class](#)
[BookmarkCollection class](#)
[BookmarkManager property](#)
[BookmarkManager class](#)
[Bookmarks property](#)
[BookmarksByMarkerName property](#)
[Border class](#)
[BorderLines class](#)
[BorderLines property](#)
[BorderOffset property](#)
[BorderStyleName property](#)
[BottomBorder property](#)
[BottomExternalMargin property](#)
[BreakLink method](#)
[Breaks class](#)
[Breaks property](#)
[BreaksStyleName property](#)
[BringFrameToFront method](#)
[BringFrameToFrontOne method](#)
[Bullet class](#)
[Bullet property](#)
[BulletFonts property](#)
[BulletStyleName property](#)

C

[CalculateSmartLevels method](#)
[CanCreatePreviewBitmap property](#)
[CanEditProperty property](#)
[CanEmbed property](#)
[CanRepeatClickHere method](#)
[CanWePrint property](#)
[Caption property](#)
[Cascade method](#)
[CascadeWindow method](#)
[Case property](#)
[Cell property](#)
[CellCollection class](#)

[CellContainer class](#)
[CellEngine property](#)
[CellEngine class](#)
[CellEngines property](#)
[CellGroupLayout class](#)
[CellLayout method](#)
[CellLayout class](#)
[CellLayout property](#)
[CellLayoutCollection class](#)
[CellLayouts property](#)
[CellLayoutStyles property](#)
[CellRevert method](#)
[CellStyleName property](#)
[Center property](#)
[Changed property](#)
[ChangeKeyboardToLanguage property](#)
[ChangeSmartMaster method](#)
[ChangeTextToMatchkeyboard property](#)
[CharacterBorder class](#)
[CharacterBorder property](#)
[CharacterBorderName property](#)
[CharacterSet class](#)
[CharacterSet property](#)
[CharacterSetName property](#)
[CharacterStyle class](#)
[CharacterStyle property](#)
[CharacterStyleCollection class](#)
[CharacterStyleName property](#)
[CharacterStyles property](#)
[CharSet property](#)
[Checked property](#)
[CheckFieldEntries method](#)
[CheckForScriptEnumError property](#)
[CheckForScriptPropertyError property](#)
[ChgLineStyle method](#)
[ChildLayouts property](#)
[CityState property](#)
[ClassId property](#)
[ClassName property](#)
[CleanScreenMode property](#)
[CleanScreenOnStartUp property](#)
[Clear method](#)
[ClearAll method](#)
[ClearDivisionList method](#)
[ClearParaRevisionTags method](#)
[ClearSpellHints method](#)
[ClearSplits method](#)
[ClearTempFoundry method](#)
[ClearUpdate method](#)
[ClickHere class](#)
[ClickHereCollection class](#)
[ClickHerePrompts property](#)
[ClickHeres property](#)
[ClientHeight property](#)
[ClientWidth property](#)
[ClientWndHeight property](#)
[ClientWndWidth property](#)
[Close method](#)

[CloseAll method](#)
[CloseDocMacroName property](#)
[CloseDocWindow method](#)
[CloseMergeDataFile method](#)
[CloseObject method](#)
[CodePage property](#)
[Collapsible property](#)
[Collate property](#)
[Color property](#)
[Color class](#)
[ColorOverride property](#)
[ColumnBalance property](#)
[ColumnGap property](#)
[ColumnGroupLayout class](#)
[ColumnLayouts property](#)
[ColumnNumber property](#)
[ColumnWidth property](#)
[CombineDivisions method](#)
[CombineSections method](#)
[Company property](#)
[CompareFiles method](#)
[Condition property](#)
[ConditionType property](#)
[Configure method](#)
[Connect method](#)
[ConnectCells method](#)
[ConnectContainer method](#)
[ConnectedLayout class](#)
[ConnectedLayoutCollection class](#)
[ConnectedLayouts property](#)
[ConnectRows method](#)
[ConnectSectionTabs method](#)
[ConsistencyCheck method](#)
[Container property](#)
[Content class](#)
[Content property](#)
[ContentCollection class](#)
[ContentHeight property](#)
[ContentName property](#)
[Contents property](#)
[ContentStyleName property](#)
[ContentType property](#)
[ContentWidth property](#)
[ContextLocked property](#)
[ContextMenuOptions property](#)
[ContextMenuOptions class](#)
[ContinuedFromAlignment property](#)
[ContinuedFromMessage property](#)
[ContinuedFromStory property](#)
[ContinuedOnAlignment property](#)
[ContinuedOnMessage property](#)
[ContinuedOnStory property](#)
[Contract method](#)
[ContractOutlineLevel method](#)
[ConvertOnNew property](#)
[ConvertTo method](#)
[ConvertToClass method](#)
[Copies property](#)

[Copy method](#)
[CopyDivision method](#)
[CopyItem method](#)
[CopyMeaning method](#)
[CopySelection method](#)
[Count property](#)
[CountBlankLines property](#)
[Create method](#)
[Created event](#)
[CreateDataFile method](#)
[CreateDivision method](#)
[CreateDropCap method](#)
[CreateEmptyList method](#)
[CreateExternalDivision method](#)
[CreateFrame method](#)
[CreateFromBitmap method](#)
[CreateFromClipBrd method](#)
[CreateFromDataObject method](#)
[CreateFromMetafile method](#)
[CreateGlossary method](#)
[CreateGlossaryEntry method](#)
[CreateGraphic method](#)
[CreateLayer method](#)
[CreateNew method](#)
[CreateNewButton method](#)
[CreateOleEmbeddedFile method](#)
[CreateOleFromFile method](#)
[CreateOleLinkedFile method](#)
[CreateOleNew method](#)
[CreateParallelColumns method](#)
[CreatePreviewBitmap property](#)
[CreateRemark method](#)
[CreateTable method](#)
[CreateTextEntryField method](#)
[CreateVersion method](#)
[CreationDateString property](#)
[CreationTimeString property](#)
[CreationTimeValue property](#)
[Crop property](#)
[CurrentCell property](#)
[CurrentColumn property](#)
[CurrentEditor property](#)
[CurrentLanguage property](#)
[CurrentlyPrintingInBackground property](#)
[CurrentRow property](#)
[CurrentRunningScriptName property](#)
[CurrentRunningScriptPath property](#)
[CurrentVersion property](#)
[CustomLength property](#)
[CustomViewLevel property](#)
[CutDivision method](#)
[CutSelection method](#)

D

[DarkMode method](#)
[DataFileFieldNames property](#)
[DataFileName property](#)

[DataFormat property](#)
[DataNames property](#)
[DataObjectGetData method](#)
[DataObjectGetDataHere method](#)
[DateCreatedValue property](#)
[DateRevisedValue property](#)
[DbfUnderline method](#)
[DdeEnabled property](#)
[DdeLink class](#)
[DdeLinkCollection class](#)
[DdeLinkManager property](#)
[DdeLinkManager class](#)
[DdeLinks property](#)
[DdeOutboundInfo property](#)
[DebugVariable property](#)
[DefaultBinName property](#)
[DefaultCellStyleDescription property](#)
[DefaultColumnName property](#)
[DefaultDropCapStyleDescription property](#)
[DefaultFilePath property](#)
[DefaultFileType property](#)
[DefaultFrameStyleDescription property](#)
[DefaultLatinFont property](#)
[DefaultLeftColumnName property](#)
[DefaultMenu property](#)
[DefaultNewCategory property](#)
[DefaultNonLatinFont property](#)
[DefaultPageHeight property](#)
[DefaultPageStyleDescription property](#)
[DefaultPageWidth property](#)
[DefaultPitch property](#)
[DefaultRightColumnName property](#)
[DefaultTableStyleDescription property](#)
[DefaultTextStyleDescription property](#)
[DefCellStyleName property](#)
[DefColWidth property](#)
[Definition property](#)
[DefRowHeight property](#)
[DelayedGreeting property](#)
[DeleteBag method](#)
[DeleteButton method](#)
[DeleteChars method](#)
[DeleteClickHere method](#)
[DeleteContainer method](#)
[DeleteContent method](#)
[DeleteContents method](#)
[DeleteDivision method](#)
[DeleteField method](#)
[DeleteFont property](#)
[DeleteItem method](#)
[DeleteItemByPosition method](#)
[DeleteKey method](#)
[DeleteLayout method](#)
[DeleteMacroAccelerator method](#)
[DeleteMarker method](#)
[DeleteParallelColumns method](#)
[DeleteSection method](#)
[DeleteSmartCorrect method](#)

[DeleteTab method](#)
[DeleteTable method](#)
[DeleteVersion method](#)
[DemandLoad property](#)
[Demote method](#)
[DemoteOutlineLevel method](#)
[Descent property](#)
[Description property](#)
[DescriptionFileName property](#)
[Deselect method](#)
[Destroy method](#)
[DestroyDocWindow method](#)
[DestroyPowerFields method](#)
[DiagonalLines property](#)
[DirectionDown property](#)
[DirectionLeft property](#)
[DirectionRight property](#)
[DirectionUp property](#)
[DisableClickHereAction property](#)
[DisableClickHeres property](#)
[DisableConsistencyCheck property](#)
[DisableExportToNotes property](#)
[DisableHarmlessMessages property](#)
[DisableVersionReview property](#)
[Disconnect method](#)
[DisconnectCells method](#)
[DisplayablePageNum property](#)
[DisplayAsIcon property](#)
[DistanceFromLeftMargin property](#)
[Division property](#)
[Division class](#)
[DivisionCollection class](#)
[DivisionInfo property](#)
[DivisionInfo class](#)
[DivisionName property](#)
[DivisionNames property](#)
[DivisionOptions property](#)
[DivisionOptions class](#)
[Divisions property](#)
[DocControl property](#)
[DocControl class](#)
[DocControlPassword property](#)
[DocControlProtection property](#)
[DocControlRestrictedToEditor property](#)
[DocInfo property](#)
[DocInfo class](#)
[DocInfoField class](#)
[DocInfoFieldCollection class](#)
[DocInfoFieldManager class](#)
[DocOptions property](#)
[DocPath property](#)
[DocSize property](#)
[Document class](#)
[Document property](#)
[DocumentClose event](#)
[DocumentClosed event](#)
[DocumentCreate event](#)
[DocumentCreated event](#)

[DocumentExport event](#)
[DocumentExported event](#)
[DocumentImport event](#)
[DocumentImported event](#)
[DocumentInsert event](#)
[DocumentInserted event](#)
[DocumentLevel property](#)
[DocumentOpen event](#)
[DocumentOpened event](#)
[DocumentPaths property](#)
[DocumentPrint event](#)
[DocumentPrinted event](#)
[Documents class](#)
[Documents property](#)
[DocumentSave event](#)
[DocumentSaveAs event](#)
[DocumentSaved event](#)
[DocumentSavedAs event](#)
[DocVersionID property](#)
[DocWindow class](#)
[DocWindowCollection class](#)
[DocWindows property](#)
[DoesMarkerNameMatch method](#)
[DoesMultiCellPaste method](#)
[DoInitialCaps property](#)
[DoneWithRightMouseMenu method](#)
[DoSentenceInitialCaps property](#)
[DoSmartLinks property](#)
[DoSmartQuotes property](#)
[DoubleUnderline property](#)
[DoVerb method](#)
[DownloadGraphics property](#)
[DragDropOn property](#)
[DriverName property](#)
[DropCapContainer class](#)
[DropCapLayout class](#)
[DropCapLayoutCollection class](#)
[DropCapPosition property](#)
[DropCaps property](#)
[DropCapStyleName property](#)
[DropCapStyles property](#)

E

[Editable property](#)
[EditHyperLink method](#)
[EditLinkInfo method](#)
[Editor class](#)
[EditorCollection class](#)
[EditorInitials property](#)
[EditorManager property](#)
[EditorManager class](#)
[EditorName property](#)
[Editors property](#)
[EditorVerificationType property](#)
[EditWrap method](#)
[EffectiveColumnWidth property](#)
[EMail property](#)

[Embed method](#)
[Embedded property](#)
[EmbedFonts property](#)
[EmitsEvents property](#)
[Enabled property](#)
[EndChange method](#)
[EndCustomLines method](#)
[Ending method](#)
[EndingColOfSelection property](#)
[EndingRowOfSelection property](#)
[EndnoteDivisionGroupNum property](#)
[EndnoteDivisionGroupNum class](#)
[EndnoteDivisionNum property](#)
[EndnoteDivisionNum class](#)
[EndnoteDocNum property](#)
[EndnoteDocNum class](#)
[EndnoteLayout class](#)
[EndnoteLayoutCollection class](#)
[Endnotes property](#)
[EndPrinting method](#)
[EnterClickHere event](#)
[EnterLayout event](#)
[EntireDocument property](#)
[EnumerateChartLinks method](#)
[EnumerateTerm method](#)
[Enumeration property](#)
[EnvelopeBarCode method](#)
[EnvelopePrint method](#)
[Epoch property](#)
[EqnFontHeight property](#)
[ExchangeItem method](#)
[ExitClickHere event](#)
[Expand method](#)
[Expandable property](#)
[ExpandOutline method](#)
[ExpandOutlineLevel method](#)
[ExportAllAsNotesFX method](#)
[ExportAsNotesFX method](#)
[ExportToNotesFX property](#)
[ExternallyControlledUndo property](#)
[ExternalName property](#)
[ExternalType property](#)
[ExtractText method](#)

F

[FaceName property](#)
[FaceNames property](#)
[FaceStyleName property](#)
[FaxNumber property](#)
[FieldDelimiter property](#)
[FieldDelimiterText property](#)
[FieldManager property](#)
[FieldNumber property](#)
[Fields property](#)
[FieldType property](#)
[FileFormat property](#)
[FilePassword property](#)

[FileProtectionType property](#)
[FilesToCompare property](#)
[FileType property](#)
[FillerPageText property](#)
[FillStyle property](#)
[Filter property](#)
[Filter class](#)
[FilterHelper property](#)
[FilterHelper class](#)
[FilterName property](#)
[Find method](#)
[FindAndReplace property](#)
[FindAndReplace class](#)
[FindCellLayout method](#)
[FindClass method](#)
[FindExactCase property](#)
[FindFont property](#)
[FindForwardDirection property](#)
[FindString property](#)
[FindStyleName property](#)
[FindTerm method](#)
[FindWithProperties property](#)
[FinishedSpellChecking property](#)
[First property](#)
[FirstChild property](#)
[FirstCursorableDivision property](#)
[FirstDivision property](#)
[FirstName property](#)
[FirstPage property](#)
[FitType property](#)
[FixAcronymns property](#)
[FixBullets property](#)
[Font class](#)
[Font property](#)
[FontColor property](#)
[FontFaceSupport property](#)
[FontMatching property](#)
[FontMetrics property](#)
[FontMetrics class](#)
[FontName property](#)
[FontStyleName property](#)
[FontUnitName property](#)
[FontUnits property](#)
[Footer property](#)
[FooterLayout class](#)
[FooterLayoutCollection class](#)
[Footers property](#)
[FooterStyles property](#)
[Footnote class](#)
[FootnoteCollection class](#)
[FootnoteContSep property](#)
[FootnoteContSep class](#)
[FootnoteLayout class](#)
[FootnoteLayoutCollection class](#)
[FootnoteLayouts property](#)
[FootnoteNumbering property](#)
[FootnoteNumbering class](#)
[FootnoteNumOpt class](#)

[FootnoteOptions property](#)
[FootnoteOptions class](#)
[Footnotes property](#)
[FootnoteSeparator property](#)
[FootnoteSeparator class](#)
[FootnoteSepOpt class](#)
[FootnoteTable class](#)
[ForceDocToLoad method](#)
[ForeColorIndex property](#)
[FormatCheckPref class](#)
[FormatCheckPreferences property](#)
[FormatType property](#)
[Formula property](#)
[Forward method](#)
[Foundry property](#)
[Foundry class](#)
[Frame property](#)
[FrameCaptionOptions class](#)
[FrameCaptionOptions property](#)
[FrameContainer class](#)
[FrameGroupLayout class](#)
[FrameLayout class](#)
[FrameLayoutCollection class](#)
[FrameRevert method](#)
[Frames property](#)
[FrameStyleName property](#)
[FrameStyles property](#)
[FreeMenus property](#)
[FullName property](#)
[FXGetNotesString method](#)
[FXGetNotesWriteHandle method](#)
[FXSetNotesString method](#)

G

[GapBetweenPanels property](#)
[Get method](#)
[GetActiveList method](#)
[GetAfidHelpInfo property](#)
[GetArrayProp method](#)
[GetAspectMetafilePict method](#)
[GetButtonId property](#)
[GetButtonType property](#)
[GetContents method](#)
[GetCopyFormatCategories method](#)
[GetCount method](#)
[GetCurrentMarkerName method](#)
[GetData method](#)
[GetDisplayableFilterName method](#)
[GetEnumerator method](#)
[GetEnvelopeDefaults method](#)
[GetFileDescription method](#)
[GetFilterExtension property](#)
[GetFilterExtForDialogBox property](#)
[GetFilterId property](#)
[GetFormatName property](#)
[GetFormula method](#)
[GetHomeDirectory property](#)

