Word 2004 AppleScript Reference
Using the Word Dictionary

About the Word AppleScript Dictionary ................................................................. 15
Referring to the active element ................................................................................ 16
Working with document objects .............................................................................. 16
Working with the selection object ............................................................................ 18
Working with text range objects .............................................................................. 18
Assigning text ranges ............................................................................................. 20
Returning text from a document .............................................................................. 21
Applying formatting to text .................................................................................... 22
Editing text ................................................................................................................ 24
Finding and replacing text or formatting ................................................................ 25
Selecting text in a document .................................................................................... 26
Inserting text in a document .................................................................................... 26
Working with tables .................................................................................................. 27

Microsoft Office Suite Classes

assistant ..................................................................................................................... 29
balloon ...................................................................................................................... 32
balloon checkbox ..................................................................................................... 36
balloon label ............................................................................................................ 36
command bar .......................................................................................................... 37
command bar button ............................................................................................... 41
command bar combobox ......................................................................................... 42
command bar control ............................................................................................... 43
command bar popup ................................................................................................ 46
custom document property ..................................................................................... 46
document property .................................................................................................. 46
web page font .......................................................................................................... 48

Microsoft Office Suite Commands

add item to combobox ............................................................................................ 49
clear combobox ....................................................................................................... 49
execute .................................................................................................................... 50
get combobox item ................................................................................................ 50
get count of combobox items .................................................................................. 51
remove an item from combobox ............................................................................. 51
reset ....................................................................................................................... 52
show balloon .......................................................................................................... 52
Microsoft Word Suite Commands

accept ............................................................................................................................................. 216
accept all revisions ...................................................................................................................... 216
activate object .................................................................................................................................. 217
append to spike ............................................................................................................................ 217
apply bullet default ....................................................................................................................... 218
apply list format template ............................................................................................................. 218
apply list template ......................................................................................................................... 219
apply number default ..................................................................................................................... 220
apply page borders to all sections .................................................................................................. 220
apply theme ................................................................................................................................... 221
auto format ..................................................................................................................................... 221
auto summarize .............................................................................................................................. 222
automatic change .......................................................................................................................... 222
break link ................................................................................................................................------- 223
build key code ................................................................................................................................ 223
calculate selection ......................................................................................................................... 224
can continue previous list ............................................................................................................... 224
centimeters to points ...................................................................................................................... 225
change file open directory ............................................................................................................. 225
check ............................................................................................................................................... 226
check consistency ............................................................................................................................ 226
check grammar ............................................................................................................................... 226
check spelling .................................................................................................................................. 227
clean string ..................................................................................................................................... 227
clear ............................................................................................................................................... 228
clear all fuzzy options .................................................................................................................... 229
clear formatting .............................................................................................................................. 230
click object ...................................................................................................................................... 231
close print preview .......................................................................................................................... 231
collapse outline .............................................................................................................................. 232
compare .......................................................................................................................................... 232
compute statistics ........................................................................................................................... 233
correct ............................................................................................................................................. 233
correct text to numbers .................................................................................................................. 234
get webpage font ................................................................. 273
get zoom ................................................................. 274
grow font ................................................................. 274
home key ................................................................. 275
inches to points ............................................................. 275
insert ................................................................. 276
insert auto text ............................................................. 276
insert auto text entry ...................................................... 277
insert break ................................................................. 277
insert caption ................................................................. 278
insert cells ................................................................. 278
insert columns ............................................................. 279
insert cross reference .................................................. 279
insert database ............................................................ 280
insert date time ............................................................ 282
insert envelope data ..................................................... 283
insert file ................................................................. 284
insert paragraph .......................................................... 285
insert rows ................................................................. 286
insert symbol ............................................................... 286
key string ................................................................. 287
large scroll ................................................................. 288
lines to points ............................................................. 289
list commands ............................................................. 289
list indent ................................................................. 289
list outdent ............................................................... 290
make compatibility default .............................................. 290
make new data merge ask field ......................................... 291
make new data merge fill in field ........................................ 292
make new data merge if field ............................................. 292
make new data merge next field ........................................ 293
make new data merge next if field ....................................... 294
make new data merge rec field ......................................... 294
make new data merge sequence field ................................... 295
make new data merge set field ............................................. 295
make new data merge skip if field ....................................... 296
manual hyphenation .......................................................... 296
mark entry for table of contents ......................................... 297
mark entry for table of figures ............................................ 297
mark for index ............................................................. 298
merge ................................................................. 299
merge subdocuments ..................................................... 299
millimeters to points ...................................................... 299
modified ................................................................. 300
next for browser .......................................................... 300
next header footer .......................................................... 301
next revision .............................................................. 301
on time ................................................................. 301
open as document .......................................................... 302
open data source .......................................................... 303
open document from workmenu ........................................ 304
open header source ....................................................... 304
open recent file ............................................................ 305
open subdocument ........................................................ 305
open version ............................................................. 306
organizer copy ............................................................. 306
Drawing Suite Classes

callout........................................................................................................... 353

callout format............................................................................................. 354

fill format................................................................................................... 357

horizontal line format................................................................................ 362

inline horizontal line.................................................................................. 363

inline picture .............................................................................................. 363

inline picture bullet.................................................................................... 364

inline shape.................................................................................................. 365

line format.................................................................................................... 367

line shape..................................................................................................... 367

picture.......................................................................................................... 372

picture format............................................................................................. 373

shadow format............................................................................................ 375

shape............................................................................................................. 376

standard inline horizontal line................................................................. 382

set document compatibility....................................................................... 335

set number to text columns........................................................................ 338

set private profile string............................................................................ 338

set profile string........................................................................................ 338

show.............................................................................................................. 339

show all headings....................................................................................... 339

show heading............................................................................................... 340

show me....................................................................................................... 340

shrink discontinuous selection.................................................................. 340

shrink font.................................................................................................... 340

shrink selection........................................................................................... 341

small scroll.................................................................................................. 341

speak text....................................................................................................... 342

split subdocument ..................................................................................... 342

split table in selection.............................................................................. 343

substitute font............................................................................................. 343

swap with endnotes................................................................................... 343

swap with footnotes.................................................................................. 344

toggle portrait............................................................................................ 344

type backspace........................................................................................... 344

type paragraph........................................................................................... 344

type text........................................................................................................ 345

undo.............................................................................................................. 345

undo clear.................................................................................................... 346

unlink.......................................................................................................... 346

unprotect...................................................................................................... 346

update.......................................................................................................... 347

update document....................................................................................... 348

update field.................................................................................................. 349

update page numbers................................................................................ 349

update source............................................................................................ 349

update styles............................................................................................... 350

update summary properties....................................................................... 350

use address book......................................................................................... 351

use default folder suffix........................................................................... 351

view property browser............................................................................... 351

web page preview....................................................................................... 352