[GetInternetFile method](#)
[GetItemName property](#)
[GetLastUsedFilter method](#)
[GetLineMix method](#)
[GetLineStyle method](#)
[GetLinkDisplayNameFileLength method](#)
[GetLinkName method](#)
[GetLinkSourceName method](#)
[GetListName method](#)
[GetMarkedText method](#)
[GetMarkerName method](#)
[GetMisspelledWord method](#)
[GetNamedProperty method](#)
[GetNameFromPage method](#)
[GetObjectList method](#)
[GetPageNumFromDisplayablePageNum method](#)
[GetParagraphNumber method](#)
[GetParagraphNumberString method](#)
[GetParaNumber method](#)
[GetPasteFormatCategories method](#)
[GetPosition method](#)
[GetPrinterInfo method](#)
[GetProfileString method](#)
[GetRedoWhatDesc property](#)
[GetRGB method](#)
[GetRightContextMenu method](#)
[GetServerName property](#)
[GetSource method](#)
[GetSpellStatus method](#)
[GetSpellUserDictStatus method](#)
[GetStandardButtonId method](#)
[GetStatus property](#)
[GetString method](#)
[GetTabAlignChar method](#)
[GetTabLeaderType method](#)
[GetTabPosition method](#)
[GetTabRelativeType method](#)
[GetTabType method](#)
[GetText method](#)
[GetTopicName property](#)
[GetUndoWhatDesc property](#)
[GetUniqueName method](#)
[GetUserClassNameFull method](#)
[GetUserClassNameShort method](#)
[GetValue property](#)
[GetWordMisspelled method](#)
[GIFAnimationOn property](#)
[Glossarize method](#)
[Glossary class](#)
[GlossaryCollection class](#)
[GlossaryDataFileName property](#)
[GlossaryDataFiles property](#)
[GlossaryDataPaths property](#)
[GlossaryInsert method](#)
[GlossaryOpen method](#)
[GlossaryPath property](#)
[Glossarys property](#)
[GotFocus event](#)

[GoTo method](#)
[GoToBookmark method](#)
[GoToClickHere method](#)
[GoToContainer method](#)
[GoToDivision method](#)
[GoToLayout method](#)
[GotoNextParallelColumn method](#)
[GoToObject method](#)
[GoToPage method](#)
[GoToSection method](#)
[GoToTableCell method](#)
[Grammar class](#)
[GrammarFormalityLevel property](#)
[GrammarOptions property](#)
[GrammarProofLevel property](#)
[GrammarSetName property](#)
[Graphic class](#)
[Graphic property](#)
[GraphicCollection class](#)
[GraphicExports property](#)
[GraphicImports property](#)
[GraphicOleObject class](#)
[GraphicOleObject property](#)
[GraphicOleObjectCollection class](#)
[GraphicPath property](#)
[GraphicPaths property](#)
[Graphics property](#)
[GraphicType property](#)
[Green property](#)
[Greeting property](#)
[GridDistance property](#)
[GridType property](#)
[GroupDivision method](#)
[GroupLayout class](#)
[GroupLayoutCollection class](#)
[Groups property](#)
[Gutter property](#)
[Gutter class](#)

H

[HandsOffStorage method](#)
[Hang property](#)
[HasContents property](#)
[HasFocus property](#)
[HasIndex property](#)
[HasLocalTabs property](#)
[HasNamedProperty method](#)
[HasTabs property](#)
[HasTOC property](#)
[Header property](#)
[HeaderLayout class](#)
[HeaderLayoutCollection class](#)
[Headers property](#)
[HeaderStyles property](#)
[Heading property](#)
[Height property](#)
[Help method](#)

[HelpText property](#)
[HiddenMode property](#)
[Hide method](#)
[HideCaretAndSelection method](#)
[HideFormula property](#)
[HideIconBar method](#)
[HideIconBars method](#)
[HideOutlineLevels property](#)
[HideStatusBar method](#)
[HighLightMode property](#)
[HighlightToggle method](#)
[HiLiteColor property](#)
[Hit method](#)
[HorizontalSplitWindow property](#)
[HorizScrollBarVisible property](#)
[HorzRuler property](#)
[HourGlass method](#)
[HTMLOptions class](#)
[HTMLOptions property](#)
[Hwnd property](#)
[HyphenateLastWordInColumnOrPage property](#)
[HyphenateLastWordInPara property](#)
[HyphenationOptions property](#)
[HyphenationOptions class](#)
[HyphZoneAfter property](#)
[HyphZoneBefore property](#)

I

[IconBar class](#)
[IconBarCollection class](#)
[IconBarManager class](#)
[IconBarManager property](#)
[IconBarPositionState property](#)
[IconBars property](#)
[IconBarSets property](#)
[IconHelpText property](#)
[IconPath property](#)
[IconPaths property](#)
[IconScript property](#)
[IconSetName property](#)
[IconSize property](#)
[ID property](#)
[IDispatch property](#)
[IgnoreSoftHyphens property](#)
[IgnoreTab property](#)
[ImageProcess method](#)
[ImportAsSource property](#)
[ImportGraphic method](#)
[ImportInsert event](#)
[ImportInserted event](#)
[ImportPicture method](#)
[ImportWatermarkGraphic method](#)
[IncludeFooter property](#)
[IncludeHeader property](#)
[IncludeInitialsInNotes property](#)
[IncludeList property](#)
[IncludeURL property](#)

[Indent class](#)
[Indent property](#)
[IndentFromLeft property](#)
[IndentStyleName property](#)
[IndexAll method](#)
[IndexAlphabeticSeparator property](#)
[IndexDivision property](#)
[IndexIndentType property](#)
[IndexParent property](#)
[IndexPrimaryStyleName property](#)
[IndexRange property](#)
[IndexSecondaryStyleName property](#)
[IndexSection class](#)
[IndexSection property](#)
[IndexSeparatorStyleName property](#)
[IndexSource property](#)
[InDocument property](#)
[InfoBoxSelectionText property](#)
[InitializeRoute property](#)
[InitFindAndReplace method](#)
[Initialize property](#)
[InitialParaStyle property](#)
[InitialsForFilters property](#)
[InsertBreak method](#)
[InsertBullet method](#)
[InsertClickHere method](#)
[InsertColumnBreak method](#)
[InsertDate method](#)
[InsertDocInfo method](#)
[InsertDocument method](#)
[InsertField method](#)
[InsertFont property](#)
[InsertFootnote method](#)
[InsertHardSpace method](#)
[InsertHyperLink method](#)
[InsertIndex method](#)
[InsertionMode property](#)
[InsertLink method](#)
[InsertMarker method](#)
[InsertNote method](#)
[InsertNumber method](#)
[InsertOleDivision method](#)
[InsertOne method](#)
[InsertPageBreak method](#)
[InsertPageLayout method](#)
[InsertPageNumber method](#)
[InsertPath method](#)
[InsertRowOrColumn method](#)
[InsertRuby method](#)
[InsertSection method](#)
[InsertSpecialTab method](#)
[InsertTab method](#)
[InsertText method](#)
[InsertTOC method](#)
[Interactive property](#)
[InternalCopy method](#)
[InternalCut method](#)
[InternalPaste method](#)

[InternetExtraFile method](#)
[InternetFileExists method](#)
[Interval property](#)
[Invalidate method](#)
[InvalidateButton method](#)
[InvalidateWholeBar method](#)
[IsActionOnButtonDown property](#)
[IsActive property](#)
[IsAmikake property](#)
[IsAmiProTableImport property](#)
[IsAnnotateOnly property](#)
[IsAsciiCRLF property](#)
[IsAsciiKeepStyle property](#)
[IsAutoGrow property](#)
[IsBadReference property](#)
[IsBorder property](#)
[IsBothSidesEqual property](#)
[IsBreakable property](#)
[IsBubbleHelp property](#)
[IsCaseExact method](#)
[IsCellMenuEnabled property](#)
[IsChanged property](#)
[IsChangedOtherThanLinkTo property](#)
[IsChangedSinceTimeSave property](#)
[IsChangedToLinkTo property](#)
[IsChartLink property](#)
[IsChildSpannable property](#)
[IsCollapsed property](#)
[IsCollapsible property](#)
[IsColumnBreakable property](#)
[IsColumnBreakAfter property](#)
[IsColumnBreakBefore property](#)
[IsComplex property](#)
[IsConnected property](#)
[IsContinuedFrom property](#)
[IsContinuedOn property](#)
[IsCumulative property](#)
[IsDataNameUsed method](#)
[IsDebug property](#)
[IsDisableWarningMessages property](#)
[IsDisplayMisspelled property](#)
[IsDivisionExternal property](#)
[IsDocLoading property](#)
[IsDocObject property](#)
[IsDocumentInRoute property](#)
[IsDoubleWordError property](#)
[IsDraw property](#)
[IsEmpty property](#)
[IsEmpty method](#)
[IsEmptyDoc property](#)
[IsEnabled property](#)
[IsEndnoteDivision property](#)
[IsEquation property](#)
[IsErrorChecking property](#)
[IsExpandDown property](#)
[IsExpandLeft property](#)
[IsExpandRight property](#)
[IsExpandUp property](#)

[IsExportedAsNotesFX method](#)
[IsExportedToNotesFX property](#)
[IsExternalFile property](#)
[IsFilterTypePresent method](#)
[IsFixedLength property](#)
[IsFooter property](#)
[IsFrameCaptionOptions property](#)
[IsFrameMenuEnabled property](#)
[IsGotoable property](#)
[IsGrabBar property](#)
[IsGrammarError property](#)
[IsGraphicalObject property](#)
[IsGraphicMenuEnabled property](#)
[IsHangover property](#)
[IsHeader property](#)
[IsHiddenMark property](#)
[IsHideHeaderFooter property](#)
[IsHighlightNote property](#)
[IsHorizontalScrollBarCleanScrn property](#)
[IsIconDepressible property](#)
[IsInBulletEditMode property](#)
[IsIndex property](#)
[IsIndexMarkAble property](#)
[IsInDraft property](#)
[IsInOleDivision property](#)
[IsInOutline property](#)
[IsInPageSort property](#)
[IsLastStop property](#)
[IsLesser property](#)
[IsLocal property](#)
[IsLocked property](#)
[IsLockedForRevisions property](#)
[IsLotusChart property](#)
[IsMarginSameAsParent property](#)
[IsMarginsInColor property](#)
[IsMarkerEqualToSelection method](#)
[IsMarkerValid property](#)
[IsMisspelled property](#)
[IsNoPrintZone property](#)
[IsNotCopyable property](#)
[IsNotCopyImage property](#)
[IsNotGroupable property](#)
[IsNoUICommAllowed property](#)
[IsOpen property](#)
[IsOverridden property](#)
[IsOverride property](#)
[IsPageBreak property](#)
[IsPageBreakAfter property](#)
[IsPageBreakBefore property](#)
[IsPageBreakMarks property](#)
[IsPageBreakWithin property](#)
[IsParagraphNumberingDown property](#)
[IsParagraphParent property](#)
[IsParallelColumnsMenuEnabled property](#)
[IsPartOfGroup property](#)
[IsPersistent property](#)
[IsPointWithin method](#)
[IsPrePrintedForm property](#)

[IsPrintable property](#)
[IsPrintClickHereBlocks property](#)
[IsPrivate property](#)
[IsPrompting property](#)
[IsProtected property](#)
[IsRegistered property](#)
[IsRepeat property](#)
[IsReplaceable property](#)
[IsResetParagraphNumber property](#)
[IsRevisionMark property](#)
[IsSame property](#)
[IsSave property](#)
[IsScalable property](#)
[IsScalableBorder property](#)
[IsScrollable property](#)
[IsShowing property](#)
[IsSingleClickEntry property](#)
[IsSizable property](#)
[IsSizingViaMouse property](#)
[IsSmartCorrectEnabled property](#)
[IsSmartEditEnabled property](#)
[IsSmartShadeEnabled property](#)
[IsSnapTo property](#)
[IsSortFromEnd property](#)
[IsSpellBarUp property](#)
[IsSpellMode property](#)
[IsSqueeze property](#)
[IsStyle property](#)
[IsStyleSheet property](#)
[IsSymbolic property](#)
[IsTableHeading property](#)
[IsTemp property](#)
[IsTextLocked property](#)
[IsTextMenuEnabled property](#)
[IsTOC property](#)
[IsTOCMarkAble property](#)
[IsTrueType property](#)
[IsUndoOn property](#)
[IsUpdateAutomatic property](#)
[IsUpdateIndex property](#)
[IsUpdateTOC property](#)
[IsValid property](#)
[IsVerticalScrollBarCleanScrn property](#)
[IsViewAnchor property](#)
[IsViewBookmarks property](#)
[IsViewClickHereBlocks property](#)
[IsViewColGuides property](#)
[IsViewColumnBreakMarks property](#)
[IsViewDDELinks property](#)
[IsViewHorzRuler property](#)
[IsViewHorzScrollBar property](#)
[IsViewMenuCleanScrn property](#)
[IsViewNotes property](#)
[IsViewOutlineFlowToScreen property](#)
[IsViewOutlineIndent property](#)
[IsViewPageLayoutMarks property](#)
[IsViewParallelColumnBorder property](#)
[IsViewPictures property](#)

[IsViewReturnIconCleanScrn property](#)
[IsViewReturns property](#)
[IsViewRubyMarks property](#)
[IsViewRulerMarks property](#)
[IsViewSectionBreakMarks property](#)
[IsViewSectionTabs property](#)
[IsViewSmartIconsCleanScrn property](#)
[IsViewStatusBarCleanScrn property](#)
[IsViewStatusSpell property](#)
[IsViewTableCellSelection property](#)
[IsViewTableGridLines property](#)
[IsViewTableHeadings property](#)
[IsViewTabs property](#)
[IsViewThumbBarCleanScrn property](#)
[IsViewTitleBarCleanScrn property](#)
[IsViewVertRuler property](#)
[IsViewVertScrollBar property](#)
[IsWMCommandValid method](#)
[IsWordProChart property](#)
[IsWorkingDir property](#)
[Italic property](#)
[Italic method](#)
[Item method](#)
[Items property](#)

J

[Join class](#)
[Join property](#)
[JoinCorners property](#)
[JoinHeight property](#)
[JoinName property](#)
[JoinType property](#)
[JoinWidth property](#)
[Justifiable property](#)

K

[KeepWithNext property](#)
[KeepWithPrev property](#)
[KeyboardLanguage property](#)
[KeyStroke event](#)
[Keywords property](#)
[Kinsoku class](#)
[Kinsoku property](#)

L

[LabelCreate method](#)
[LabelMerge method](#)
[LandscapeMode property](#)
[Language property](#)
[Language class](#)
[Last property](#)
[LastChild property](#)
[LastCursorableDivision property](#)
[LastDivision property](#)
[LastDocPath property](#)

[LastEditorName property](#)
[LastGraphicPath property](#)
[LastGraphicType property](#)
[LastInsertSymbolFont property](#)
[LastMacroPath property](#)
[LastMessageID property](#)
[LastMessageStr property](#)
[LastMessageType property](#)
[LastName property](#)
[LastPage property](#)
[LastUsedDateFormula property](#)
[Layer property](#)
[LayerName property](#)
[Layout class](#)
[Layout property](#)
[LayoutCollection class](#)
[LayoutName property](#)
[LayoutOverride class](#)
[LayoutOverride property](#)
[Layouts property](#)
[LeaderDotDashChar property](#)
[LeaderDotDotChar property](#)
[LeaderDotType property](#)
[LeaderDotUnderscoreChar property](#)
[LeadingText property](#)
[LeastRecentVersion method](#)
[Left property](#)
[LeftBorder property](#)
[LeftExternalMargin property](#)
[LeftPage property](#)
[LeftTopCellId property](#)
[Length property](#)
[Level property](#)
[LineLocation property](#)
[LineMix property](#)
[LineNumberOptions property](#)
[LineNumberOptions class](#)
[LinePlacement property](#)
[Lines property](#)
[LinesSpacedEveryNthUnit property](#)
[LineValid property](#)
[Link method](#)
[LinkAvailable property](#)
[LinkContainers method](#)
[LinkDisplayName property](#)
[LinkDisplayNameFileLength property](#)
[Linked property](#)
[LinkedFileName property](#)
[LinkFrame property](#)
[LinkFrameContents method](#)
[LinkGraphic property](#)
[ListCount property](#)
[LoadFilesMaximized property](#)
[Localize method](#)
[Location property](#)
[Locked property](#)
[LockLevel property](#)
[LockResult property](#)

[Locks property](#)
[LostFocus event](#)
[LotusMiscStatus property](#)
[LowerCase property](#)
[LowerCase method](#)
[LowerCaseAscent property](#)
[LwpMenuBar property](#)
[LWPTimer class](#)

M

[Macro class](#)
[Macro property](#)
[MacroEndRecord method](#)
[MacroName property](#)
[MacroPath property](#)
[MacroPaths property](#)
[MacroPlay method](#)
[MacroRecord method](#)
[MacroStatus property](#)
[MailDocument method](#)
[MailRouting class](#)
[MailRouting property](#)
[MailRoutingPtr property](#)
[MaintainAspectRatio property](#)
[MaintainEditor property](#)
[MakeTableFromText method](#)
[MakeUniqueLinkName method](#)
[ManualFrame method](#)
[ManualLinkFrames method](#)
[ManualTable method](#)
[MarginBetween property](#)
[MarginBottom property](#)
[MarginColor property](#)
[MarginLeft property](#)
[MarginRight property](#)
[MarginTop property](#)
[Mark method](#)
[MarkCharacter property](#)
[Marker class](#)
[MarkerClass property](#)
[MarkerCollection class](#)
[MarkerName property](#)
[Markers property](#)
[MarkPosition property](#)
[MarkRevisionInsert method](#)
[MarkType property](#)
[Master property](#)
[MasterName property](#)
[MatchType property](#)
[MaxBottomBorder property](#)
[MaxBottomGutter property](#)
[MaxContentHeight property](#)
[MaxContentWidth property](#)
[MaxHorzPaneDistance property](#)
[MaxHyphLines property](#)
[MaxIdenticalConsecSentOpens property](#)
[MaxIdenticalSentOpensWithin10 property](#)

[Maximize method](#)
[MaximizeOnStartUp property](#)
[MaximumWordsinaSentence property](#)
[MaxLeftBorder property](#)
[MaxLeftGutter property](#)
[MaxNumColsAllowed property](#)
[MaxNumRowsAllowed property](#)
[MaxRightBorder property](#)
[MaxRightGutter property](#)
[MaxSplitCols property](#)
[MaxSplitRows property](#)
[MaxTopBorder property](#)
[MaxTopGutter property](#)
[MaxVertPaneDistance property](#)
[MenuItem class](#)
[MenuItemCollection class](#)
[MenuPath property](#)
[MenuPaths property](#)
[Merge method](#)
[MergeAddDataRecord method](#)
[MergeFileType property](#)
[MergeInfoPtr property](#)
[MergeOptions class](#)
[MergeOptions property](#)
[MergeSetDataFile method](#)
[MergeStart method](#)
[MergeStepNumber property](#)
[MergeToFile property](#)
[Messages method](#)
[MetafilePict property](#)
[MinBottomMargin property](#)
[MinHeight property](#)
[Minimize method](#)
[MinLeftMargin property](#)
[MinRightMargin property](#)
[MinTopMargin property](#)
[MirrorName property](#)
[MirrorPage method](#)
[ModifiedDateString property](#)
[ModifiedTimeString property](#)
[ModifiedTimeValue property](#)
[MorphSelectionToTable method](#)
[MostRecentVersion method](#)
[MouseDown event](#)
[MouseUp event](#)
[Move method](#)
[Moved event](#)
[MoveDivision method](#)
[MoveDown method](#)
[MoveItem method](#)
[MoveParagraph method](#)
[MoveToBack method](#)
[MoveToEnd method](#)
[MoveToFront method](#)
[MoveToStart method](#)
[MoveUp method](#)
[MultiCellPasteOn property](#)
[MultiCompareParaTag property](#)

[MultiCompareParaTagSet property](#)

N

[Name property](#)

[NameBasedOnStyle property](#)

[NameForFilters property](#)

[NameOfExternalBorder property](#)

[NavigationLink property](#)

[Negative property](#)

[NewDivision method](#)

[NewDocMacroName property](#)

[NewDocument method](#)

[NewFile property](#)

[NewFrame method](#)

[NewInfoBusLink method](#)

[NewItem method](#)

[NewItemByPosition method](#)

[NewVersion property](#)

[NewWindow method](#)

[Next method](#)

[NextClickHere property](#)

[NextCursorableDivision property](#)

[NextCycleAlign method](#)

[NextCycleAttribute method](#)

[NextCycleBullet method](#)

[NextCycleFont method](#)

[NextCycleFontSize method](#)

[NextCycleIndent method](#)

[NextCycleNumber method](#)

[NextCycleStyle method](#)

[NextName property](#)

[NextNeighbor property](#)

[NextSection property](#)

[NextStyleName property](#)

[NextText property](#)

[NextToObject method](#)

[NextVersion method](#)

[NoHyphenate property](#)

[NonUserDocument property](#)

[Normal property](#)

[NormalParagraph property](#)

[NormalText method](#)

[NoteContainer class](#)

[NoteLayout class](#)

[NoteLayoutCollection class](#)

[NoteLayouts property](#)

[NotesFlow property](#)

[Number property](#)

[NumberEveryNthLine property](#)

[NumberFound property](#)

[Numbering class](#)

[Numbering property](#)

[NumberingPosition property](#)

[NumberingStyleName property](#)

[NumberOfDataFields property](#)

[NumberOfLines property](#)

[NumberOfMergeConditions property](#)

[NumberOfReplacements property](#)
[NumberOfRevisions property](#)
[NumberOfVersions method](#)
[NumberSequenceName property](#)
[NumberWhichLines property](#)
[NumCharsInDoc property](#)
[NumCharsInParagraph property](#)
[NumCols property](#)
[NumColsSpannedOneCell property](#)
[NumContainers property](#)
[NumDecimalPlaces property](#)
[NumDisplayablePagesInDoc property](#)
[NumericFormat class](#)
[NumericFormat property](#)
[NumericFormatSubset class](#)
[NumFields property](#)
[NumLinesOfSpace property](#)
[NumLinesOfSpaceAbove property](#)
[NumLinesOfSpaceAboveLine property](#)
[NumLinesOfSpaceBelow property](#)
[NumOfRecentFiles property](#)
[NumPagesInDoc property](#)
[NumParagraphs property](#)
[NumRows property](#)
[NumRowsSpannedOneCell property](#)
[NumRowsThatFit property](#)
[NumTabs property](#)
[NumUndoLevels property](#)
[NumWindowsViewingDoc property](#)
[NumWordsInDoc property](#)

O

[Object property](#)
[ObjectType property](#)
[Oblique property](#)
[OCXDesignMode property](#)
[Ole1Object property](#)
[OleContainer property](#)
[OleContainerDocName property](#)
[OleControl class](#)
[OleEmbeddedSize property](#)
[OLEEnabled property](#)
[OleMinHeight property](#)
[OleObject class](#)
[OleObject property](#)
[OleObjectCollection class](#)
[OleObjects property](#)
[OleObjectSize property](#)
[Open method](#)
[OpenDataFile method](#)
[OpenDocMacroName property](#)
[OpenDocsVisible property](#)
[OpenDocument method](#)
[OpenDocumentFromNotes method](#)
[Opened event](#)
[OpenExisingFileInWelcomeBox property](#)
[OpenFromStorage method](#)

[OpenObject method](#)
[OpenReadOnly property](#)
[Options property](#)
[Options class](#)
[OrigFileType property](#)
[OrigHeight property](#)
[OrigWidth property](#)
[OStype property](#)
[Outline property](#)
[OutlineBorderLines property](#)
[OutlineButtons property](#)
[OutlineHeadingButtonsOnly property](#)
[OutlineLevel property](#)
[OutlineMoveTextDown method](#)
[OutlineMoveTextUp method](#)
[OutlineOnlyHeadingsWhenCollapsed property](#)
[OutlineSeqCollection class](#)
[OutlineSeqItemCollection class](#)
[OutlineSeqItems property](#)
[OutlineStyleSequence class](#)
[OutlineStyleSequence method](#)
[OutlineStyleSequences property](#)
[OutputToFile property](#)
[OutSeqItem class](#)
[Override property](#)
[OverrideCheckAndEnable property](#)
[Overstrike property](#)
[OverstrikeCharacter property](#)

P

[Page property](#)
[PageContainer class](#)
[PageDown method](#)
[PageLayout class](#)
[PageLayoutCollection class](#)
[PageNo property](#)
[PageNum property](#)
[PageNumber property](#)
[PageNumberAsText property](#)
[PageNumFirstPageShowing property](#)
[PageNumLastPageShowing property](#)
[PageOrder property](#)
[Pages property](#)
[PageStyleName property](#)
[PageStyles property](#)
[PageToUseLayoutOn property](#)
[PageUp method](#)
[PairKerning property](#)
[PaperName property](#)
[PaperNames property](#)
[ParagraphBackground class](#)
[ParagraphBackground property](#)
[ParagraphBorder property](#)
[ParagraphBorder class](#)
[ParagraphHasDropCap property](#)
[ParagraphHasText property](#)
[ParagraphStyle class](#)

[ParagraphStyle property](#)
[ParagraphStyleCollection class](#)
[ParagraphStyleName property](#)
[ParagraphStyles property](#)
[ParagraphSymbolChar property](#)
[ParallelColsCollection class](#)
[ParallelColsContainer class](#)
[ParallelColumns property](#)
[ParallelColumns class](#)
[Parent property](#)
[ParentMenuHWND property](#)
[ParentName property](#)
[Partial property](#)
[Paste method](#)
[PasteDivision method](#)
[PasteLink method](#)
[PasteSpecial method](#)
[Path property](#)
[PathName property](#)
[Pattern property](#)
[PColConnectCells method](#)
[PColConnectRows method](#)
[PColDisconnectCells method](#)
[PColSelectColumn method](#)
[PColSelectRow method](#)
[PColSelectTable method](#)
[Percentage property](#)
[PersonalData1 property](#)
[PersonalData2 property](#)
[PersonalData3 property](#)
[PersonalData4 property](#)
[PhoneNumber property](#)
[PitchAndFamily property](#)
[Placement property](#)
[Plain property](#)
[Play method](#)
[PointSize property](#)
[Position property](#)
[PositionType property](#)
[PositionXInContainer property](#)
[PositionXOnPage property](#)
[PositionYInContainer property](#)
[PositionYOnPage property](#)
[PowerField class](#)
[PowerField property](#)
[PowerFieldCollection class](#)
[PowerFields property](#)
[PreClose event](#)
[Preferences class](#)
[Preferences property](#)
[Prefix property](#)
[PrePrint event](#)
[Present method](#)
[Presentation property](#)
[Presentation class](#)
[Previous method](#)
[PreviousClickHere property](#)
[PreviousCursorableDivision property](#)

[PreviousName property](#)
[PreviousNeighbor property](#)
[PreviousVersion method](#)
[Print method](#)
[PrintDestination property](#)
[Printed event](#)
[PrinterName property](#)
[PrintGraphics property](#)
[PrintManager property](#)
[PrintManager class](#)
[PrintOut method](#)
[PrintPagesFrom property](#)
[PrintPagesTo property](#)
[PrintPageType property](#)
[PrintRange property](#)
[PrintSettings property](#)
[PrintSettings class](#)
[Private property](#)
[ProcessAccelKey method](#)
[ProductVersion property](#)
[ProgID property](#)
[Promote method](#)
[PromoteOutlineLevel method](#)
[Prompt property](#)
[PromptHidden property](#)
[ProtectedMode property](#)
[Purge method](#)

Q

[QueueName property](#)
[QuickAlignFrame method](#)
[QuickAlignTable method](#)
[Quit event](#)
[Quit method](#)

R

[Read method](#)
[ReadCompressed property](#)
[ReadOnly property](#)
[Recursive property](#)
[Red property](#)
[Redo method](#)
[Refresh method](#)
[RegisterWPDataSet method](#)
[Relative property](#)
[RelativeIndent property](#)
[RelativeIndent class](#)
[RelativePageNum property](#)
[RelativeType property](#)
[RelativeXDistance property](#)
[RelativeYDistance property](#)
[Release method](#)
[ReleaseNumber property](#)
[Remove method](#)
[RemoveBookmark method](#)
[RemoveChildFromLayout method](#)

[RemoveDataFile method](#)
[RemoveDdeLink method](#)
[RemoveDepOnDocFile method](#)
[RemoveDivision method](#)
[RemoveEditor method](#)
[RemoveHyperlink method](#)
[RemoveIndexEntry method](#)
[RemoveList method](#)
[RemoveNamedProperty method](#)
[RemovePersistentAccelerators method](#)
[RemoveTOCEntry method](#)
[Render method](#)
[RenderClipBitmap method](#)
[RenderClipDIB method](#)
[RenderClipMetafile method](#)
[RenderClipPalette method](#)
[RenderedPageNumber property](#)
[Repaint method](#)
[Replace method](#)
[ReplaceAll method](#)
[ReplaceAttributes property](#)
[ReplaceCmd method](#)
[ReplaceContents method](#)
[ReplaceExactCase property](#)
[ReplaceFont property](#)
[ReplaceLanguage property](#)
[Replacements property](#)
[ReplaceString property](#)
[ReplaceStyleName property](#)
[ReplaceWithProperties property](#)
[RequestAndProcessData method](#)
[RequestRemarkOnClose property](#)
[RequireStartupScripts property](#)
[Reset method](#)
[ResetFindAndReplace method](#)
[ResetNumberOpts method](#)
[ResetOnEachPage property](#)
[ResetPrinting method](#)
[ResetWhen property](#)
[Resize method](#)
[Rest property](#)
[RestartStyleName property](#)
[Restore method](#)
[RestorePreviousView method](#)
[Result property](#)
[ResumePausedMacro property](#)
[RetainNameOfImportedFile property](#)
[RetrieveInternetFile method](#)
[RetrieveInternetFileAndOpen method](#)
[ReturnExits property](#)
[RevertToSaved method](#)
[RevertToStyle method](#)
[ReviewVersions property](#)
[ReviewVersions class](#)
[ReviewVersions method](#)
[ReviseAcceptAll method](#)
[ReviseCancelAll method](#)
[Revision class](#)

[RevisionAccept method](#)
[RevisionAcceptLayoutChange method](#)
[RevisionCancel method](#)
[RevisionCancelLayoutChange method](#)
[RevisionDisplay property](#)
[RevisionDisplay class](#)
[RevisionMark property](#)
[RevisionMarkMode property](#)
[RevisionType property](#)
[RevMarkCharacter property](#)
[RevMarkPosition property](#)
[RevMarkType property](#)
[Right property](#)
[RightAlign property](#)
[RightBorder property](#)
[RightExternalMargin property](#)
[RightMouseMenus property](#)
[RightMousePropId property](#)
[RightMousePropText property](#)
[RightPage property](#)
[RowContainer class](#)
[RowGroupLayout class](#)
[RowLayout class](#)
[RowLayoutCollection class](#)
[RowLayouts property](#)
[Rows property](#)
[RubyContainer class](#)
[RubyLayout class](#)
[RubyLayoutCollection class](#)
[RubyLayouts property](#)
[RubyMarker class](#)
[Ruler class](#)
[Run method](#)
[RunMacroOnDocEvents property](#)
[RunMacroOnLoad property](#)
[RunOnCloseDoc property](#)
[RunOnNewDoc property](#)
[RunOnOpenDoc property](#)
[RunScript method](#)

S

[Save event](#)
[Save method](#)
[SaveAs event](#)
[SaveAs method](#)
[SaveAsToNotes method](#)
[Saved event](#)
[Saved property](#)
[SavedAs event](#)
[SaveData method](#)
[SaveDivision method](#)
[SaveDivisions property](#)
[SaveGlossary method](#)
[SaveInternetFile method](#)
[SaveMacro method](#)
[SaveMergeDataFile method](#)
[SavePath property](#)

[SavePreviewFile method](#)
[SaveSections property](#)
[SaveSnapShot property](#)
[SaveSnapshot method](#)
[SaveThumbnailBitmap method](#)
[SaveToStorage method](#)
[SaveToStorageComplete method](#)
[SaveUserDefaults method](#)
[SaveVersion method](#)
[ScaleHeight property](#)
[ScaleMode property](#)
[ScalePercentage property](#)
[ScaleWidth property](#)
[ScreenPositionX property](#)
[ScreenPositionY property](#)
[ScriptDataSet class](#)
[Search method](#)
[SearchAttributes property](#)
[SearchLanguage property](#)
[Section property](#)
[Section class](#)
[SectionCollection class](#)
[SectionName property](#)
[Sections property](#)
[SectionTabs class](#)
[SectionTabs property](#)
[SectionUserName property](#)
[Select method](#)
[SelectCell method](#)
[SelectColumn method](#)
[SelectCustomIcon method](#)
[SelectDoc method](#)
[SelectedPages property](#)
[SelectEntireCellRange method](#)
[SelectEntireColumn method](#)
[SelectEntirePCol method](#)
[SelectEntirePColCellRange method](#)
[SelectEntirePColColumn method](#)
[SelectEntirePColRow method](#)
[SelectEntireRow method](#)
[SelectEntireTable method](#)
[SelectionBorderColor1 property](#)
[SelectionBorderColor2 property](#)
[SelectionBorderColor3 property](#)
[SelectionHidden property](#)
[SelectionType property](#)
[SelectParagraph method](#)
[SelectPColCell method](#)
[SelectRow method](#)
[SelectSection method](#)
[SelectSentence method](#)
[SelectStandardIcon method](#)
[SelectTab property](#)
[SelectTable method](#)
[SelectTableItem method](#)
[SelectTWAINSources method](#)
[SelectType property](#)
[SelectWord method](#)