Word help................................................................................................... 352
Drawing Suite Commands

activate object........................................................................................................... 395
apply ........................................................................................................................... 396
automatic length...................................................................................................... 396
break forward link................................................................................................ 397
convert to frame...................................................................................................... 398
convert to inline shape........................................................................................... 398
custom drop ............................................................................................................ 399
custom shape.......................................................................................................... 399
flip ............................................................................................................................ 400
get border ............................................................................................................. 400
one color gradient................................................................................................. 401
patterned ................................................................................................................ 401
pick up ..................................................................................................................... 402
preset drop ........................................................................................................... 402
preset gradient ..................................................................................................... 403
preset textured ..................................................................................................... 404
rout connections ................................................................................................... 405
reset ....................................................................................................................... 405
reset rotation ........................................................................................................ 406
scale height .......................................................................................................... 406
scale width ............................................................................................................ 406
set extrusion direction ......................................................................................... 407
set shapes default properties .............................................................................. 407
set threeD format ................................................................................................. 408
solid ....................................................................................................................... 409
toggle vertical text ............................................................................................... 409
two color gradient ............................................................................................... 410
user picture ........................................................................................................... 410
user textured ......................................................................................................... 411
valid link target ..................................................................................................... 412
z order ................................................................................................................... 412

Text Suite Classes

character .................................................................................................................. 413
grammatical error ................................................................................................. 414
paragraph ............................................................................................................... 414
paragraph format .................................................................................................. 420
section .................................................................................................................... 425
sentence ............................................................................................................... 426
shading ................................................................................................................... 427
spelling error ....................................................................................................... 429
text range ............................................................................................................. 430
word ....................................................................................................................... 430
Word style .............................................................................................................. 439
Text Suite Commands

auto format text range .............................................................................................................. 444
calculate range .......................................................................................................................... 444
change end of range .................................................................................................................. 445
change start of range ............................................................................................................... 445
check grammar ....................................................................................................................... 446
check spelling ......................................................................................................................... 446
check synonyms ..................................................................................................................... 447
close up .................................................................................................................................. 448
collapse range ......................................................................................................................... 448
compute statistics .................................................................................................................. 449
convert to table ....................................................................................................................... 449
copy as picture ....................................................................................................................... 451
copy object ............................................................................................................................. 451
cut object ................................................................................................................................. 451
expand ..................................................................................................................................... 451
get border ................................................................................................................................. 452
get footer ................................................................................................................................. 452
get header ................................................................................................................................ 453
get range information ............................................................................................................. 454
going to next ............................................................................................................................ 458
going to previous ..................................................................................................................... 458
in range .................................................................................................................................... 459
in story ....................................................................................................................................... 459
indent ....................................................................................................................................... 460
indent char width ...................................................................................................................... 460
indent first line char width ........................................................................................................ 461
is equivalent ............................................................................................................................... 461
link to list template .................................................................................................................. 462
merge subdocuments .............................................................................................................. 462
modify enclosure ...................................................................................................................... 463
move end of range .................................................................................................................... 463
move range ................................................................................................................................ 463
move range end until .................................................................................................................. 465
move range end while .............................................................................................................. 465
move range start until .............................................................................................................. 466
move range start while .............................................................................................................. 466
move range until ....................................................................................................................... 467
move range while ..................................................................................................................... 467
move start of range .................................................................................................................. 468
navigate ..................................................................................................................................... 468
next paragraph ......................................................................................................................... 469
next range .................................................................................................................................. 470
next subdocument ................................................................................................................. 470
open or close up ......................................................................................................................... 470
open up ...................................................................................................................................... 470
outdent ...................................................................................................................................... 471
outline demote to body ............................................................................................................. 471
outline promote ......................................................................................................................... 472
paste and format ...................................................................................................................... 473
paste append table .................................................................................................................... 474
paste as nested table ................................................................................................................ 474
paste excel table ...................................................................................................................... 474
paste object .......................................................................................................................................................... 475
paste special .......................................................................................................................................................... 475
previous paragraph ............................................................................................................................................... 476
previous range ..................................................................................................................................................... 477
previous subdocument .......................................................................................................................................... 477
relocate ............................................................................................................................................................... 478
reset ................................................................................................................................................................... 478
set range ............................................................................................................................................................. 479
sort ascending ..................................................................................................................................................... 479
sort ...................................................................................................................................................................... 480
sort descending .................................................................................................................................................. 482
space 1 ............................................................................................................................................................... 482
space 15 ............................................................................................................................................................... 482
space 2 ............................................................................................................................................................... 483
tab hanging indent ............................................................................................................................................... 483
tab indent ............................................................................................................................................................. 483
text range spelling suggestions .......................................................................................................................... 484

Proofing Suite Classes

autocorrect ......................................................................................................................................................... 486
autocorrect entry ............................................................................................................................................... 488
dictionary ............................................................................................................................................................ 489
first letter exception ........................................................................................................................................... 491
language .............................................................................................................................................................. 491
other corrections exception .................................................................................................................................. 492
readability statistic ............................................................................................................................................. 495
synonym info ..................................................................................................................................................... 496
two initial caps exception .................................................................................................................................. 498

Proofing Suite Commands

apply correction ..................................................................................................................................................... 499
get synonym list for ........................................................................................................................................... 499
get synonym list from .......................................................................................................................................... 500

Table Suite Classes

cell ................................................................................................................................................................. 501
column .............................................................................................................................................................. 504
column options ................................................................................................................................................... 505
row ................................................................................................................................................................. 506
row options ....................................................................................................................................................... 508
table ............................................................................................................................................................... 511
### Table Suite Commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>auto fit</td>
<td>514</td>
</tr>
<tr>
<td>auto fit behavior</td>
<td>515</td>
</tr>
<tr>
<td>auto format table</td>
<td>515</td>
</tr>
<tr>
<td>auto sum</td>
<td>516</td>
</tr>
<tr>
<td>convert row to text</td>
<td>517</td>
</tr>
<tr>
<td>convert to text</td>
<td>517</td>
</tr>
<tr>
<td>distribute row height</td>
<td>518</td>
</tr>
<tr>
<td>distribute width</td>
<td>518</td>
</tr>
<tr>
<td>formula</td>
<td>518</td>
</tr>
<tr>
<td>get border</td>
<td>519</td>
</tr>
<tr>
<td>get cell from table</td>
<td>520</td>
</tr>
<tr>
<td>merge cell</td>
<td>520</td>
</tr>
<tr>
<td>set left indent</td>
<td>521</td>
</tr>
<tr>
<td>set table item height</td>
<td>521</td>
</tr>
<tr>
<td>set table item width</td>
<td>522</td>
</tr>
<tr>
<td>sort ascending</td>
<td>523</td>
</tr>
<tr>
<td>sort descending</td>
<td>523</td>
</tr>
<tr>
<td>split cell</td>
<td>524</td>
</tr>
<tr>
<td>split table</td>
<td>524</td>
</tr>
<tr>
<td>table sort</td>
<td>524</td>
</tr>
<tr>
<td>update auto format</td>
<td>526</td>
</tr>
</tbody>
</table>
### About the Word AppleScript Dictionary