[SendFrameToBack method](#)
[SendFrameToBackOne method](#)
[SendMailAndAttach method](#)
[SendMailSelectedText method](#)
[SequenceName property](#)
[SequenceNumber property](#)
[ServerFormat property](#)
[SetAllMargins method](#)
[SetArrayProp method](#)
[SetButtonPercent method](#)
[SetContextOfBar property](#)
[SetCursorPosition method](#)
[SetCustomNumber method](#)
[SetData method](#)
[SetDocumentEpoch method](#)
[SetFieldFormula method](#)
[SetFocus method](#)
[SetFormula method](#)
[SetIndexInfo method](#)
[SetJapanIndexInfo method](#)
[SetLastUsedFilter method](#)
[SetLineOneSide method](#)
[SetLinesAllSides method](#)
[SetLinkSource method](#)
[SetMinimumOrigin method](#)
[SetMultiCellPaste method](#)
[SetNamedProperty method](#)
[SetNoFields method](#)
[SetNumberingLevelInfo method](#)
[SetOverrideGraphic method](#)
[SetOverrideText method](#)
[SetPageBottomMargin method](#)
[SetPageTopMargin method](#)
[SetPaperNameAndUpdateSize method](#)
[SetPattern method](#)
[SetPopupAlignment method](#)
[SetPopupIndex method](#)
[SetPopupWidth method](#)
[SetPopupWidthType method](#)
[SetRGB method](#)
[SetStorage method](#)
[SetStyle method](#)
[SetTabsDialog class](#)
[SetTabsDialog property](#)
[SetTOCLevelContent method](#)
[SetTOCLevelPageInfo method](#)
[SetUpEnvelopeMerge method](#)
[SetupForCropping method](#)
[Shade method](#)
[Shadow class](#)
[Shadow property](#)
[Shape property](#)
[Show method](#)
[ShowAnyGreeting method](#)
[ShowBubbleHelp property](#)
[ShowCaretAndSelection method](#)
[ShowContainers method](#)
[ShowCursor method](#)

[ShowDivisionTabs property](#)
[ShowExportWarningMessages property](#)
[ShowFileNew property](#)
[ShowGraphicPreview property](#)
[ShowHeaderFooterBar property](#)
[ShowHiddenText property](#)
[ShowIconBars method](#)
[ShowInContext property](#)
[ShowMailDisabled property](#)
[ShowMarks property](#)
[ShowMenuCustomization property](#)
[ShowMenuWarningMessage property](#)
[ShowNoWelcomeBox property](#)
[ShowScrollBars method](#)
[ShowStatistics property](#)
[ShowStatusBar method](#)
[ShowTabs property](#)
[SilverBullet class](#)
[SilverBullet property](#)
[SilverBulletCollection class](#)
[SilverBullets property](#)
[SimulateButtonClick method](#)
[SingleCellSelected property](#)
[Size property](#)
[SizeStyleName property](#)
[SizingUnitName property](#)
[SizingUnits property](#)
[Skip method](#)
[Skipped property](#)
[SkipWordMode property](#)
[SmallCaps property](#)
[SmallCaps method](#)
[SmallFileFormat property](#)
[SmartCorrect property](#)
[SmartCorrect class](#)
[SmartCorrectCollection class](#)
[SmartCorrects property](#)
[SmartFill property](#)
[SmartFill class](#)
[SmartFillCollection class](#)
[SmartLabelsEnabled property](#)
[SmartLevel property](#)
[SmartSumColumn method](#)
[SmartSumRow method](#)
[SnapShotSaveOptions property](#)
[SortKey class](#)
[SortLevel1 property](#)
[SortLevel2 property](#)
[SortLevel3 property](#)
[SortNumbers property](#)
[SortOptions property](#)
[SortOptions class](#)
[SortOrder property](#)
[SortParagraphs method](#)
[SortType property](#)
[SortWord property](#)
[SortWordOption property](#)
[SpaceAbove property](#)

[SpaceBelow property](#)
[SpacesBetweenSentences property](#)
[Spacing property](#)
[Spacing class](#)
[SpacingStyleName property](#)
[SpacingUnitName property](#)
[SpacingUnits property](#)
[Span property](#)
[SpecialView method](#)
[SpellAddToUserDict method](#)
[SpellCheckInitialCaps property](#)
[SpellCheckRepeatedWords property](#)
[SpellCheckUserDictAlternatives property](#)
[SpellCheckWordsWithNums property](#)
[SpellClearSkippedWords method](#)
[SpellColor property](#)
[SpellFocusedColor property](#)
[SpellMarkSkippedWords method](#)
[SpellSkipAll method](#)
[SpellWord method](#)
[Split method](#)
[SplitDivision method](#)
[SplitParagraph method](#)
[SplitPercentage property](#)
[SplitWindow method](#)
[SRRReplace method](#)
[Start method](#)
[Start property](#)
[StartColumns property](#)
[StartEditMergeData method](#)
[StartEnvelopeDiv method](#)
[StartFieldInsert method](#)
[StartingColOfSelection property](#)
[StartingColStringOfSelection property](#)
[StartingNumber property](#)
[StartingRowOfSelection property](#)
[StartRow property](#)
[StateID property](#)
[StatusBar class](#)
[StatusBar property](#)
[StatusBarButton class](#)
[StatusBarButtonClicked event](#)
[StatusBarButtonCollection class](#)
[StatusBarButtonFillPopupList event](#)
[StatusBarButtonItemSelected event](#)
[StatusBarButtonOverrideGraphic event](#)
[StatusBarButtonOverrideText event](#)
[StatusBarButtonOverrideTextAndGraphic event](#)
[StatusBarButtons property](#)
[StatusBarVisible property](#)
[StatusSpellReplaceAll property](#)
[StoreInternetFile method](#)
[StrikeThrough property](#)
[StrikeThru method](#)
[StringCollection class](#)
[Style property](#)
[StyleExceptions property](#)
[StyleName property](#)

[StylePath property](#)
[StylePaths property](#)
[StyleSheetFullPath property](#)
[StyleSheetName property](#)
[StyleSheetPath property](#)
[SubPageContainer class](#)
[SubScript method](#)
[Subscript property](#)
[Suffix property](#)
[Suggestions property](#)
[Sum method](#)
[SuperPageContainer class](#)
[SuperScript method](#)
[Superscript property](#)
[SuperTable class](#)
[SuperTableCollection class](#)
[SuperTableContainer class](#)
[SuperTableContainer property](#)
[SuperTableGroupLayout class](#)
[SuperTableLayout class](#)
[SuperTableLayoutCollection class](#)
[SuperTableLayouts property](#)
[SuperTables property](#)
[SuppressHeaders property](#)

T

[TabExits property](#)
[Table property](#)
[Table class](#)
[TableCollection class](#)
[TableContainer property](#)
[TableContainer class](#)
[TableExports property](#)
[TableFill class](#)
[TableFill property](#)
[TableHeading class](#)
[TableHeadingCollection class](#)
[TableHeadingLayout class](#)
[TableHeadingLayoutCollection class](#)
[TableHeadingLayouts property](#)
[TableHeadings property](#)
[TableImports property](#)
[TableLayout class](#)
[TableLayoutCollection class](#)
[TableLayouts property](#)
[TableLine class](#)
[TableLine property](#)
[TableMarker class](#)
[TableMarkerCollection class](#)
[TableMarkers property](#)
[TableOfContents property](#)
[TableOnlyCollection class](#)
[TableOnlyCont class](#)
[TableOnlyContainer property](#)
[Tables property](#)
[TableStyleName property](#)
[TableStyles property](#)

[TabOrder property](#)
[TabRack class](#)
[TabRack property](#)
[TabRelativeTo property](#)
[TabSpacing property](#)
[TabSymbolChar property](#)
[TabType property](#)
[TeamMail method](#)
[TempFindAndReplace property](#)
[TempFoundry property](#)
[Text class](#)
[Text property](#)
[TextAfter property](#)
[TextandTableExports property](#)
[TextandTableImports property](#)
[TextAttributes property](#)
[TextBefore property](#)
[TextCollection class](#)
[TextDocument class](#)
[TextMarker class](#)
[TextMarkerCollection class](#)
[TextMarkers property](#)
[TextMarkersInSelection property](#)
[TextOrient property](#)
[TextOrientation property](#)
[Texts property](#)
[TextStyleCollection class](#)
[TextStyleName property](#)
[TextStyles property](#)
[TextTightness property](#)
[TextViewAttributes property](#)
[ThesMaxNumSynonymsReturned property](#)
[Tile method](#)
[Tile property](#)
[TileWindow property](#)
[TileWindowHorz method](#)
[TileWindowVert method](#)
[Time property](#)
[TimedSave method](#)
[TimedSaveFileExtension property](#)
[TimerTick event](#)
[Title property](#)
[TitleBarDocNumber property](#)
[TitleBarVisible property](#)
[TOCRange property](#)
[TOCSource property](#)
[TOCSuperTableLayout class](#)
[ToggleCleanScreen method](#)
[ToggleIconBar method](#)
[Top property](#)
[TopBorder property](#)
[TopExternalMargin property](#)
[TopLeftCellRowId property](#)
[TopLevelMenuText property](#)
[TotalEditingTime property](#)
[TrailingText property](#)
[Type property](#)
[Type method](#)

[TypeAbove property](#)
[TypeAboveLine property](#)
[TypeBelow property](#)
[TypeBetween property](#)
[TypeRight property](#)
[Typos property](#)

U

[UIActive property](#)
[Underline property](#)
[Underline method](#)
[Undo method](#)
[UndoLevels property](#)
[Units property](#)
[UnknownTagsInCommentNotes property](#)
[Unlink method](#)
[UnLinkContainers method](#)
[UnlinkFrameContents method](#)
[UnregisterWPDataSet method](#)
[Update method](#)
[UpdateAllMenus method](#)
[UpdateFields property](#)
[UpdateFootersText method](#)
[UpdateHeadersText method](#)
[UpdateIndexSection method](#)
[UpdateLink method](#)
[UpdateOle method](#)
[UpdateOnLoad property](#)
[UpdateOnLoadImmediate property](#)
[UpdatePageSizeChange method](#)
[UpdatePowerFields method](#)
[UpdatePowerFieldsOnNew method](#)
[UpdatePrinterBins method](#)
[UpdatePrinterChanges method](#)
[UpdateSelectedFields method](#)
[UpdateTabs method](#)
[UpdateTOC method](#)
[UpdateUI method](#)
[UpperCase property](#)
[UpperCase method](#)
[URLAddress property](#)
[URLDescription property](#)
[URLDownloadComplete property](#)
[UseConsistentSpaceBetweenSentences property](#)
[UseContents property](#)
[UseCycleKeys property](#)
[UseDefaultPrinter property](#)
[UsedFirstPageHeight property](#)
[UseFindStyle property](#)
[UseFooter property](#)
[UseGreeting property](#)
[UseHeader property](#)
[UseLSX method](#)
[UseNextStyle property](#)
[UsePrinterSettings property](#)
[UserClassNameApp property](#)
[UserClassNameFull property](#)

[UserClassNameShort property](#)
[UserDefinedFilter property](#)
[UserDictFiles property](#)
[UserDictionaryFiles property](#)
[UserDictionaryPath property](#)
[UserDictionaryPaths property](#)
[UserDictStates property](#)
[UseRelative property](#)
[UseReplaceStyle property](#)
[UserInitials property](#)
[UserInterfacePrefs property](#)
[UserInterfacePrefs class](#)
[UserLinkName property](#)
[UserName property](#)
[UseSeparatorLine property](#)
[UsesHelp property](#)
[UsesPalette property](#)
[UseSuperscriptReferenceNum property](#)
[UseTwoSpacesBetweenSentences property](#)
[UseWhen class](#)
[UseWhen property](#)

V

[Valid property](#)
[ValidateValue method](#)
[Version class](#)
[VersionCollection class](#)
[VersionID property](#)
[VersionManager property](#)
[VersionManager class](#)
[VersionName property](#)
[Versions property](#)
[VertAlign property](#)
[VerticalSplitWindow property](#)
[VertRuler property](#)
[VertScrollBarVisible property](#)
[ViewLevel property](#)
[ViewType property](#)
[Visible property](#)

W

[WasDeletedInRevMarkMode property](#)
[WasInsertedInRevMarkMode property](#)
[WaterMarkName property](#)
[WatermarksPath property](#)
[Weight property](#)
[Where property](#)
[WidowOrphan property](#)
[Width property](#)
[WidthAbove property](#)
[WidthBelow property](#)
[WidthBetween property](#)
[WidthInTwips property](#)
[Window class](#)
[WindowId property](#)
[WindowsName property](#)

[WinViewPrefs class](#)
[WinViewPrefs property](#)
[WithComments property](#)
[WMCommand event](#)
[WMCommand method](#)
[WordCount method](#)
[WordDoubleUnderline property](#)
[WordUnderline method](#)
[WordUnderline property](#)
[WorkingType property](#)
[WPApplication class](#)
[WPDataSet class](#)
[WPDataSetCollection class](#)
[WPDataSets property](#)
[WrapType property](#)
[Write method](#)
[WriteProfileString method](#)

X

[XOffset property](#)
[XPosition property](#)

Y

[YOffset property](#)
[YPosition property](#)

Z

[Zero property](#)
[ZipCode property](#)

Word Pro Classes

A
B
C
D
E
F
G
H
I
J
K
L
M
N
O
P
Q
R
S
T
U
V
W
X
Y
Z

A

Accelerators class
Alignment class
Amikake class
Application class
ApplicationWindow class
AppViewPrefs class
Attributes class
AutoRunMacro class

B

Background class
Bag class
BagCollection class
BaseCollection class
BaseContainer class
BaseObject class
BaseTable class
BetweenLines class
Bookmark class
BookmarkCollection class
BookmarkManager class
Border class
BorderLines class
Breaks class
Bullet class

C

[CellCollection class](#)
[CellContainer class](#)
[CellEngine class](#)
[CellGroupLayout class](#)
[CellLayout class](#)
[CellLayoutCollection class](#)
[CharacterBorder class](#)
[CharacterSet class](#)
[CharacterStyle class](#)
[CharacterStyleCollection class](#)
[ClickHere class](#)
[ClickHereCollection class](#)
[Color class](#)
[ColumnGroupLayout class](#)
[ConnectedLayout class](#)
[ConnectedLayoutCollection class](#)
[Content class](#)
[ContentCollection class](#)
[ContextMenuOptions class](#)

D

[DdeLink class](#)
[DdeLinkCollection class](#)
[DdeLinkManager class](#)
[Division class](#)
[DivisionCollection class](#)
[DivisionInfo class](#)
[DivisionOptions class](#)
[DocControl class](#)
[DocInfo class](#)
[DocInfoField class](#)
[DocInfoFieldCollection class](#)
[DocInfoFieldManager class](#)
[Document class](#)
[Documents class](#)
[DocWindow class](#)
[DocWindowCollection class](#)
[DropCapContainer class](#)
[DropCapLayout class](#)
[DropCapLayoutCollection class](#)

E

[Editor class](#)
[EditorCollection class](#)
[EditorManager class](#)
[EndnoteDivisionGroupNum class](#)
[EndnoteDivisionNum class](#)
[EndnoteDocNum class](#)
[EndnoteLayout class](#)
[EndnoteLayoutCollection class](#)

F

[Filter class](#)
[FilterHelper class](#)
[FindAndReplace class](#)

[Font class](#)
[FontMetrics class](#)
[FooterLayout class](#)
[FooterLayoutCollection class](#)
[Footnote class](#)
[FootnoteCollection class](#)
[FootnoteContSep class](#)
[FootnoteLayout class](#)
[FootnoteLayoutCollection class](#)
[FootnoteNumbering class](#)
[FootnoteNumOpt class](#)
[FootnoteOptions class](#)
[FootnoteSeparator class](#)
[FootnoteSepOpt class](#)
[FootnoteTable class](#)
[FormatCheckPref class](#)
[Foundry class](#)
[FrameCaptionOptions class](#)
[FrameContainer class](#)
[FrameGroupLayout class](#)
[FrameLayout class](#)
[FrameLayoutCollection class](#)

G

[Glossary class](#)
[GlossaryCollection class](#)
[Grammar class](#)
[Graphic class](#)
[GraphicCollection class](#)
[GraphicOleObject class](#)
[GraphicOleObjectCollection class](#)
[GroupLayout class](#)
[GroupLayoutCollection class](#)
[Gutter class](#)

H

[HeaderLayout class](#)
[HeaderLayoutCollection class](#)
[HTMLOptions class](#)
[HyphenationOptions class](#)

I

[IconBar class](#)
[IconBarCollection class](#)
[IconBarManager class](#)
[Indent class](#)
[IndexSection class](#)

J

[Join class](#)

K

[Kinsoku class](#)

L

[Language class](#)
[Layout class](#)
[LayoutCollection class](#)
[LayoutOverride class](#)
[LineNumberOptions class](#)
[LWPTimer class](#)

M

[Macro class](#)
[MailRouting class](#)
[Marker class](#)
[MarkerCollection class](#)
[MenuItem class](#)
[MenuItemCollection class](#)
[MergeOptions class](#)

N

[NoteContainer class](#)
[NoteLayout class](#)
[NoteLayoutCollection class](#)
[Numbering class](#)
[NumericFormat class](#)
[NumericFormatSubset class](#)

O

[OleControl class](#)
[OleObject class](#)
[OleObjectCollection class](#)
[Options class](#)
[OutlineSeqCollection class](#)
[OutlineSeqItemCollection class](#)
[OutlineStyleSequence class](#)
[OutSeqItem class](#)

P

[PageContainer class](#)
[PageLayout class](#)
[PageLayoutCollection class](#)
[ParagraphBackground class](#)
[ParagraphBorder class](#)
[ParagraphStyle class](#)
[ParagraphStyleCollection class](#)
[ParallelColsCollection class](#)
[ParallelColsContainer class](#)
[ParallelColumns class](#)
[PowerField class](#)
[PowerFieldCollection class](#)
[Preferences class](#)
[Presentation class](#)
[PrintManager class](#)
[PrintSettings class](#)

Q

(none)

R

RelativeIndent class
ReviewVersions class
Revision class
RevisionDisplay class
RowContainer class
RowGroupLayout class
RowLayout class
RowLayoutCollection class
RubyContainer class
RubyLayout class
RubyLayoutCollection class
RubyMarker class
Ruler class

S

ScriptDataSet class
Section class
SectionCollection class
SectionTabs class
SetTabsDialog class
Shadow class
SilverBullet class
SilverBulletCollection class
SmartCorrect class
SmartCorrectCollection class
SmartFill class
SmartFillCollection class
SortKey class
SortOptions class
Spacing class
StatusBar class
StatusBarButton class
StatusBarButtonCollection class
StringCollection class
SubPageContainer class
SuperPageContainer class
SuperTable class
SuperTableCollection class
SuperTableContainer class
SuperTableGroupLayout class
SuperTableLayout class
SuperTableLayoutCollection class

T

Table class
TableCollection class
TableContainer class
TableFill class
TableHeading class

[TableHeadingCollection class](#)
[TableHeadingLayout class](#)
[TableHeadingLayoutCollection class](#)
[TableLayout class](#)
[TableLayoutCollection class](#)
[TableLine class](#)
[TableMarker class](#)
[TableMarkerCollection class](#)
[TableOnlyCollection class](#)
[TableOnlyCont class](#)
[TabRack class](#)
[Text class](#)
[TextCollection class](#)
[TextDocument class](#)
[TextMarker class](#)
[TextMarkerCollection class](#)
[TextStyleCollection class](#)
[TOCSuperTableLayout class](#)

U

[UserInterfacePrefs class](#)
[UseWhen class](#)

V

[Version class](#)
[VersionCollection class](#)
[VersionManager class](#)

W

[Window class](#)
[WinViewPrefs class](#)
[WPApplication class](#)
[WPDataSet class](#)
[WPDataSetCollection class](#)

X

(none)

Y

(none)

Z

(none)

Word Pro Events

Created event
DocumentClose event
DocumentClosed event
DocumentCreate event
DocumentCreated event
DocumentExport event
DocumentExported event
DocumentImport event
DocumentImported event
DocumentInsert event
DocumentInserted event
DocumentOpen event
DocumentOpened event
DocumentPrint event
DocumentPrinted event
DocumentSave event
DocumentSaveAs event
DocumentSaved event
DocumentSavedAs event
EnterClickHere event
EnterLayout event
ExitClickHere event
GotFocus event
ImportInsert event
ImportInserted event
KeyStroke event
LostFocus event
MouseDown event
MouseUp event
Moved event
Opened event
PreClose event
PrePrint event
Printed event
Quit event
Save event
SaveAs event
Saved event
SavedAs event
StatusBarButtonClicked event
StatusBarButtonFillPopupList event
StatusBarButtonItemSelected event
StatusBarButtonOverrideGraphic event
StatusBarButtonOverrideText event
StatusBarButtonOverrideTextAndGraphic event
TimerTick event
WMCommand event

Word Pro Methods

A
B
C
D
E
F
G
H
I
J
K
L
M
N
O
P
Q
R
S
T
U
V
W
X
Y
Z

A

[Abandon method](#)
[AbortURLDownload method](#)
[AcquireTWAImage method](#)
[Activate method](#)
[ActivateAs method](#)
[Add method](#)
[AddAccelerators method](#)
[AddACondition method](#)
[AddBookmark method](#)
[AddChildToLayout method](#)
[AddContainer method](#)
[AddDdeLink method](#)
[AddDivision method](#)
[AddDivisionToPrint method](#)
[AddEditorManager method](#)
[AddField method](#)
[AddIcon method](#)
[AddIndexAllEntry method](#)
[AddIndexEntry method](#)
[AddLayoutOverride method](#)
[AddNewSectionTabs method](#)
[AddOutlineSequenceItem method](#)
[AddPopupGraphicItem method](#)
[AddPopupPointSizeItem method](#)
[AddPopupTextItem method](#)
[AddSectionTabs method](#)
[AddSmartCorrect method](#)
[AddStringToList method](#)
[AddTOCEntry method](#)

[AddVerbMenu method](#)
[AdjustShade method](#)
[Adopt method](#)
[AdviseOnRename method](#)
[AdviseOnSave method](#)
[Anchor method](#)
[AnswerMsgBox method](#)
[AnyEdits method](#)
[AppendMacro method](#)

B

[Backspace method](#)
[Backward method](#)
[BeginChange method](#)
[BeginCustomLines method](#)
[BinNameFromNumber method](#)
[Bold method](#)
[BreakLink method](#)
[BringFrameToFront method](#)
[BringFrameToFrontOne method](#)

C

[CalculateSmartLevels method](#)
[CanRepeatClickHere method](#)
[Cascade method](#)
[CascadeWindow method](#)
[CellLayout method](#)
[CellRevert method](#)
[ChangeSmartMaster method](#)
[CheckFieldEntries method](#)
[ChgLineStyle method](#)
[Clear method](#)
[ClearAll method](#)
[ClearDivisionList method](#)
[ClearParaRevisionTags method](#)
[ClearSpellHints method](#)
[ClearSplits method](#)
[ClearTempFoundry method](#)
[ClearUpdate method](#)
[Close method](#)
[CloseAll method](#)
[CloseDocWindow method](#)
[CloseMergeDataFile method](#)
[CloseObject method](#)
[CombineDivisions method](#)
[CombineSections method](#)
[CompareFiles method](#)
[Configure method](#)
[Connect method](#)
[ConnectCells method](#)
[ConnectContainer method](#)
[ConnectRows method](#)
[ConnectSectionTabs method](#)
[ConsistencyCheck method](#)
[Contract method](#)
[ContractOutlineLevel method](#)

[ConvertTo method](#)
[ConvertToClass method](#)
[Copy method](#)
[CopyDivision method](#)
[CopyItem method](#)
[CopyMeaning method](#)
[CopySelection method](#)
[Create method](#)
[CreateDataFile method](#)
[CreateDivision method](#)
[CreateDropCap method](#)
[CreateEmptyList method](#)
[CreateExternalDivision method](#)
[CreateFrame method](#)
[CreateFromBitmap method](#)
[CreateFromClipBrd method](#)
[CreateFromDataObject method](#)
[CreateFromMetafile method](#)
[CreateGlossary method](#)
[CreateGlossaryEntry method](#)
[CreateGraphic method](#)
[CreateLayer method](#)
[CreateNew method](#)
[CreateNewButton method](#)
[CreateOleEmbeddedFile method](#)
[CreateOleFromFile method](#)
[CreateOleLinkedFile method](#)
[CreateOleNew method](#)
[CreateParallelColumns method](#)
[CreateRemark method](#)
[CreateTable method](#)
[CreateTextEntryField method](#)
[CreateVersion method](#)
[CutDivision method](#)
[CutSelection method](#)

D

[DarkMode method](#)
[DataObjectGetData method](#)
[DataObjectGetDataHere method](#)
[DbfUnderline method](#)
[DeleteBag method](#)
[DeleteButton method](#)
[DeleteChars method](#)
[DeleteClickHere method](#)
[DeleteContainer method](#)
[DeleteContent method](#)
[DeleteContents method](#)
[DeleteDivision method](#)
[DeleteField method](#)
[DeleteItem method](#)
[DeleteItemByPosition method](#)
[DeleteKey method](#)
[DeleteLayout method](#)
[DeleteMacroAccelerator method](#)
[DeleteMarker method](#)
[DeleteParallelColumns method](#)

[DeleteSection method](#)
[DeleteSmartCorrect method](#)
[DeleteTab method](#)
[DeleteTable method](#)
[DeleteVersion method](#)
[Demote method](#)
[DemoteOutlineLevel method](#)
[Deselect method](#)
[Destroy method](#)
[DestroyDocWindow method](#)
[DestroyPowerFields method](#)
[Disconnect method](#)
[DisconnectCells method](#)
[DoesMarkerNameMatch method](#)
[DoesMultiCellPaste method](#)
[DoneWithRightContextMenu method](#)
[DoVerb method](#)

E

[EditHyperLink method](#)
[EditLinkInfo method](#)
[EditWrap method](#)
[Embed method](#)
[EndChange method](#)
[EndCustomLines method](#)
[Ending method](#)
[EndPrinting method](#)
[EnumerateChartLinks method](#)
[EnumerateTerm method](#)
[EnvelopeBarCode method](#)
[EnvelopePrint method](#)
[ExchangeItem method](#)
[Expand method](#)
[ExpandOutline method](#)
[ExpandOutlineLevel method](#)
[ExportAllAsNotesFX method](#)
[ExportAsNotesFX method](#)
[ExtractText method](#)

F

[Find method](#)
[FindCellLayout method](#)
[FindClass method](#)
[FindTerm method](#)
[ForceDocToLoad method](#)
[Forward method](#)
[FrameRevert method](#)
[FXGetNotesString method](#)
[FXGetNotesWriteHandle method](#)
[FXSetNotesString method](#)

G

[Get method](#)
[GetActiveList method](#)
[GetArrayProp method](#)

[GetAspectMetafilePict method](#)
[GetContents method](#)
[GetCopyFormatCategories method](#)
[GetCount method](#)
[GetCurrentMarkerName method](#)
[GetData method](#)
[GetDisplayableFilterName method](#)
[GetEnum method](#)
[GetEnvelopeDefaults method](#)
[GetFileDescription method](#)
[GetFormula method](#)
[GetInternetFile method](#)
[GetLastUsedFilter method](#)
[GetLineMix method](#)
[GetLineStyle method](#)
[GetLinkDisplayNameFileLength method](#)
[GetLinkName method](#)
[GetLinkSourceName method](#)
[GetListName method](#)
[GetMarkedText method](#)
[GetMarkerName method](#)
[GetMisspelledWord method](#)
[GetNamedProperty method](#)
[GetNameFromPage method](#)
[GetObjectList method](#)
[GetPageNumFromDisplayablePageNum method](#)
[GetParagraphNumber method](#)
[GetParagraphNumberString method](#)
[GetParaNumber method](#)
[GetPasteFormatCategories method](#)
[GetPosition method](#)
[GetPrinterInfo method](#)
[GetProfileString method](#)
[GetRGB method](#)
[GetRightMouseMenu method](#)
[GetSource method](#)
[GetSpellStatus method](#)
[GetSpellUserDictStatus method](#)
[GetStandardButtonId method](#)
[GetString method](#)
[GetTabAlignChar method](#)
[GetTabLeaderType method](#)
[GetTabPosition method](#)
[GetTabRelativeType method](#)
[GetTabType method](#)
[GetText method](#)
[GetUniqueName method](#)
[GetUserClassNameFull method](#)
[GetUserClassNameShort method](#)
[GetWordMisspelled method](#)
[Glossarize method](#)
[GlossaryInsert method](#)
[GlossaryOpen method](#)
[GoTo method](#)
[GoToBookmark method](#)
[GoToClickHere method](#)
[GoToContainer method](#)
[GoToDivision method](#)

[GoToLayout method](#)
[GotoNextParallelColumn method](#)
[GoToObject method](#)
[GoToPage method](#)
[GoToSection method](#)
[GoToTableCell method](#)
[GroupDivision method](#)

H

[HandsOffStorage method](#)
[HasNamedProperty method](#)
[Help method](#)
[Hide method](#)
[HideCaretAndSelection method](#)
[HidelconBar method](#)
[HidelconBars method](#)
[HideStatusBar method](#)
[HighlightToggle method](#)
[Hit method](#)
[HourGlass method](#)

I

[ImageProcess method](#)
[ImportGraphic method](#)
[ImportPicture method](#)
[ImportWatermarkGraphic method](#)
[IndexAll method](#)
[InitFindAndReplace method](#)
[InsertBreak method](#)
[InsertBullet method](#)
[InsertClickHere method](#)
[InsertColumnBreak method](#)
[InsertDate method](#)
[InsertDocInfo method](#)
[InsertDocument method](#)
[InsertField method](#)
[InsertFootnote method](#)
[InsertHardSpace method](#)
[InsertHyperLink method](#)
[InsertIndex method](#)
[InsertLink method](#)
[InsertMarker method](#)
[InsertNote method](#)
[InsertNumber method](#)
[InsertOleDivision method](#)
[InsertOne method](#)
[InsertPageBreak method](#)
[InsertPageLayout method](#)
[InsertPageNumber method](#)
[InsertPath method](#)
[InsertRowOrColumn method](#)
[InsertRuby method](#)
[InsertSection method](#)
[InsertSpecialTab method](#)
[InsertTab method](#)
[InsertText method](#)

[InsertTOC method](#)
[InternalCopy method](#)
[InternalCut method](#)
[InternalPaste method](#)
[InternetExtraFile method](#)
[InternetFileExists method](#)
[Invalidate method](#)
[InvalidateButton method](#)
[InvalidateWholeBar method](#)
[IsCaseExact method](#)
[IsDataNameUsed method](#)
[IsEmpty method](#)
[IsExportedAsNotesFX method](#)
[IsFilterTypePresent method](#)
[IsMarkerEqualToSelection method](#)
[IsPointWithin method](#)
[IsWMCCommandValid method](#)
[Italic method](#)
[Item method](#)

J

(none)

K

(none)

L

[LabelCreate method](#)
[LabelMerge method](#)
[LeastRecentVersion method](#)
[Link method](#)
[LinkContainers method](#)
[LinkFrameContents method](#)
[Localize method](#)
[LowerCase method](#)

M

[MacroEndRecord method](#)
[MacroPlay method](#)
[MacroRecord method](#)
[MailDocument method](#)
[MakeTableFromText method](#)
[MakeUniqueLinkName method](#)
[ManualFrame method](#)
[ManualLinkFrames method](#)
[ManualTable method](#)
[Mark method](#)
[MarkRevisionInsert method](#)
[Maximize method](#)
[Merge method](#)
[MergeAddDataRecord method](#)
[MergeSetDataFile method](#)
[MergeStart method](#)
[Messages method](#)

[Minimize method](#)
[MirrorPage method](#)
[MorphSelectionToTable method](#)
[MostRecentVersion method](#)
[Move method](#)
[MoveDivision method](#)
[MoveDown method](#)
[MoveItem method](#)
[MoveParagraph method](#)
[MoveToBack method](#)
[MoveToEnd method](#)
[MoveToFront method](#)
[MoveToStart method](#)
[MoveUp method](#)

N

[NewDivision method](#)
[NewDocument method](#)
[NewFrame method](#)
[NewInfoBusLink method](#)
[NewItem method](#)
[NewItemByPosition method](#)
[NewWindow method](#)
[Next method](#)
[NextCycleAlign method](#)
[NextCycleAttribute method](#)
[NextCycleBullet method](#)
[NextCycleFont method](#)
[NextCycleFontSize method](#)
[NextCycleIndent method](#)
[NextCycleNumber method](#)
[NextCycleStyle method](#)
[NextToObject method](#)
[NextVersion method](#)
[NormalText method](#)
[NumberOfVersions method](#)

O

[Open method](#)
[OpenDataFile method](#)
[OpenDocument method](#)
[OpenDocumentFromNotes method](#)
[OpenFromStorage method](#)
[OpenObject method](#)
[OutlineMoveTextDown method](#)
[OutlineMoveTextUp method](#)
[OutlineStyleSequence method](#)

P

[PageDown method](#)
[PageUp method](#)
[Paste method](#)
[PasteDivision method](#)
[PasteLink method](#)
[PasteSpecial method](#)

[PColConnectCells method](#)
[PColConnectRows method](#)
[PColDisconnectCells method](#)
[PColSelectColumn method](#)
[PColSelectRow method](#)
[PColSelectTable method](#)
[Play method](#)
[Present method](#)
[Previous method](#)
[PreviousVersion method](#)
[Print method](#)
[PrintOut method](#)
[ProcessAccelKey method](#)
[Promote method](#)
[PromoteOutlineLevel method](#)
[Purge method](#)

Q

[QuickAlignFrame method](#)
[QuickAlignTable method](#)
[Quit method](#)