The Microsoft Word AppleScript Dictionary provides comprehensive access to the classes and commands you need for programmatically controlling Word by using AppleScript. The Word AppleScript Dictionary is based on the Visual Basic object model for Word, and as such its terminology and syntax closely mirror the terminology and syntax used in Visual Basic. As a result of this close relationship with Visual Basic, the Word AppleScript Dictionary might appear more complex and less "English-like" than other AppleScript Dictionaries.

#### Using classes and commands

Classes and commands in the Word AppleScript Dictionary often have many properties and parameters. Many of these properties and parameters are represented by enumerations — or lists — of constants rather than text or numbers or Boolean operators. These enumerations use many of the same words as their parameters or properties and are usually descriptive of what they represent. Because of the nature of the AppleScript language and the complexity of the underlying object model, expressions containing several properties and parameters can be very long and often look confusing. In such cases, it is helpful to use parentheses to separate out one property from another to make your scripts easier to read. Additionally, you will find that the Word object model is very deep, requiring several layers to access certain objects. You might find it helpful to use variables to store these objects and reduce the complexity of your script.

As in other applications, optional command parameters are denoted in the dictionary by square brackets [ ]. Class properties always have default values, which means that when you create new objects it is not necessary to supply values for all of the properties. Properties denoted by [r/o] are read-only and cannot be set once an object exists. However, when you use the make command to create new objects, most read-only properties can be set by using the with properties clause. For more information about the properties available for various objects, see the object’s entry in the Dictionary Reference portion of this guide.

---

Using the Word Dictionary

<table>
<thead>
<tr>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>15</td>
</tr>
<tr>
<td>15</td>
</tr>
<tr>
<td>16</td>
</tr>
<tr>
<td>16</td>
</tr>
<tr>
<td>18</td>
</tr>
<tr>
<td>18</td>
</tr>
<tr>
<td>18</td>
</tr>
<tr>
<td>20</td>
</tr>
<tr>
<td>21</td>
</tr>
<tr>
<td>22</td>
</tr>
<tr>
<td>24</td>
</tr>
<tr>
<td>25</td>
</tr>
<tr>
<td>26</td>
</tr>
<tr>
<td>26</td>
</tr>
<tr>
<td>27</td>
</tr>
</tbody>
</table>

---

---
Using the Word Dictionary

**Referring to the active element**

To refer to the selected, or highlighted, paragraph, table, field, or other document element, use the selection object. From the selection object, you can access all paragraphs and other elements in the selection. The following example applies a border around the first paragraph in the selection.

```applescript
set enable borders of border options of paragraph 1 of selection to true
```

The following example applies a border around all the paragraphs in the selection.

```applescript
set enable borders of border options of paragraph format of selection to true
```

The following example applies shading to the first row of the first table in the selection.

```applescript
set texture of shading of row 1 of table 1 of selection to texture10 percent
```

An error occurs if the selection doesn't include a table. To determine whether the selection includes a table, use the `count` property. The following example applies shading to the first row of the first table in the selection.

```applescript
if (count of tables of selection) ≥ 1 then
    set texture of shading of row 1 of table 1 of selection to texture10 percent
else
    display dialog "Selection doesn't include a table"
end if
```

The following example applies shading to the first row of every table in the selection. The Repeat With...In loop is used to step through the individual tables in the selection.

```applescript
if (count of tables of selection) ≥ 1 then
    repeat with aTable in (get tables of selection)
        set texture of shading of row 1 of aTable to texture10 percent
    end repeat
end if
```

**Working with document objects**

In AppleScript, the commands for modifying files are commands of the document class.

**Creating a new document**

The documents list includes all of the open documents. To create a new document, use the make command to add a document object to the documents list. The following instruction creates a new document.

```applescript
make new document
```
Using the Word Dictionary

A better way to create a new document is to assign the return value to an object variable. The `make` command returns a `document` object that refers to the new document. In the following example, the document object returned by the `make` command is assigned to an object variable, `newDoc`. Then several properties and commands of the `document` object are set. You can easily control the new document using the `newDoc` object variable.

```plaintext
set newDoc to make new document
set name of font object of text object of newDoc to "Arial"
save as newDoc file name "Sample.doc"
```

**Opening a document**

To open an existing document, use the `open` command. The following instruction opens a document named "MyDocument.doc" located in the folder named "Shared."

```plaintext
```

**Saving an existing document**

To save a single document, use the `save` command. The following instruction saves the document named `Sales.doc`.

```plaintext
save document "Sales.doc"
```

You can save all open documents by applying the `save` command to the `documents` list. The following instruction saves all open documents.

```plaintext
save documents
```

**Saving a new document**

To save a new document, use the `save as` command. The following instruction saves the active document as "Temp.doc" in the current folder.

```plaintext
save as active document file name "Temp.doc"
```

The `file name` argument can include only the file name or the complete path.

**Closing documents**

To close a single document, use the `close` command. The following instruction closes and saves the document named `Sales.doc`.

```plaintext
close document "Sales.doc" saving yes
```

You can close all open documents by applying the `close` method to the `documents` list. The following instruction closes all documents without saving changes.

```plaintext
close documents saving no
```

**Activating a document**

To change the active document, use the `activate object` command. The following instruction activates the open document named "MyDocument.doc".

```plaintext
activate
activate object document "MyDocument.doc"
```

The `activate object` command only works if Word is the frontmost application. Use the `activate` command to ensure that Word is at the forefront.
Using the Word Dictionary

Referring to the active document

Instead of referring to a document by name or index number — for example document "Sales.doc" — the active document property of the application class returns a document object which refers to the active document (the document in the forefront). The following example displays the name of the active document, or if no documents are open, displays a message.

```applescript
if (count of documents) ≥ 1 then
    display dialog (get name of active document)
else
    display dialog "No documents are open"
end if
```

Working with the selection object

When you work on a document in Word, you usually select text and then perform an action, such as formatting the text or typing text. In a script, it is usually not necessary to select text before modifying the text. Instead, you create a text range object that refers to a specific portion of the document. For information about defining text range objects, see Working with text range objects. However, when you want your code to respond to or change the selection, you can do so with the selection object.