R

[Read method](#)
[Redo method](#)
[Refresh method](#)
[RegisterWPDataSet method](#)
[Release method](#)
[Remove method](#)
[RemoveBookmark method](#)
[RemoveChildFromLayout method](#)
[RemoveDataFile method](#)
[RemoveDdeLink method](#)
[RemoveDepOnDocFile method](#)
[RemoveDivision method](#)
[RemoveEditor method](#)
[RemoveHyperlink method](#)
[RemoveIndexEntry method](#)
[RemoveList method](#)
[RemoveNamedProperty method](#)
[RemovePersistentAccelerators method](#)
[RemoveTOCEntry method](#)
[Render method](#)
[RenderClipBitmap method](#)
[RenderClipDIB method](#)
[RenderClipMetafile method](#)
[RenderClipPalette method](#)
[Repaint method](#)
[Replace method](#)
[ReplaceAll method](#)
[ReplaceCmd method](#)
[ReplaceContents method](#)
[RequestAndProcessData method](#)
[Reset method](#)
[ResetFindAndReplace method](#)
[ResetNumberOpts method](#)

[ResetPrinting method](#)
[Resize method](#)
[Restore method](#)
[RestorePreviousView method](#)
[RetrieveInternetFile method](#)
[RetrieveInternetFileAndOpen method](#)
[RevertToSaved method](#)
[RevertToStyle method](#)
[ReviewVersions method](#)
[ReviseAcceptAll method](#)
[ReviseCancelAll method](#)
[RevisionAccept method](#)
[RevisionAcceptLayoutChange method](#)
[RevisionCancel method](#)
[RevisionCancelLayoutChange method](#)
[Run method](#)
[RunScript method](#)

S

[Save method](#)
[SaveAs method](#)
[SaveAsToNotes method](#)
[SaveData method](#)
[SaveDivision method](#)
[SaveGlossary method](#)
[SaveInternetFile method](#)
[SaveMacro method](#)
[SaveMergeDataFile method](#)
[SavePreviewFile method](#)
[SaveSnapshot method](#)
[SaveThumbnailBitmap method](#)
[SaveToStorage method](#)
[SaveToStorageComplete method](#)
[SaveUserDefaults method](#)
[SaveVersion method](#)
[Search method](#)
[Select method](#)
[SelectCell method](#)
[SelectColumn method](#)
[SelectCustomIcon method](#)
[SelectDoc method](#)
[SelectEntireCellRange method](#)
[SelectEntireColumn method](#)
[SelectEntirePCol method](#)
[SelectEntirePColCellRange method](#)
[SelectEntirePColColumn method](#)
[SelectEntirePColRow method](#)
[SelectEntireRow method](#)
[SelectEntireTable method](#)
[SelectParagraph method](#)
[SelectPColCell method](#)
[SelectRow method](#)
[SelectSection method](#)
[SelectSentence method](#)
[SelectStandardIcon method](#)
[SelectTable method](#)
[SelectTableItem method](#)

[SelectTWAINSource method](#)
[SelectWord method](#)
[SendFrameToBack method](#)
[SendFrameToBackOne method](#)
[SendMailAndAttach method](#)
[SendMailSelectedText method](#)
[SetAllMargins method](#)
[SetArrayProp method](#)
[SetButtonPercent method](#)
[SetCursorPosition method](#)
[SetCustomNumber method](#)
[SetData method](#)
[SetDocumentEpoch method](#)
[SetFieldFormula method](#)
[SetFocus method](#)
[SetFormula method](#)
[SetIndexInfo method](#)
[SetJapanIndexInfo method](#)
[SetLastUsedFilter method](#)
[SetLineOneSide method](#)
[SetLinesAllSides method](#)
[SetLinkSource method](#)
[SetMinimumOrigin method](#)
[SetMultiCellPaste method](#)
[SetNamedProperty method](#)
[SetNoFields method](#)
[SetNumberingLevelInfo method](#)
[SetOverrideGraphic method](#)
[SetOverrideText method](#)
[SetPageBottomMargin method](#)
[SetPageTopMargin method](#)
[SetPaperNameAndUpdateSize method](#)
[SetPattern method](#)
[SetPopupAlignment method](#)
[SetPopupIndex method](#)
[SetPopupWidth method](#)
[SetPopupWidthType method](#)
[SetRGB method](#)
[SetStorage method](#)
[SetStyle method](#)
[SetTOCLevelContent method](#)
[SetTOCLevelPageInfo method](#)
[SetUpEnvelopeMerge method](#)
[SetupForCropping method](#)
[Shade method](#)
[Show method](#)
[ShowAnyGreeting method](#)
[ShowCaretAndSelection method](#)
[ShowContainers method](#)
[ShowCursor method](#)
[ShowIconBars method](#)
[ShowScrollBars method](#)
[ShowStatusBar method](#)
[SimulateButtonClick method](#)
[Skip method](#)
[SmallCaps method](#)
[SmartSumColumn method](#)
[SmartSumRow method](#)

[SortParagraphs method](#)
[SpecialView method](#)
[SpellAddToUserDict method](#)
[SpellClearSkippedWords method](#)
[SpellMarkSkippedWords method](#)
[SpellSkipAll method](#)
[SpellWord method](#)
[Split method](#)
[SplitDivision method](#)
[SplitParagraph method](#)
[SplitWindow method](#)
[SRReplace method](#)
[Start method](#)
[StartEditMergeData method](#)
[StartEnvelopeDiv method](#)
[StartFieldInsert method](#)
[StoreInternetFile method](#)
[StrikeThru method](#)
[SubScript method](#)
[Sum method](#)
[SuperScript method](#)

T

[TeamMail method](#)
[Tile method](#)
[TileWindowHorz method](#)
[TileWindowVert method](#)
[TimedSave method](#)
[ToggleCleanScreen method](#)
[ToggleIconBar method](#)
[Type method](#)

U

[Underline method](#)
[Undo method](#)
[Unlink method](#)
[UnLinkContainers method](#)
[UnlinkFrameContents method](#)
[UnregisterWPDataSet method](#)
[Update method](#)
[UpdateAllMenus method](#)
[UpdateFootersText method](#)
[UpdateHeadersText method](#)
[UpdateIndexSection method](#)
[UpdateLink method](#)
[UpdateOle method](#)
[UpdatePageSizeChange method](#)
[UpdatePowerFields method](#)
[UpdatePowerFieldsOnNew method](#)
[UpdatePrinterBins method](#)
[UpdatePrinterChanges method](#)
[UpdateSelectedFields method](#)
[UpdateTabs method](#)
[UpdateTOC method](#)
[UpdateUI method](#)
[UpperCase method](#)

UseLSX method

V

ValidateValue method

W

WMCommand method

WordCount method

WordUnderline method

Write method

WriteProfileString method

X

(none)

Y

(none)

Z

(none)

Word Pro Properties

A
B
C
D
E
F
G
H
I
J
K
L
M
N
O
P
Q
R
S
T
U
V
W
X
Y
Z

A

[Abilities property](#)
[AboutToReturnToOriginator property](#)
[AbsoluteOn property](#)
[AbsoluteTextOrientation property](#)
[AbsoluteXPos property](#)
[AbsoluteYPos property](#)
[Accelerator property](#)
[Accelerators property](#)
[AccessibilityOn property](#)
[Action property](#)
[Active property](#)
[ActiveDocument property](#)
[ActiveDocWindow property](#)
[ActualEnumName property](#)
[ActualName property](#)
[Address1 property](#)
[Address2 property](#)
[Afid property](#)
[AfidClassName property](#)
[Align property](#)
[Alignment property](#)
[AlignmentChar property](#)
[AlignmentType property](#)
[AlignStyleName property](#)
[All property](#)
[AllBorders property](#)
[AllowAlternateVerification property](#)
[AllowListEdit property](#)
[AllowListMultiValues property](#)

[AllowResizeWhenCrop property](#)
[AlternateName property](#)
[Always property](#)
[Amikake property](#)
[AmikakeName property](#)
[AmikakeType property](#)
[Amount property](#)
[AmountOfSpaceAbove property](#)
[AmountOfSpaceAboveLine property](#)
[AmountOfSpaceBelow property](#)
[AmtTether property](#)
[AmtToRotateContent property](#)
[AmtToTetherFrom property](#)
[Animated property](#)
[AnyNumber property](#)
[AnyOLEDELinks property](#)
[AppFoundry property](#)
[Application property](#)
[ApplicationWindow property](#)
[ApplyAdjectivePos property](#)
[ApplyAdjectNounPart property](#)
[ApplyAgreementWithHereThere property](#)
[ApplyAnglicisms property](#)
[ApplyArchaicExpressions property](#)
[ApplyArticleAgreement property](#)
[ApplyBadComparatives property](#)
[ApplyBadInflection property](#)
[ApplyBadNoun property](#)
[ApplyBadNounGender property](#)
[ApplyBadPlural property](#)
[ApplyBadPrepositions property](#)
[ApplyBelgianExpression property](#)
[ApplyBorrowedForeign property](#)
[ApplyBureuaJargon property](#)
[ApplyCalque property](#)
[ApplyCapitalizationCheck property](#)
[ApplyClauseErrors property](#)
[ApplyCliches property](#)
[ApplyColloquialExpression property](#)
[ApplyCommonlyConfusedWords property](#)
[ApplyCommonMisspell property](#)
[ApplyComplexWords property](#)
[ApplyConfusedEasy property](#)
[ApplyConfusedEnglish property](#)
[ApplyConfusedHard property](#)
[ApplyConfusedMedium property](#)
[ApplyConfusedVerb property](#)
[ApplyConsecutiveNouns property](#)
[ApplyContractions property](#)
[ApplyDerogatory property](#)
[ApplyDifferentPrep property](#)
[ApplyDoubleNegative property](#)
[ApplyDoublePlural property](#)
[ApplyDoubleWordCheck property](#)
[ApplyElision property](#)
[ApplyEnglishDerived property](#)
[ApplyEnglishWords property](#)
[ApplyExotic property](#)

[ApplyExtraPrepositionCheck property](#)
[ApplyFalseFriend property](#)
[ApplyFemaleOccupation property](#)
[ApplyFixedExpression property](#)
[ApplyForeignWord property](#)
[ApplyFormalTerms property](#)
[ApplyFormatErrors property](#)
[ApplyGallicisms property](#)
[ApplyGenderExpressions property](#)
[ApplyGermanisms property](#)
[ApplyHomoGraphs property](#)
[ApplyHomonyms property](#)
[ApplyHomonymsEasy property](#)
[ApplyHomonymsHard property](#)
[ApplyHomoPhone1 property](#)
[ApplyHomoPhone2 property](#)
[ApplyHomoPhones property](#)
[ApplyIncorrectPlural property](#)
[ApplyInformalExpressions property](#)
[ApplyJargonWords property](#)
[ApplykSplitInfinitives property](#)
[ApplyLowercaseAdjective property](#)
[ApplyLowercaseColor property](#)
[ApplyLowercaseNumbers property](#)
[ApplyLowercasePhrases property](#)
[ApplyLowercasePronouns property](#)
[ApplyMassVsCount property](#)
[ApplyMisspelledExpressions property](#)
[ApplyMisspelledForeignExpressions property](#)
[ApplyMisspelledItalian property](#)
[ApplyMisspelledWords property](#)
[ApplyMisusedWords property](#)
[ApplyNonStandardExpression property](#)
[ApplyNonStandardModifiers property](#)
[ApplyNoudModifierOrderCheck property](#)
[ApplyNounConsistency property](#)
[ApplyNounPhraseAgree property](#)
[ApplyNSAdjective property](#)
[ApplyNSClause property](#)
[ApplyNSCompare property](#)
[ApplyNSContract property](#)
[ApplyNSGeography property](#)
[ApplyNSInflection property](#)
[ApplyNSNegation property](#)
[ApplyNSPrep property](#)
[ApplyNSPronoun property](#)
[ApplyNSSpell property](#)
[ApplyNSUsage property](#)
[ApplyNSVerbForm property](#)
[ApplyOpenClosedSpelling property](#)
[ApplyOpenUsage property](#)
[ApplyOverusedPhrases property](#)
[ApplyPassiveVerbErrors property](#)
[ApplyPostClitAgree property](#)
[ApplyPrepExpression property](#)
[ApplyPrepositionalPhrases property](#)
[ApplyPretentiousWords property](#)
[ApplyPronounErrors property](#)

[ApplyPunctuationErrors property](#)
[ApplyRedundantExpressions property](#)
[ApplyRegionalExpression property](#)
[ApplyRelatedWord property](#)
[ApplySensitiveExp property](#)
[ApplySexistExpressions property](#)
[ApplySpellStandard property](#)
[ApplyStockPhrase property](#)
[ApplyStyleParameters property](#)
[ApplySubjectVerbAgreement property](#)
[ApplySwedishGender property](#)
[ApplySwedishNegation property](#)
[ApplySwedishUsage property](#)
[ApplyTrite property](#)
[ApplyTwoGender property](#)
[ApplyTypicalMisspell property](#)
[ApplyUnGrammaticalExpressions property](#)
[ApplyVagueQuantifiers property](#)
[ApplyVerbGroupConsistency property](#)
[ApplyWeakModifiers property](#)
[ApplyWordChoice property](#)
[ApplyWordCompoundingCheck property](#)
[ApplyWordConfusion property](#)
[ApplyWordForm property](#)
[ApplyWordGender property](#)
[ApplyWordParts property](#)
[ApplyWordyPhraseCheck property](#)
[AppViewPrefs property](#)
[AreDisabledIconsGrayed property](#)
[Ascent property](#)
[AsciiCodePage property](#)
[AtBeginning property](#)
[AtBeginningOfLine property](#)
[AtBeginningOfObject property](#)
[AtBeginningOfParagraph property](#)
[AtBeginningOfStream property](#)
[AtBeginningOfWord property](#)
[AtEnd property](#)
[AtEndOfLine property](#)
[AtEndOfObject property](#)
[AtEndOfParagraph property](#)
[AtEndOfStream property](#)
[AtEndOfWord property](#)
[Attributes property](#)
[AttrStyleName property](#)
[AuthorName property](#)
[AutoBackup property](#)
[AutoHyphenate property](#)
[AutomaticLink property](#)
[AutoRunMacro property](#)
[AutoSave property](#)
[AutoSaveMinutes property](#)
[AutoVersion property](#)

B

[BackColor property](#)
[BackColorIndex property](#)

[Background property](#)
[BackgroundPrintingOn property](#)
[BackgroundSpellingOn property](#)
[BackupPath property](#)
[BackupPaths property](#)
[Bags property](#)
[BaseLineOffset property](#)
[BaseTable property](#)
[BaseTables property](#)
[BaseURL property](#)
[BetweenLines property](#)
[BinName property](#)
[BinNames property](#)
[BlockPaint property](#)
[Blue property](#)
[BodyOnly property](#)
[Bold property](#)
[BookletPrinting property](#)
[BookmarkManager property](#)
[Bookmarks property](#)
[BookmarksByMarkerName property](#)
[BorderLines property](#)
[BorderOffset property](#)
[BorderStyleName property](#)
[BottomBorder property](#)
[BottomExternalMargin property](#)
[Breaks property](#)
[BreaksStyleName property](#)
[Bullet property](#)
[BulletFonts property](#)
[BulletStyleName property](#)

C

[CanCreatePreviewBitmap property](#)
[CanEditProperty property](#)
[CanEmbed property](#)
[CanWePrint property](#)
[Caption property](#)
[Case property](#)
[Cell property](#)
[CellEngine property](#)
[CellEngines property](#)
[CellLayout property](#)
[CellLayouts property](#)
[CellLayoutStyles property](#)
[CellStyleName property](#)
[Center property](#)
[Changed property](#)
[ChangeKeyboardToLanguage property](#)
[ChangeTextToMatchkeyboard property](#)
[CharacterBorder property](#)
[CharacterBorderName property](#)
[CharacterSet property](#)
[CharacterSetName property](#)
[CharacterStyle property](#)
[CharacterStyleName property](#)
[CharacterStyles property](#)

[CharSet property](#)
[Checked property](#)
[CheckForScriptEnumError property](#)
[CheckForScriptPropertyError property](#)
[ChildLayouts property](#)
[CityState property](#)
[ClassId property](#)
[ClassName property](#)
[CleanScreenMode property](#)
[CleanScreenOnStartUp property](#)
[ClickHerePrompts property](#)
[ClickHeres property](#)
[ClientHeight property](#)
[ClientWidth property](#)
[ClientWndHeight property](#)
[ClientWndWidth property](#)
[CloseDocMacroName property](#)
[CodePage property](#)
[Collapsible property](#)
[Collate property](#)
[Color property](#)
[ColorOverride property](#)
[ColumnBalance property](#)
[ColumnGap property](#)
[ColumnLayouts property](#)
[ColumnNumber property](#)
[ColumnWidth property](#)
[Company property](#)
[Condition property](#)
[ConditionType property](#)
[ConnectedLayouts property](#)
[Container property](#)
[Content property](#)
[ContentHeight property](#)
[ContentName property](#)
[Contents property](#)
[ContentStyleName property](#)
[ContentType property](#)
[ContentWidth property](#)
[ContextLocked property](#)
[ContextMenuOptions property](#)
[ContinuedFromAlignment property](#)
[ContinuedFromMessage property](#)
[ContinuedFromStory property](#)
[ContinuedOnAlignment property](#)
[ContinuedOnMessage property](#)
[ContinuedOnStory property](#)
[ConvertOnNew property](#)
[Copies property](#)
[Count property](#)
[CountBlankLines property](#)
[CreatePreviewBitmap property](#)
[CreationDateString property](#)
[CreationTimeString property](#)
[CreationTimeValue property](#)
[Crop property](#)
[CurrentCell property](#)
[CurrentColumn property](#)

CurrentEditor property
CurrentLanguage property
CurrentlyPrintingInBackground property
CurrentRow property
CurrentRunningScriptName property
CurrentRunningScriptPath property
CurrentVersion property
CustomLength property
CustomViewLevel property

D

DataFileFieldNames property
DataFileName property
DataFormat property
DataNames property
DateCreatedValue property
DateRevisedValue property
DdeEnabled property
DdeLinkManager property
DdeLinks property
DdeOutboundInfo property
DebugVariable property
DefaultBinName property
DefaultCellStyleDescription property
DefaultColumnStyleName property
DefaultDropCapStyleDescription property
DefaultFilePath property
DefaultFileType property
DefaultFrameStyleDescription property
DefaultLatinFont property
DefaultLeftColumnStyleName property
DefaultMenu property
DefaultNewCategory property
DefaultNonLatinFont property
DefaultPageHeight property
DefaultPageStyleDescription property
DefaultPageWidth property
DefaultPitch property
DefaultRightColumnStyleName property
DefaultTableStyleDescription property
DefaultTextStyleDescription property
DefCellStyleName property
DefColWidth property
Definition property
DefRowHeight property
DelayedGreeting property
DeleteFont property
DemandLoad property
Descent property
Description property
DescriptionFileName property
DiagonalLines property
DirectionDown property
DirectionLeft property
DirectionRight property
DirectionUp property
DisableClickHereAction property

[DisableClickHeres property](#)
[DisableConsistencyCheck property](#)
[DisableExportToNotes property](#)
[DisableHarmlessMessages property](#)
[DisableVersionReview property](#)
[DisplayablePageNum property](#)
[DisplayAsIcon property](#)
[DistanceFromLeftMargin property](#)
[Division property](#)
[DivisionInfo property](#)
[DivisionName property](#)
[DivisionNames property](#)
[DivisionOptions property](#)
[Divisions property](#)
[DocControl property](#)
[DocControlPassword property](#)
[DocControlProtection property](#)
[DocControlRestrictedToEditor property](#)
[DocInfo property](#)
[DocOptions property](#)
[DocPath property](#)
[DocSize property](#)
[Document property](#)
[DocumentLevel property](#)
[DocumentPaths property](#)
[Documents property](#)
[DocVersionID property](#)
[DocWindows property](#)
[DoInitialCaps property](#)
[DoSentenceInitialCaps property](#)
[DoSmartLinks property](#)
[DoSmartQuotes property](#)
[DoubleUnderline property](#)
[DownLoadGraphics property](#)
[DragDropOn property](#)
[DriverName property](#)
[DropCapPosition property](#)
[DropCaps property](#)
[DropCapStyleName property](#)
[DropCapStyles property](#)

E

[Editable property](#)
[EditorInitials property](#)
[EditorManager property](#)
[EditorName property](#)
[Editors property](#)
[EditorVerificationType property](#)
[EffectiveColumnWidth property](#)
[EMail property](#)
[Embedded property](#)
[EmbedFonts property](#)
[EmitsEvents property](#)
[Enabled property](#)
[EndingColOfSelection property](#)
[EndingRowOfSelection property](#)
[EndnoteDivisionGroupNum property](#)

[EndnoteDivisionNum property](#)
[EndnoteDocNum property](#)
[Endnotes property](#)
[EntireDocument property](#)
[Enumeration property](#)
[Epoch property](#)
[EqnFontHeight property](#)
[Expandable property](#)
[ExportToNotesFX property](#)
[ExternallyControlledUndo property](#)
[ExternalName property](#)
[ExternalType property](#)

F

[FaceName property](#)
[FaceNames property](#)
[FaceStyleName property](#)
[FaxNumber property](#)
[FieldDelimiter property](#)
[FieldDelimiterText property](#)
[FieldManager property](#)
[FieldNumber property](#)
[Fields property](#)
[FieldType property](#)
[FileFormat property](#)
[FilePassword property](#)
[FileProtectionType property](#)
[FilesToCompare property](#)
[FileType property](#)
[FillerPageText property](#)
[FillStyle property](#)
[Filter property](#)
[FilterHelper property](#)
[FilterName property](#)
[FindAndReplace property](#)
[FindExactCase property](#)
[FindFont property](#)
[FindForwardDirection property](#)
[FindString property](#)
[FindStyleName property](#)
[FindWithProperties property](#)
[FinishedSpellChecking property](#)
[First property](#)
[FirstChild property](#)
[FirstCursorableDivision property](#)
[FirstDivision property](#)
[FirstName property](#)
[FirstPage property](#)
[FitType property](#)
[FixAcronymns property](#)
[FixBullets property](#)
[Font property](#)
[FontColor property](#)
[FontFaceSupport property](#)
[FontMatching property](#)
[FontMetrics property](#)
[FontName property](#)

[FontStyleName property](#)
[FontUnitName property](#)
[FontUnits property](#)
[Footer property](#)
[Footers property](#)
[FooterStyles property](#)
[FootnoteContSep property](#)
[FootnoteLayouts property](#)
[FootnoteNumbering property](#)
[FootnoteOptions property](#)
[Footnotes property](#)
[FootnoteSeparator property](#)
[ForeColorIndex property](#)
[FormatCheckPreferences property](#)
[FormatType property](#)
[Formula property](#)
[Foundry property](#)
[Frame property](#)
[FrameCaptionOptions property](#)
[Frames property](#)
[FrameStyleName property](#)
[FrameStyles property](#)
[FreeMenus property](#)
[FullName property](#)

G

[GapBetweenPanels property](#)
[GetAfidHelpInfo property](#)
[GetButtonId property](#)
[GetButtonType property](#)
[GetFilterExtension property](#)
[GetFilterExtForDialogBox property](#)
[GetFilterId property](#)
[GetFormatName property](#)
[GetHomeDirectory property](#)
[GetItemName property](#)
[GetRedoWhatDesc property](#)
[GetServerName property](#)
[GetStatus property](#)
[GetTopicName property](#)
[GetUndoWhatDesc property](#)
[GetValue property](#)
[GIFAnimationOn property](#)
[GlossaryDataFileName property](#)
[GlossaryDataFiles property](#)
[GlossaryDataPaths property](#)
[GlossaryPath property](#)
[Glossarys property](#)
[GrammarFormalityLevel property](#)
[GrammarOptions property](#)
[GrammarProofLevel property](#)
[GrammarSetName property](#)
[Graphic property](#)
[GraphicExports property](#)
[GraphicImports property](#)
[GraphicOleObject property](#)
[GraphicPath property](#)

[GraphicPaths property](#)
[Graphics property](#)
[GraphicType property](#)
[Green property](#)
[Greeting property](#)
[GridDistance property](#)
[GridType property](#)
[Groups property](#)
[Gutter property](#)

H

[Hang property](#)
[HasContents property](#)
[HasFocus property](#)
[HasIndex property](#)
[HasLocalTabs property](#)
[HasTabs property](#)
[HasTOC property](#)
[Header property](#)
[Headers property](#)
[HeaderStyles property](#)
[Heading property](#)
[Height property](#)
[HelpText property](#)
[HiddenMode property](#)
[HideFormula property](#)
[HideOutlineLevels property](#)
[HighLightMode property](#)
[HiLiteColor property](#)
[HorizontalSplitWindow property](#)
[HorzScrollBarVisible property](#)
[HorzRuler property](#)
[HTMLOptions property](#)
[Hwnd property](#)
[HyphenateLastWordInColumnOrPage property](#)
[HyphenateLastWordInPara property](#)
[HyphenationOptions property](#)
[HyphZoneAfter property](#)
[HyphZoneBefore property](#)

I

[IconBarManager property](#)
[IconBarPositionState property](#)
[IconBars property](#)
[IconBarSets property](#)
[IconHelpText property](#)
[IconPath property](#)
[IconPaths property](#)
[IconScript property](#)
[IconSetName property](#)
[IconSize property](#)
[ID property](#)
[IDispatch property](#)
[IgnoreSoftHyphens property](#)
[IgnoreTab property](#)
[ImportAsSource property](#)

[IncludeFooter property](#)
[IncludeHeader property](#)
[IncludeInitialsInNotes property](#)
[IncludeList property](#)
[IncludeURL property](#)
[Indent property](#)
[IndentFromLeft property](#)
[IndentStyleName property](#)
[IndexAlphabeticSeparator property](#)
[IndexDivision property](#)
[IndexIndentType property](#)
[IndexParent property](#)
[IndexPrimaryStyleName property](#)
[IndexRange property](#)
[IndexSecondaryStyleName property](#)
[IndexSection property](#)
[IndexSeparatorStyleName property](#)
[IndexSource property](#)
[InDocument property](#)
[InfoBoxSelectionText property](#)
[InitializeRoute property](#)
[Initialize property](#)
[InitialParaStyle property](#)
[InitialsForFilters property](#)
[InsertFont property](#)
[InsertionMode property](#)
[Interactive property](#)
[Interval property](#)
[IsActionOnButtonDown property](#)
[IsActive property](#)
[IsAmikake property](#)
[IsAmiProTableImport property](#)
[IsAnnotateOnly property](#)
[IsAsciiCRLF property](#)
[IsAsciiKeepStyle property](#)
[IsAutoGrow property](#)
[IsBadReference property](#)
[IsBorder property](#)
[IsBothSidesEqual property](#)
[IsBreakable property](#)
[IsBubbleHelp property](#)
[IsCellMenuEnabled property](#)
[IsChanged property](#)
[IsChangedOtherThanLinkTo property](#)
[IsChangedSinceTimeSave property](#)
[IsChangedToLinkTo property](#)
[IsChartLink property](#)
[IsChildSpannable property](#)
[IsCollapsed property](#)
[IsCollapsible property](#)
[IsColumnBreakable property](#)
[IsColumnBreakAfter property](#)
[IsColumnBreakBefore property](#)
[IsComplex property](#)
[IsConnected property](#)
[IsContinuedFrom property](#)
[IsContinuedOn property](#)
[IsCumulative property](#)

IsDebug property
IsDisableWarningMessages property
IsDisplayMisspelled property
IsDivisionExternal property
IsDocLoading property
IsDocObject property
IsDocumentInRoute property
IsDoubleWordError property
IsDraw property
IsEmpty property
IsEmptyDoc property
IsEnabled property
IsEndnoteDivision property
IsEquation property
IsErrorChecking property
IsExpandDown property
IsExpandLeft property
IsExpandRight property
IsExpandUp property
IsExportedToNotesFX property
IsExternalFile property
IsFixedLength property
IsFooter property
IsFrameCaptionOptions property
IsFrameMenuEnabled property
IsGotoable property
IsGrabBar property
IsGrammarError property
IsGraphicalObject property
IsGraphicMenuEnabled property
IsHangover property
IsHeader property
IsHiddenMark property
IsHideHeaderFooter property
IsHighlightNote property
IsHorizontalScrollBarCleanScrn property
IsIconDepressible property
IsInBulletEditMode property
IsIndex property
IsIndexMarkAble property
IsInDraft property
IsInOleDivision property
IsInOutline property
IsInPageSort property
IsLastStop property
IsLesser property
IsLocal property
IsLocked property
IsLockedForRevisions property
IsLotusChart property
IsMarginSameAsParent property
IsMarginsInColor property
IsMarkerValid property
IsMisspelled property
IsNoPrintZone property
IsNotCopyable property
IsNotCopyImage property
IsNotGroupable property

IsNoUICommAllowed property
IsOpen property
IsOverridden property
IsOverride property
IsPageBreak property
IsPageBreakAfter property
IsPageBreakBefore property
IsPageBreakMarks property
IsPageBreakWithin property
IsParagraphNumberingDown property
IsParagraphParent property
IsParallelColumnsMenuEnabled property
IsPartOfGroup property
IsPersistent property
IsPrePrintedForm property
IsPrintable property
IsPrintClickHereBlocks property
IsPrivate property
IsPrompting property
IsProtected property
IsRegistered property
IsRepeat property
IsReplaceable property
IsResetParagraphNumber property
IsRevisionMark property
IsSame property
IsSave property
IsScalable property
IsScalableBorder property
IsScrollable property
IsShowing property
IsSingleClickEntry property
IsSizable property
IsSizingViaMouse property
IsSmartCorrectEnabled property
IsSmartEditEnabled property
IsSmartShadeEnabled property
IsSnapTo property
IsSortFromEnd property
IsSpellBarUp property
IsSpellMode property
IsSqueeze property
IsStyle property
IsStyleSheet property
IsSymbolic property
IsTableHeading property
IsTemp property
IsTextLocked property
IsTextMenuEnabled property
IsTOC property
IsTOCMarkAble property
IsTrueType property
IsUndoOn property
IsUpdateAutomatic property
IsUpdateIndex property
IsUpdateTOC property
IsValid property
IsVerticalScrollBarCleanScrn property

[IsViewAnchor property](#)
[IsViewBookmarks property](#)
[IsViewClickHereBlocks property](#)
[IsViewColGuides property](#)
[IsViewColumnBreakMarks property](#)
[IsViewDDELinks property](#)
[IsViewHorzRuler property](#)
[IsViewHorzScrollBar property](#)
[IsViewMenuCleanScrn property](#)
[IsViewNotes property](#)
[IsViewOutlineFlowToScreen property](#)
[IsViewOutlineIndent property](#)
[IsViewPageLayoutMarks property](#)
[IsViewParallelColumnBorder property](#)
[IsViewPictures property](#)
[IsViewReturnIconCleanScrn property](#)
[IsViewReturns property](#)
[IsViewRubyMarks property](#)
[IsViewRulerMarks property](#)
[IsViewSectionBreakMarks property](#)
[IsViewSectionTabs property](#)
[IsViewSmartIconsCleanScrn property](#)
[IsViewStatusBarCleanScrn property](#)
[IsViewStatusSpell property](#)
[IsViewTableCellSelection property](#)
[IsViewTableGridLines property](#)
[IsViewTableHeadings property](#)
[IsViewTabs property](#)
[IsViewThumbBarCleanScrn property](#)
[IsViewTitleBarCleanScrn property](#)
[IsViewVertRuler property](#)
[IsViewVertScrollBar property](#)
[IsWordProChart property](#)
[IsWorkingDir property](#)
[Italic property](#)
[Items property](#)

J

[Join property](#)
[JoinCorners property](#)
[JoinHeight property](#)
[JoinName property](#)
[JoinType property](#)
[JoinWidth property](#)
[Justifiable property](#)

K

[KeepWithNext property](#)
[KeepWithPrev property](#)
[KeyboardLanguage property](#)
[Keywords property](#)
[Kinsoku property](#)

L

[LandscapeMode property](#)

[Language property](#)
[Last property](#)
[LastChild property](#)
[LastCursorableDivision property](#)
[LastDivision property](#)
[LastDocPath property](#)
[LastEditorName property](#)
[LastGraphicPath property](#)
[LastGraphicType property](#)
[LastInsertSymbolFont property](#)
[LastMacroPath property](#)
[LastMessageID property](#)
[LastMessageStr property](#)
[LastMessageType property](#)
[LastName property](#)
[LastPage property](#)
[LastUsedDateFormula property](#)
[Layer property](#)
[LayerName property](#)
[Layout property](#)
[LayoutName property](#)
[LayoutOverride property](#)
[Layouts property](#)
[LeaderDotDashChar property](#)
[LeaderDotDotChar property](#)
[LeaderDotType property](#)
[LeaderDotUnderscoreChar property](#)
[LeadingText property](#)
[Left property](#)
[LeftBorder property](#)
[LeftExternalMargin property](#)
[LeftPage property](#)
[LeftTopCellId property](#)
[Length property](#)
[Level property](#)
[LineLocation property](#)
[LineMix property](#)
[LineNumberOptions property](#)
[LinePlacement property](#)
[Lines property](#)
[LinesSpacedEveryNthUnit property](#)
[LineValid property](#)
[LinkAvailable property](#)
[LinkDisplayName property](#)
[LinkDisplayNameFileLength property](#)
[Linked property](#)
[LinkedFileName property](#)
[LinkFrame property](#)
[LinkGraphic property](#)
[ListCount property](#)
[LoadFilesMaximized property](#)
[Location property](#)
[Locked property](#)
[LockLevel property](#)
[LockResult property](#)
[Locks property](#)
[LotusMiscStatus property](#)
[LowerCase property](#)

[LowerCaseAscent property](#)
[LwpMenuBar property](#)