The select command selects or highlights an object. For example, the following instruction selects the first word in the active document.

```applescript
select word 1 of active document
```

For more select command examples, see Selecting text in a document.

The selection property of the application class returns a selection object that represents the selection in the active document window pane. The selection property of the window class represents the selection of the window. Word maintains the selection object for a window even if the window is not active. There can be only one selection object per document window pane and only one selection object can be active. If there is no highlighted selection, the selection object represents the insertion point. For example, the following example changes the paragraph formatting of the paragraphs in the selection.

```applescript
set left indent of paragraph format of selection to ¬ inches to points inches 0.5
```

The following example applies bold formatting to the selected text.

```applescript
set bold of font object of selection to true
```

Working with text range objects

A common use of AppleScript is to specify an area in a document and then do something with it, such as insert text or apply formatting. For example, you may want to write a script that locates a word or phrase within a portion of a document. The portion of the document can be represented by a text range object. After the text range object is identified, commands and properties of the text range object can be applied in order to modify the contents of the text range.
A text range object refers to a contiguous area in a document. Each text range object is defined by a starting and ending character position. Similar to the way bookmarks are used in a document, text range objects are used in procedures to identify specific portions of a document. A text range object can be as small as the insertion point or as large as the entire document. However, unlike a bookmark, a text range object exists only while the procedure that defined it is running.

The start of content, end of content, and story type properties uniquely identify a text range object. The start of content and end of content properties return or set the starting and ending character positions of the text range object. The character position at the beginning of the document is 0, the position after the first character is 1, and so on. There are 11 different story types represented by the constants of the story type property.

Note text range objects are independent of the selection. That is, you can define and modify a text range without changing the current selection. You can also define multiple text ranges in a document, while there is only one selection per document pane.

Using the create range command

The create range command is used to create a text range object in the specified document. The create range command returns a text range object located in the main story given a start and end point. The following example creates a text range object that is assigned to the variable MyRange.

set myRange to create range active document start 0 end 10

MyRange refers to the first 10 characters in the active document. You can see that the text range object has been created when you apply a property or command to the text range object stored in the MyRange variable. The following example applies bold formatting to the first 10 characters in the active document.

set myRange to create range active document start 0 end 10
set bold of myRange to true

When you need to refer to a text range object multiple times, you can set a variable equal to the text range object. However, if you need to perform only a single action on a text range object, there's no need to store the object in a variable. The same results can be achieved using just one instruction that identifies the text range and changes the bold property.

set bold of (create range active document start 0 end 10) to true

Like a bookmark, a text range can span a group of characters or mark a location in a document. The text range object in the following example has the same starting and ending points. The text range does not include any text. The following example inserts text at the beginning of the active document.

set myRange to create range active document start 0 end 0
insert text "Hello " at myRange

You can define the beginning and end points of a text range using the character position numbers as shown above, or use the start of content and end of content properties of the text range. The following example creates a text range object beginning at the start of the second paragraph and ending after the third paragraph.

set myDoc to active document
set myRange to create range myDoc start (start of content of paragraph 2 of myDoc) end (end of content of paragraph 3 of myDoc)
Using the text object property

The text object property appears on multiple objects, such as paragraph, bookmark, and cell, and is used to return a text range object. The following example returns a text range object that refers to the first paragraph in the active document.

set myRange to text object of paragraph 1 of active document

After you have a text range object, you can use any of its properties or commands to modify the text range object. The following example selects the second paragraph in the active document.

select text object of paragraph 2 of active document

Redefining a text range object

Use the set range command to redefine an existing text range object. Unlike in Visual Basic for Applications, the set range command does not modify the original range. The set range command creates a new text range from the existing range. You can assign a variable to an existing text range, and then reassign the variable name to the set range command. The following example defines myRange to the first sentence in the document. The set range command redefines myRange so that it refers to the first sentence plus the next 10 characters.

set myRange to sentence 1 of active document
set myRange to set range myRange start (start of content of myRange) ¬
        end ((end of content of myRange) + 10)

Note When debugging your scripts, you can use the select command to ensure that a text range object is referring to the correct range of text. For example, the following example selects the text range object named aRange. The aRange object refers to the second and third paragraphs in the active document.

set aRange to sentence 2 of active document
set aRange to set range aRange start (start of content of aRange) ¬
        end (end of content of sentence 3 of active document)
select aRange

Assigning text ranges

There are several ways to assign an existing text range object to a variable. This topic explains the results of two different techniques. In the following examples, the Range1 and Range2 variables refer to text range objects. For example, the following instructions assign the first and second words in the active document to the Range1 and Range2 variables.

set Range1 to word 1 of active document
set Range2 to word 2 of active document
Setting a text range object variable equal to another text range object variable

The following instruction assigns a range variable named Range2 to represent the same location as Range1.

set Range2 to Range1

You now have two variables that represent the same range. When you manipulate the start or end point or the text of Range2, it affects Range1 and vice versa.

Note that the following instruction is not the same as the preceding instruction. This instruction assigns the content property of Range1 to the content property of Range2. It doesn't change what the objects actually refer to.

set content of Range2 to content of Range1

The ranges (Range2 and Range1) have the same contents, but they may point to different locations in the document or even to different documents.

Returning text from a document

Use the content property to return text from a text range. The following example selects the next paragraph formatted with the Heading 1 style. The contents of the content property are displayed by the display dialog command.

set selfInd to find object of selection
clear formatting selfInd
set style of selfInd to style heading1
execute find selfInd find text "" wrap find find stop with ¬
find format and match forward
if found of selfInd is true then
    display dialog (get content of text object of selection)
end if

The following instruction returns the selected text.

set strText to content of text object of selection

The following example returns the first word in the active document. Each item in the words list is a text range object that represents one word.

set aFirst to word 1 of text object of active document
display dialog (get content of aFirst)

The following example returns the text associated with the first bookmark in the active document.

if (count of bookmarks of active document) ≥ 1 then
    set bookText to content of text object of bookmark 1 of active document
display dialog bookText
end if
Applying formatting to text

This topic includes information and examples related to the following tasks:

- Applying formatting to the selection
- Applying formatting to a text range
- Inserting text and applying character and paragraph formatting
- Toggling the space before a paragraph between 12 points and none
- Increasing the left margin by 0.5 inch

Applying formatting to the selection

The following example uses the selection object to apply character and paragraph formatting to the selected text. To access character formatting properties, use the font object property. To access paragraph formatting properties and commands, use the paragraph format property.

```plaintext
set fSel to font object of selection
set name of fSel to "Times New Roman"
set font size of fSel to 14
set all caps of fSel to True
set pSel to paragraph format of selection
set paragraph format left indent of pSel to (inches to points inches 0.5)
space 1 pSel
```

Applying formatting to a text range

The following example defines a text range object that refers to the first three paragraphs in the active document. The text range (myRange) is formatted by applying properties of the font and paragraph format objects.

```plaintext
set myRange to create range active document start (start of content of text object of paragraph 1 of active document) end (end of content of text object of paragraph 3 of active document)
set name of font object of myRange to "Arial"
set alignment of paragraph format of myRange to align paragraph justify
```
Using the Word Dictionary

**Inserting text and applying character and paragraph formatting**

The following example adds the word Title at the top of the current document. The first paragraph is center aligned, and a half-inch space is added after the paragraph. The word Title is formatted with 24 point Arial font.

```
set oRange to create range active document start 0 end 0
set content of oRange to "Title"
set oRange to change end of range oRange by a word item
   extend type by selecting
set name of font object of oRange to "Arial"
set font size of font object of oRange to 24
type paragraph selection
set alignment of paragraph 1 of active document to align paragraph center
set space after of paragraph 1 of active document to
   (inches to points inches 0.5)
```

**Toggling the space before a paragraph between 12 points and none**

The following example toggles the space before formatting of the first paragraph in the selection. The script retrieves the current space before value; if the value is 12 points, the space before formatting is removed (the *space before* property is set to zero). If the space before value is anything other than 12, then *space before* property is set to 12 points.

```
set oParagraph to paragraph 1 of selection
if space before of oParagraph is 12 then
   set space before of oParagraph to 0
else
   set space before of oParagraph to 12
end if
```

**Increasing the left margin by 0.5 inch**

The following example increases the left margin by 0.5 inch. The *page setup* object contains all the page setup attributes of a document (left margin, bottom margin, paper size, and so on) as properties. The *left margin* property is used to return and set the left margin setting.

```
set iMargin to left margin of page setup of active document
set iMargin to iMargin + (inches to points inches 0.5)
set left margin of page setup of active document to iMargin
```
Using the Word Dictionary

Editing text
This topic includes examples related to the following tasks:

- Determining whether text is selected
- Collapsing a selection or range
- Extending a selection or range
- Redefining a text range object
- Changing text

For information about and examples of other editing tasks, see the following topics:

- Returning text from a document
- Selecting text in a document
- Inserting text in a document

Determining whether text is selected
The selection type property of the selection object returns information about the type of selection. The following example displays a message if the selection is an insertion point.

```list
if selection type of selection is selection ip then ¬

display dialog "Nothing is selected"
```

Collapsing a selection or range
To collapse a text range object to its beginning or ending point, use the collapse range command. To collapse a selection to its beginning or ending point, use the collapse range command on the text object of the selection. The following example collapses the selection to an insertion point at the beginning of the selection.

```list
collapse range text object of selection direction collapse start
```

The following example cancels the myRange object to its ending point (after the first word).

```list
set myRange to word 1 of active document
set myRange to collapse range myRange direction collapse end
```

Extending a selection or range
The following example uses the move end of range command to extend oRange to include the first three paragraphs in the active document.

```list
set oRange to text object of paragraph 1 of active document
set oRange to move end of range oRange by a paragraph item count 2
```

Redefining a text range object
To redefine an existing text range object, use the set range command. For more information, see Working with text range objects.

Changing text
You can change existing text by changing the contents of a range. The following instruction changes the first word in the active document by setting the content property to "The."

```list
set content of word 1 of active document to "The "
```
Finding and replacing is exposed by the **find** and **replacement** objects. The **find** object is available from the **selection** and **text range** object. The find action differs slightly depending upon whether you access the **find** object from the **selection** or **range** object.

### Finding text and selecting it

If the **find** object is accessed from the **selection** object, the selection is changed when the find criteria is found. The following example selects the next occurrence of the word "Hello." If the end of the document is reached before the word "Hello" is found, the search is stopped.

```plaintext
set selFind to find object of selection
set forward of selFind to true
set wrap of selFind to find stop
set content of selFind to "Hello"
execute find selFind
```

The **find** object includes properties that relate to the options in the **Find and Replace** dialog box (*Edit* menu, **Find** command). You can set the individual properties of the **find** object or use arguments with the **execute find** command as shown in the following example.

```plaintext
execute find find object of selection find text "Hello" wrap find find stop ¬
with match forward
```

### Finding text without changing the selection

If the **find** object is accessed from a **text range** object, the selection is not changed. The following example searches for the text "blue" in the active document and returns true if it is found.

```plaintext
set theFind to find object of text object of active document
tell theFind
    set content to "blue"
    set forward to true
    set myFind to execute find
end tell
```

The following example performs the same result as the previous example using arguments of the **execute find** method.

```plaintext
set myRange to text object of active document
execute find find object of myRange find text "blue" with match forward
```

### Using the replacement object

The **replacement** object represents the replace criteria for a find-and-replace operation. The properties and commands of the **replacement** object correspond to the options in the **Find and Replace** dialog box (*Edit* menu).
Using the Word Dictionary

The replacement object is available from the find object. The following example replaces all occurrences of the word "hi" with "hello." The selection changes when the find criteria is found because the find object is accessed from the selection object.

```
set selfFind to find object of selection
tell selfFind
  clear formatting
  set content to "hi"
  clear formatting replacement
  set content of replacement to "hello"
  execute find wrap find find continue ¬
      replace replace all with match forward
end tell
```

Selecting text in a document

Use the select command to select an item in a document. The select command is available from several objects, such as bookmark, field, text range, and table. The following example selects the first table in the active document.

```
select table 1 of active document
```

The following example selects the first field in the active document.

```
select field 1 of active document
```

The following example selects the first four paragraphs in the active document. The create range command is used to create a text range object that refers to the first four paragraphs. The select command is then applied to the text range object.

```
set myRange to create range active document start (start of content of ¬
    text object of paragraph 1 of active document) end (end of content ¬
    of text object of paragraph 4 of active document)
select myRange
```

For more information, see Working with the selection object.