M

[Macro property](#)
[MacroName property](#)
[MacroPath property](#)
[MacroPaths property](#)
[MacroStatus property](#)
[MailRouting property](#)
[MailRoutingPtr property](#)
[MaintainAspectRatio property](#)
[MaintainEditor property](#)
[MarginBetween property](#)
[MarginBottom property](#)
[MarginColor property](#)
[MarginLeft property](#)
[MarginRight property](#)
[MarginTop property](#)
[MarkCharacter property](#)
[MarkerClass property](#)
[MarkerName property](#)
[Markers property](#)
[MarkPosition property](#)
[MarkType property](#)
[Master property](#)
[MasterName property](#)
[MatchType property](#)
[MaxBottomBorder property](#)
[MaxBottomGutter property](#)
[MaxContentHeight property](#)
[MaxContentWidth property](#)
[MaxHorzPaneDistance property](#)
[MaxHyphLines property](#)
[MaxIdenticalConsecSentOpens property](#)
[MaxIdenticalSentOpensWithin10 property](#)
[MaximizeOnStartUp property](#)
[MaximumWordsinaSentence property](#)
[MaxLeftBorder property](#)
[MaxLeftGutter property](#)
[MaxNumColsAllowed property](#)
[MaxNumRowsAllowed property](#)
[MaxRightBorder property](#)
[MaxRightGutter property](#)
[MaxSplitCols property](#)
[MaxSplitRows property](#)
[MaxTopBorder property](#)
[MaxTopGutter property](#)
[MaxVertPaneDistance property](#)
[MenuPath property](#)
[MenuPaths property](#)
[MergeFileType property](#)
[MergeInfoPtr property](#)
[MergeOptions property](#)
[MergeStepNumber property](#)
[MergeToFile property](#)
[MetafilePict property](#)

[MinBottomMargin property](#)
[MinHeight property](#)
[MinLeftMargin property](#)
[MinRightMargin property](#)
[MinTopMargin property](#)
[MirrorName property](#)
[ModifiedDateString property](#)
[ModifiedTimeString property](#)
[ModifiedTimeValue property](#)
[MultiCellPasteOn property](#)
[MultiCompareParaTag property](#)
[MultiCompareParaTagSet property](#)

N

[Name property](#)
[NameBasedOnStyle property](#)
[NameForFilters property](#)
[NameOfExternalBorder property](#)
[NavigationLink property](#)
[Negative property](#)
[NewDocMacroName property](#)
[NewFile property](#)
[NewVersion property](#)
[NextClickHere property](#)
[NextCursorableDivision property](#)
[NextName property](#)
[NextNeighbor property](#)
[NextSection property](#)
[NextStyleName property](#)
[NextText property](#)
[NoHyphenate property](#)
[NonUserDocument property](#)
[Normal property](#)
[NormalParagraph property](#)
[Notelayouts property](#)
[NotesFlow property](#)
[Number property](#)
[NumberEveryNthLine property](#)
[NumberFound property](#)
[Numbering property](#)
[NumberingPosition property](#)
[NumberingStyleName property](#)
[NumberOfDataFields property](#)
[NumberOfLines property](#)
[NumberOfMergeConditions property](#)
[NumberOfReplacements property](#)
[NumberOfRevisions property](#)
[NumberSequenceName property](#)
[NumberWhichLines property](#)
[NumCharsInDoc property](#)
[NumCharsInParagraph property](#)
[NumCols property](#)
[NumColsSpannedOneCell property](#)
[NumContainers property](#)
[NumDecimalPlaces property](#)
[NumDisplayablePagesInDoc property](#)
[NumericFormat property](#)

[NumFields property](#)
[NumLinesOfSpace property](#)
[NumLinesOfSpaceAbove property](#)
[NumLinesOfSpaceAboveLine property](#)
[NumLinesOfSpaceBelow property](#)
[NumOfRecentFiles property](#)
[NumPagesInDoc property](#)
[NumParagraphs property](#)
[NumRows property](#)
[NumRowsSpannedOneCell property](#)
[NumRowsThatFit property](#)
[NumTabs property](#)
[NumUndoLevels property](#)
[NumWindowsViewingDoc property](#)
[NumWordsInDoc property](#)

O

[Object property](#)
[ObjectType property](#)
[Oblique property](#)
[OCXDesignMode property](#)
[Ole1Object property](#)
[OleContainer property](#)
[OleContainerDocName property](#)
[OleEmbeddedSize property](#)
[OLEEnabled property](#)
[OleMinHeight property](#)
[OleObject property](#)
[OleObjects property](#)
[OleObjectSize property](#)
[OpenDocMacroName property](#)
[OpenDocsVisible property](#)
[OpenExisingFileInWelcomeBox property](#)
[OpenReadOnly property](#)
[Options property](#)
[OrigFileType property](#)
[OrigHeight property](#)
[OrigWidth property](#)
[OStype property](#)
[Outline property](#)
[OutlineBorderLines property](#)
[OutlineButtons property](#)
[OutlineHeadingButtonsOnly property](#)
[OutlineLevel property](#)
[OutlineOnlyHeadingsWhenCollapsed property](#)
[OutlineSeqItems property](#)
[OutlineStyleSequences property](#)
[OutputToFile property](#)
[Override property](#)
[OverrideCheckAndEnable property](#)
[Overstrike property](#)
[OverstrikeCharacter property](#)

P

[Page property](#)
[PageNo property](#)

[PageNum property](#)
[PageNumber property](#)
[PageNumberAsText property](#)
[PageNumFirstPageShowing property](#)
[PageNumLastPageShowing property](#)
[PageOrder property](#)
[Pages property](#)
[PageStyleName property](#)
[PageStyles property](#)
[PageToUseLayoutOn property](#)
[PairKerning property](#)
[PaperName property](#)
[PaperNames property](#)
[ParagraphBackground property](#)
[ParagraphBorder property](#)
[ParagraphHasDropCap property](#)
[ParagraphHasText property](#)
[ParagraphStyle property](#)
[ParagraphStyleName property](#)
[ParagraphStyles property](#)
[ParagraphSymbolChar property](#)
[ParallelColumns property](#)
[Parent property](#)
[ParentMenuHWND property](#)
[ParentName property](#)
[Partial property](#)
[Path property](#)
[PathName property](#)
[Pattern property](#)
[Percentage property](#)
[PersonalData1 property](#)
[PersonalData2 property](#)
[PersonalData3 property](#)
[PersonalData4 property](#)
[PhoneNumber property](#)
[PitchAndFamily property](#)
[Placement property](#)
[Plain property](#)
[PointSize property](#)
[Position property](#)
[PositionType property](#)
[PositionXInContainer property](#)
[PositionXOnPage property](#)
[PositionYInContainer property](#)
[PositionYOnPage property](#)
[PowerField property](#)
[PowerFields property](#)
[Preferences property](#)
[Prefix property](#)
[Presentation property](#)
[PreviousClickHere property](#)
[PreviousCursorableDivision property](#)
[PreviousName property](#)
[PreviousNeighbor property](#)
[PrintDestination property](#)
[PrinterName property](#)
[PrintGraphics property](#)
[PrintManager property](#)

[PrintPagesFrom property](#)
[PrintPagesTo property](#)
[PrintPageType property](#)
[PrintRange property](#)
[PrintSettings property](#)
[Private property](#)
[ProductVersion property](#)
[ProgID property](#)
[Prompt property](#)
[PromptHidden property](#)
[ProtectedMode property](#)

Q

[QueueName property](#)

R

[ReadCompressed property](#)
[ReadOnly property](#)
[Recursive property](#)
[Red property](#)
[Relative property](#)
[RelativeIndent property](#)
[RelativePageNum property](#)
[RelativeType property](#)
[RelativeXDistance property](#)
[RelativeYDistance property](#)
[ReleaseNumber property](#)
[RenderedPageNumber property](#)
[ReplaceAttributes property](#)
[ReplaceExactCase property](#)
[ReplaceFont property](#)
[ReplaceLanguage property](#)
[Replacements property](#)
[ReplaceString property](#)
[ReplaceStyleName property](#)
[ReplaceWithProperties property](#)
[RequestRemarkOnClose property](#)
[RequireStartupScripts property](#)
[ResetOnEachPage property](#)
[ResetWhen property](#)
[Rest property](#)
[RestartStyleName property](#)
[Result property](#)
[ResumePausedMacro property](#)
[RetainNameOfImportedFile property](#)
[ReturnExits property](#)
[ReviewVersions property](#)
[RevisionDisplay property](#)
[RevisionMark property](#)
[RevisionMarkMode property](#)
[RevisionType property](#)
[RevMarkCharacter property](#)
[RevMarkPosition property](#)
[RevMarkType property](#)
[Right property](#)
[RightAlign property](#)

[RightBorder property](#)
[RightExternalMargin property](#)
[RightMouseMenus property](#)
[RightMousePropId property](#)
[RightMousePropText property](#)
[RightPage property](#)
[RowLayouts property](#)
[Rows property](#)
[RubyLayouts property](#)
[RunMacroOnDocEvents property](#)
[RunMacroOnLoad property](#)
[RunOnCloseDoc property](#)
[RunOnNewDoc property](#)
[RunOnOpenDoc property](#)

S

[Saved property](#)
[SaveDivisions property](#)
[SavePath property](#)
[SaveSections property](#)
[SaveSnapShot property](#)
[ScaleHeight property](#)
[ScaleMode property](#)
[ScalePercentage property](#)
[ScaleWidth property](#)
[ScreenPositionX property](#)
[ScreenPositionY property](#)
[SearchAttributes property](#)
[SearchLanguage property](#)
[Section property](#)
[SectionName property](#)
[Sections property](#)
[SectionTabs property](#)
[SectionUserName property](#)
[SelectedPages property](#)
[SelectionBorderColor1 property](#)
[SelectionBorderColor2 property](#)
[SelectionBorderColor3 property](#)
[SelectionHidden property](#)
[SelectionType property](#)
[SelectTab property](#)
[SelectType property](#)
[SequenceName property](#)
[SequenceNumber property](#)
[ServerFormat property](#)
[SetContextOfBar property](#)
[SetTabsDialog property](#)
[Shadow property](#)
[Shape property](#)
[ShowBubbleHelp property](#)
[ShowDivisionTabs property](#)
[ShowExportWarningMessages property](#)
[ShowFileNew property](#)
[ShowGraphicPreview property](#)
[ShowHeaderFooterBar property](#)
[ShowHiddenText property](#)
[ShowInContext property](#)

[ShowMailDisabled property](#)
[ShowMarks property](#)
[ShowMenuCustomization property](#)
[ShowMenuWarningMessage property](#)
[ShowNoWelcomeBox property](#)
[ShowStatistics property](#)
[ShowTabs property](#)
[SilverBullet property](#)
[SilverBullets property](#)
[SingleCellSelected property](#)
[Size property](#)
[SizeStyleName property](#)
[SizingUnitName property](#)
[SizingUnits property](#)
[Skipped property](#)
[SkipWordMode property](#)
[SmallCaps property](#)
[SmallFileFormat property](#)
[SmartCorrect property](#)
[SmartCorrects property](#)
[SmartFill property](#)
[SmartLabelsEnabled property](#)
[SmartLevel property](#)
[SnapShotSaveOptions property](#)
[SortLevel1 property](#)
[SortLevel2 property](#)
[SortLevel3 property](#)
[SortNumbers property](#)
[SortOptions property](#)
[SortOrder property](#)
[SortType property](#)
[SortWord property](#)
[SortWordOption property](#)
[SpaceAbove property](#)
[SpaceBelow property](#)
[SpacesBetweenSentences property](#)
[Spacing property](#)
[SpacingStyleName property](#)
[SpacingUnitName property](#)
[SpacingUnits property](#)
[Span property](#)
[SpellCheckInitialCaps property](#)
[SpellCheckRepeatedWords property](#)
[SpellCheckUserDictAlternatives property](#)
[SpellCheckWordsWithNums property](#)
[SpellColor property](#)
[SpellFocusedColor property](#)
[SplitPercentage property](#)
[Start property](#)
[StartColumns property](#)
[StartingColOfSelection property](#)
[StartingColStringOfSelection property](#)
[StartingNumber property](#)
[StartingRowOfSelection property](#)
[StartRow property](#)
[StateID property](#)
[StatusBar property](#)
[StatusBarButtons property](#)

[StatusBarVisible property](#)
[StatusSpellReplaceAll property](#)
[StrikeThrough property](#)
[Style property](#)
[StyleExceptions property](#)
[StyleName property](#)
[StylePath property](#)
[StylePaths property](#)
[StyleSheetFullPath property](#)
[StyleSheetName property](#)
[StyleSheetPath property](#)
[Subscript property](#)
[Suffix property](#)
[Suggestions property](#)
[Superscript property](#)
[SuperTableContainer property](#)
[SuperTableLayouts property](#)
[SuperTables property](#)
[SuppressHeaders property](#)

T

[TabExits property](#)
[Table property](#)
[TableContainer property](#)
[TableExports property](#)
[TableFill property](#)
[TableHeadingLayouts property](#)
[TableHeadings property](#)
[TableImports property](#)
[TableLayouts property](#)
[TableLine property](#)
[TableMarkers property](#)
[TableOfContents property](#)
[TableOnlyContainer property](#)
[Tables property](#)
[TableStyleName property](#)
[TableStyles property](#)
[TabOrder property](#)
[TabRack property](#)
[TabRelativeTo property](#)
[TabSpacing property](#)
[TabSymbolChar property](#)
[TabType property](#)
[TempFindAndReplace property](#)
[TempFoundry property](#)
[Text property](#)
[TextAfter property](#)
[TextandTableExports property](#)
[TextandTableImports property](#)
[TextAttributes property](#)
[TextBefore property](#)
[TextMarkers property](#)
[TextMarkersInSelection property](#)
[TextOrient property](#)
[TextOrientation property](#)
[Texts property](#)
[TextStyleName property](#)

[TextStyles property](#)
[TextTightness property](#)
[TextViewAttributes property](#)
[ThesMaxNumSynonymsReturned property](#)
[Tile property](#)
[TileWindow property](#)
[Time property](#)
[TimedSaveFileExtension property](#)
[Title property](#)
[TitleBarDocNumber property](#)
[TitleBarVisible property](#)
[TOCRange property](#)
[TOCSource property](#)
[Top property](#)
[TopBorder property](#)
[TopExternalMargin property](#)
[TopLeftCellRowId property](#)
[TopLevelMenuText property](#)
[TotalEditingTime property](#)
[TrailingText property](#)
[Type property](#)
[TypeAbove property](#)
[TypeAboveLine property](#)
[TypeBelow property](#)
[TypeBetween property](#)
[TypeRight property](#)
[Typos property](#)

U

[UIActive property](#)
[Underline property](#)
[UndoLevels property](#)
[Units property](#)
[UnknownTagsInCommentNotes property](#)
[UpdateFields property](#)
[UpdateOnLoad property](#)
[UpdateOnLoadImmediate property](#)
[UpperCase property](#)
[URLAddress property](#)
[URLDescription property](#)
[URLDownloadComplete property](#)
[UseConsistentSpaceBetweenSentences property](#)
[UseContents property](#)
[UseCycleKeys property](#)
[UseDefaultPrinter property](#)
[UsedFirstPageHeight property](#)
[UseFindStyle property](#)
[UseFooter property](#)
[UseGreeting property](#)
[UseHeader property](#)
[UseNextStyle property](#)
[UsePrinterSettings property](#)
[UserClassNameApp property](#)
[UserClassNameFull property](#)
[UserClassNameShort property](#)
[UserDefinedFilter property](#)
[UserDictFiles property](#)

[UserDictionaryFiles property](#)
[UserDictionaryPath property](#)
[UserDictionaryPaths property](#)
[UserDictStates property](#)
[UseRelative property](#)
[UseReplaceStyle property](#)
[UserInitials property](#)
[UserInterfacePrefs property](#)
[UserLinkName property](#)
[UserName property](#)
[UseSeparatorLine property](#)
[UsesHelp property](#)
[UsesPalette property](#)
[UseSuperscriptReferenceNum property](#)
[UseTwoSpacesBetweenSentences property](#)
[UseWhen property](#)

V

[Valid property](#)
[VersionID property](#)
[VersionManager property](#)
[VersionName property](#)
[Versions property](#)
[VertAlign property](#)
[VerticalSplitWindow property](#)
[VertRuler property](#)
[VertScrollBarVisible property](#)
[ViewLevel property](#)
[ViewType property](#)
[Visible property](#)

W

[WasDeletedInRevMarkMode property](#)
[WasInsertedInRevMarkMode property](#)
[WaterMarkName property](#)
[WatermarksPath property](#)
[Weight property](#)
[Where property](#)
[WidowOrphan property](#)
[Width property](#)
[WidthAbove property](#)
[WidthBelow property](#)
[WidthBetween property](#)
[WidthInTwips property](#)
[WindowId property](#)
[WindowsName property](#)
[WinViewPrefs property](#)
[WithComments property](#)
[WordDoubleUnderline property](#)
[WordUnderline property](#)
[WorkingType property](#)
[WPDataSets property](#)
[WrapType property](#)

X

XOffset property
XPosition property

Y

YOffset property
YPosition property

Z

Zero property
ZipCode property

Help is not available

The Word Pro help files are not installed on your system.

If you choose the Minimum installation, help is NOT installed. You can use the Custom installation option to install the help files.