Inserting text in a document

Use the insert command to insert text in a document. The following example inserts text at the end of the active document.

```
insert text " the end" at end of text object of active document
```

You can also use the content property of a text range to insert text in a document. The following example replaces the text in the selection.

```
set content of text object of selection to "new text"
```

After the content of a text range is changed, the text range is not changed to reflect the new text. To reset the text range to the new text, use the change start of range and change end of range commands.
Working with tables

This topic includes examples related to the following tasks:

- Inserting text into a table cell
- Creating a table, inserting text, and applying formatting
- Returning text from a table cell without returning the end-of-cell marker
- Converting existing text to a table
- Returning the contents of each table cell

**Inserting text into a table cell**

The following example inserts text into the first cell of the first table in the active document. The `getCell from table` command returns a single `cell` object.

```vba
if (count of tables of active document) > 1 then
    set content of text object of (getCell from table table 1 of active document row 1 column 1) to "Cell 1, 1"
end if
```

**Creating a table, inserting text, and applying formatting**

The following example inserts a four-column, three-row table at the beginning of the document. The `Repeat With...In` structure is used to step through each cell in the table. Within the `Repeat With...In` structure, the `insert` command is used to add text to the table cells (Cell 1, Cell 2, and so on).

```vba
set oDoc to active document
set oTable to make new table at oDoc with properties ¬
    {text object:(create range oDoc start 0 end 0), number of rows:3, ¬
     number of columns:4}
set iCount to 1
repeat with oCell in (get cells of text object of oTable)
    insert text ("Cell " & iCount) at text object of oCell
    set iCount to iCount + 1
end repeat
auto format table oTable table format table format colorful2 ¬
    with apply borders, apply font and apply color
```
Using the Word Dictionary

**Returning text from a table cell without returning the end-of-cell marker**
The following examples return and display the contents of each cell in the first row of the first document table.

```plaintext
set oTable to table 1 of active document
repeat with aCell in (get cells of row 1 of oTable)
    set myRange to create range active document start (start of content of text object of aCell) end ((end of content of text object of aCell) - 1)
    display dialog (get content of myRange)
end repeat
```

```plaintext
set oTable to table 1 of active document
repeat with aCell in (get cells of row 1 of oTable)
    set myRange to text object of aCell
    set myRange to move end of range myRange by a character item count - 1
    display dialog (get content of myRange)
end repeat
```

**Converting existing text to a table**
The following example inserts tab-delimited text at the beginning of the active document and then converts the text to a table.

```plaintext
set oRange1 to create range active document start 0 end 0
set content of oRange1 to "one" & tab & "two" & tab & "three" & tab
set oRange1 to change end of range oRange1 by a paragraph item extend type by selecting
set oTable1 to convert to table oRange1 separator separate by tabs number of rows 1 number of columns 3
```

**Returning the contents of each table cell**
The following example defines a list containing the contents of the cells in the first document table. The `Repeat With...In` structure is used to return the contents of each table cell and assign the text to the next list item.

```plaintext
if (count of tables of active document) ≥ 1 then
    set oTable to table 1 of active document
    set aCells to {}
    repeat with oCell in (get cells of text object of oTable)
        set myRange to text object of oCell
        set myRange to move end of range myRange by a character item count -1
        set end of aCells to content of myRange
    end repeat
end if
Class: assistant

Plural
assistants

Elements
balloon

Represents the Microsoft Office Assistant.

Use the office assistant property of the application class to return the assistant. Only one assistant object can be active at a time. Use the visible property to display the Assistant.

By default, the Office Assistant is off. The default Assistant is Max. To select a different Assistant programatically, use the file name property.

The following example turns on, displays, and animates the Assistant.

set assistant on of office assistant to true
set visible of office assistant to true
set animation of office assistant to greeting
Properties

**animation**

Returns or sets an animation action for the Office Assistant. When this property is applied to the **assistant** object, the Assistant is animated immediately (if it is visible). When this property is applied to the **balloon** object, the Assistant is animated only while the balloon is displayed.

Can be one of the following constants:

- `appear`
- `begin speaking`
- `character success major`
- `checking something`
- `disappear`
- `empty trash`
- `gesture down`
- `gesture left`
- `gesture right`
- `gesture up`
- `get artsy`
- `get attention major`
- `get attention minor`
- `get techy`
- `get wizardy`
- `goodbye`
- `greeting`
- `assist with help`
- `assist with wizards`
- `assistant item`
- `assistant on`

**assist with help**

`True` if the Office Assistant appears when the user presses the HELP key to display Help. Read/write.

If this property is set to `False`, the Help window appears instead of the Office Assistant.

This property corresponds to the **Respond to HELP key** option (**Office Assistant** dialog box, **Options** tab).

**assist with wizards**

`True` if the Office Assistant provides online Help with wizards. Read/write.

This property corresponds to the **Help with wizards** option (**Office Assistant** dialog box, **Options** tab).

**assistant item**

Returns the text associated with the Office Assistant. Read-only.

**assistant on**

`True` if the Office Assistant is enabled. Read/write.
Microsoft Office Suite

**balloon error**

Returns a value that indicates the last recorded balloon error. Read-only.

Can be one of the following:

- **bad picture reference** The balloon contains a graphic that couldn't be displayed because the file doesn't exist or because the graphic isn't a valid PICT file.
- **bad reference** The balloon contains an unrecognized or unsupported reference.
- **buttonless modal** The balloon you attempted to display is modal, but it contains no buttons. The balloon won't be shown because it can't be dismissed.
- **button modeless** The balloon you attempted to display is modeless, contains no buttons, and has no procedure assigned to the callback property. The balloon won't be shown because a callback procedure is required for modeless balloons.
- **none** No error was encountered.
- **bad character** The balloon contains an ASCII control character other than CR or LF and greater than 32.
- **out of memory** The balloon won't appear because there is insufficient memory.
- **too big** The balloon is too big to appear on the screen.
- **other** The balloon won't appear because some other error occurred; for example, another modal balloon is already active.

**feature tips**

True if the Office Assistant provides information about using application features more effectively. Read/write.

This property corresponds to the Using features more effectively option (Office Assistant dialog box, Options tab).

**file name**

Returns or sets the name of the file for the active Office Assistant. Read/write.

The Office Assistant files are installed in the Microsoft Office 2004/Office/Assistants folder.

**keyboard shortcut tips**

True if the Office Assistant displays Help about keyboard shortcuts. Read/write.

This property corresponds to the Keyboard shortcuts option (Office Assistant dialog box, Options tab).

**left position**

Returns the horizontal position of the Office Assistant window (in points) relative to the screen.
Microsoft Office Suite

**mouse tips**

*True* if the Office Assistant provides suggestions for using the mouse effectively. Read/write.

This property corresponds to the **Using the mouse more effectively** option (Office Assistant dialog box, Options tab).

**move when in the way**

*True* if the Office Assistant window automatically moves when it's in the way of the user's work area. For example, the Office Assistant will move if it's in the way of dragging or dropping or in the way of keystroke entries. Read/write.

This property corresponds to the **Move when in the way** option (Office Assistant dialog box, Options tab).

**name**

Returns the name of the Office Assistant currently in use. Read-only.

**sounds**

*True* if the Office Assistant produces the sounds that correspond to animations. Read/write.

If a sound card is not installed, this property has no effect.

**tip of the day**

*True* if the Office Assistant displays a special tip each time the Office application is opened. Read/write.

This property corresponds to the **Show the Tip of the Day at startup** option (Office Assistant dialog box, Options tab).

**top**

Returns the vertical position of the Office Assistant window (in points) relative to the screen.

**visible**

*True* if the Office Assistant is visible. Read-write.

---

**Class:** balloon

**Plural**

balloons

**Elements**

balloon checkbox

balloon label

Represents the balloon in which the Office Assistant displays information. A balloon can contain controls such as check boxes and labels.

Only one balloon can be visible at a time. However, it's possible to define several balloons and display any one of them when needed. For more information, see "Defining and Reusing Balloons" later in this topic.
Microsoft Office Suite

To make the specified balloon visible, use the `show balloon` command. Use the `call back` property to run procedures based on selections from modeless balloons (balloons that remain visible while a user works in the application).

The following example creates a balloon that contains tips for saving entered data.

```plaintext
set newbal to make new balloon of office assistant
set balloon type of newbal to bullets
set icon of newbal to icon tip
set balloon button of newbal to buttons ok cancel
set heading of newbal to "Tips for Saving Information."
repeat 3 times
    make new balloon label at the beginning of newbal
end repeat
set label text of balloon label 1 of newbal to "Save your work often."
set label text of balloon label 2 of newbal to "Install a surge protector."
set label text of balloon label 3 of newbal to "Exit your application properly."
show balloon newbal
```

**Defining and Reusing Balloons**

You can reuse balloon objects you've already created by assigning the object to a variable and displaying the variable when you need it. This example defines `balloon1`, `balloon2`, and `balloon3` separately so they can be reused.

```plaintext
set balloon1 to make new balloon of office assistant
set heading of balloon1 to "First balloon"

Set balloon2 to make new balloon of office assistant
set heading of balloon2 to "Second balloon"

Set balloon3 to make new balloon of office assistant
set heading of balloon3 to "Third balloon"
```

```plaintext
show balloon balloon1
show balloon balloon3
show balloon balloon2
```

Alternatively, instead of using separate variables, you can place the balloon object into an array.
Properties

balloon button

Returns or sets the type of button displayed at the bottom of the Office Assistant balloon. When you create a new balloon, this property is initially set to OK. Read/write.

Can be one of the following:

- buttons abort retry ignore
- buttons back close
- buttons back next close
- buttons back next snooze
- button cancel
- buttons next close
- button none
- button ok
- buttons ok cancel
- buttons retry cancel
- buttons search close
- buttons tips options close
- buttons yes all no cancel
- buttons yes no cancel
- buttons yes no

balloon mode

Returns or sets the type of balloon displayed. Read/write.

Can be one of the following:

- auto down balloon - Balloon is instantly dismissed when the user clicks anywhere on the screen.
- modal balloon - This is the initial default setting. User must dismiss the balloon before he or she can return to working in the application.
- modeless balloon - User can work in the application while the balloon is visible. When this value is used, a value for the call back callback property is also required. When you create a new balloon, this property is initially set to modal balloon.

balloon text

Returns or sets the text displayed after the heading but before the labels or check boxes in the Office Assistant balloon. Read/write.

You can specify a graphic to display by using the following syntax:

{type location sizing_factor},

where

- type is pict (Macintosh PICT file).
- location is the resource id or the path and file name.
- sizing_factor specifies the width of the pict.
**balloon type**

Returns or sets the type of balloon the Office Assistant uses. Read/write.

Can be one of the following:

- **buttons** - This is the initial default setting.
- **bullets**
- **numbers**

**call back**

Sets the name of the procedure to run from a modeless balloon. Read/write String.

The procedure you specify for the **call back** property must be written to receive three arguments:

- The balloon object that called the procedure
- The value of the button the user clicked
- An integer that uniquely identifies the balloon that called the procedure

**heading**

Returns or sets the heading that appears in the Office Assistant balloon. Read/write.

You can specify a graphic to display in the balloon heading by using the following syntax:

```
{type location sizing_factor}
```

where

- **type** is pict (Macintosh PICT file).
- **location** is the resource id or the path and file name
- **sizing_factor** specifies the width of the pict.

**icon**

Returns or sets the type of icon that appears in the upper-left portion of the Office Assistant balloon. Read/write.

Can be one of the following:

- **icon alert**
- **icon alert critical**
- **icon alert info**
- **icon alert warning**
- **icon application**
- **icon none**
- **icon tip**

**name**

Returns the name of the balloon. Read-only.
**Class: balloon checkbox**

Represents a check box in the Office Assistant balloon.

Use `balloon checkbox index`, where `index` is a number from 1 through 5, to return a single `balloon checkbox` object. There can be up to five check boxes in one balloon; each check box appears when a value is assigned to its `checkbox text` property.

**Remarks**

Balloon check boxes display the user's choices until the user dismisses the balloon. To record the user's choice as soon as the user clicks the button beside the label, see the topic on balloon labels.

**Properties**

- `balloon checkbox item`
  
  Returns the text associated with the balloon check box. Read-only.

- `checkbox text`
  
  Returns or sets the text displayed next to a check box in the Office Assistant balloon. Read/write.

- `checked`
  
  True if the specified check box in the Office Assistant balloon is checked. Read/write.

- `name`
  
  Returns the name of the check box label. Read-only.

**Class: balloon label**

Represents a label in the Office Assistant balloon.

Use `balloon label index`, where `index` is a number from 1 through 5, to return a `balloon label` object. There can be up to five labels on one balloon; each label appears when a value is assigned to its `label text` property.

**Remarks**

Balloon labels record the user's choice as soon as the user clicks the button beside the label. To display the user's choices until the user dismisses the balloon, see the topic on balloon check boxes.

**Properties**

- `balloon label item`
  
  Returns the text associated with the balloon label. Read-only.
label text

Returns or sets the text displayed next to a label in the Office Assistant balloon. Read/write.

Remarks

You can specify a graphic to display by using the following syntax:

{type location sizing_factor},
where

type is pict (Macintosh PICT file)
location is the resource id or the path and file name
and sizing_factor specifies the width of the pict

name

Returns the name of the balloon label. Read-only.

Class: command bar

Plural

command bars

Elements

command bar control

Represents a command bar in an Office application.

Use command bar index, where index is the name or index number of a command bar, to return a command bar object. The following example steps through the collection of command bars to find the command bar named "Forms." If it finds this command bar, the example makes it visible and protects its docking state. In this example, the variable bars represents the list of all command bars, and the variable cb represents a command bar object.

set foundFlag to false
set bars to command bars
repeat with cb in bars
    repeat 1 times
        if name of cb is "Forms" then
            set protection of cb to no change dock
            set visible of cb to true
            set foundFlag to true
        end if
    exit repeat
    set end of bars to cb
end repeat
if foundFlag is false then
    display dialog "There is no Forms command bar."
end if
You can use a name or index number to specify a menu bar or toolbar in the list of available menu bars and toolbars in the Office application. However, you must use a name to specify a menu, shortcut menu, or submenu (all of which are represented by command bar objects). This example adds a new menu item to the bottom of the Tools menu. When a user clicks the new menu item, it runs the procedure named "qtrReport."

Set newItem to make new command bar button at command bar "Tools"
set begin group of newItem to True
set caption of newItem to "Make Report"
set face id of newItem to 0
set OnAction of newItem to "qtrReport"

If two or more custom menus or submenus have the same name, command bar index returns the first one.

**Properties**

**bar position**

Returns or sets the position of a command bar. Read/write.

Can be one of the following:

- bar left
- bar top
- bar right
- bar bottom
- bar floating
- bar pop up
- bar menu

**bar type**

Returns the type of command bar. Read-only.

Can be one of the following:

- normal command bar
- menubar command bar
- popup command bar

**built in**

True if the specified command bar or command bar control is a built-in command bar or control of the container application. False if it's a custom command bar or control. Read-only.
context
Returns or sets a string that determines where a command bar will be saved. The string is defined and interpreted by the application. Read/write.

You can set the context property only for custom command bars. This property will fail if the application doesn't recognize the context string, or if the application doesn't support changing context strings programmatically.

enabled
True if the specified command bar or command bar control is enabled. Read/write.

For command bars, setting this property to True causes the name of the command bar to appear in the list of available command bars.

For built-in controls, setting this property to True causes the application to determine the state of the control. Setting this property to False forces the control to be disabled.

entry index
Returns the index number for a command bar. Read-only.

The position of the first command bar control is 1. Separators are not counted.

height
Returns or sets the height (in pixels) of a command bar or command bar control. Read/write.

An error will occur if you attempt to set the height property for a command bar that isn't in a resizable state (that is, if it's docked or protected from resizing).

left position
Returns or sets the distance (in pixels) from the left edge of the specified command bar or command bar control to the left edge of the screen. Returns the distance from the left side of the docking area. Read/write.

local name
Returns the name of a built-in command bar as it's displayed in the language version of the container application, or returns or sets the name of a custom command bar. Read/write.

Note: If you attempt to set this property for a built-in command bar, an error occurs.

The local name of a built-in command bar is displayed in the title bar (when the command bar isn't docked) and in the list of available command bars, wherever that list is displayed in the container application.

If you change the value of the local name property for a custom command bar, the value of name changes as well, and vice versa.
name

Returns or sets the name of the specified object. Read/write.

The local name of a built-in command bar is displayed in the title bar (when the command bar isn't docked) and in the list of available command bars, wherever that list is displayed in the container application.

For a built-in command bar, the name property returns the command bar's U.S. English name. To return the localized name, use the local name property.

If you change the value of the name property for a custom command bar, the value of local name changes as well, and vice versa.

protection

Returns or sets the way a command bar is protected from user customization. Read/write.

Can be one of or a sum of the following:

- no protection
- no customize
- no resize
- no move
- no change visible
- no change dock
- no vertical dock
- no horizontal dock

row index

Returns or sets the docking order of a command bar in relation to other command bars in the same docking area. Can be an integer greater than zero.

Several command bars can share the same row index, and command bars with lower numbers are docked first. If two or more command bars share the same row index, the command bar most recently assigned will be displayed first in its group.

top

Returns or sets the distance (in pixels) from the top edge of the specified command bar or command bar control to the top edge of the screen. For docked command bars, this property returns or sets the distance from the command bar to the top of the docking area. Read/write.

visible

True if a command bar or command bar control is visible. Read/write.

width

Returns or sets the width (in pixels) of the specified command bar or command bar control. Read/write.
Class: command bar button

Plural
command bar buttons

Represents a button control on a command bar.

Use command bar control index, where index is the index number of the control, to return a command bar button object. (The control type property of the control must be control button.)

Properties

button face is default
   True if the face of the specified command bar button control is its original built-in face. This property can only be set to True, which will reset the face to the built-in face. Read/write.

button state
   Returns or sets the appearance of a command bar button control. Can be one of the following:
   - button state up
   - button state down
   - button state unset

button style
   Returns or sets the way a command bar button control is displayed. Read/write.
   Can be one of the following:
   - button automatic
   - button icon
   - button caption
   - button icon and caption

face id
   Returns or sets the ID number for the face of a command bar button control. Read/write.
   The face id property dictates the look, but not the function, of a command bar button. To determine the function of the button, use the id property of the command bar control object.
   The value of the face id property for a command bar button with a custom face is 0 (zero).

shortcut text
   Returns or sets the shortcut key text displayed next to a button control when the button appears on a menu, submenu, or shortcut menu. Read/write.
   You can set this property only for command bar buttons that contain an on-action macro.
Class: command bar combobox

Plural
command bar comboboxes

Represents a combo box control on a command bar.

Use command bar control index, where index is the index number of the control, to return a command bar combobox object. (The control type property of the control must be control edit, control dropdown, control combobox, button dropdown, split dropdown, OCX dropdown, graphic combo, or graphic dropdown.)

The following example creates a new command bar combobox and adjusts the size of the control on the command bar named "Custom," and then it adds two items to the combo box.

```
set newCombo to make new command bar control at command bar "Custom" with properties {control type:control combobox}
set drop down lines of newCombo to 3
set drop down width of newCombo to 90
set list index of newCombo to 0
add item to combobox newCombo combobox item "First Item" entry index 1
add item to combobox newCombo combobox item "Second Item" entry index 2
```

Properties

combobox style

Returns or sets the way a command bar combo box control is displayed. Read/write.

Can be either of the following:

- combobox style label
- combobox style normal

combobox text

Returns or sets the text in the display or edit portion of the command bar combo box control. Read/write.

drop down lines

Returns or sets the number of lines in the specified command bar combo box control. The combo box control must be a custom control, and it must be either a drop-down list box or a combo box. Read/write.

**Note** If this property is set for a combo box control that's either an edit box or a built-in combo box control, an error occurs.

If this property is set to 0 (zero), the number of lines in the control will be based on the number of items in the list.
drop down width

Returns or sets the width (in pixels) of the list for the specified command bar combo box control. Read/write.

*Note*  An error occurs if you attempt to set this property for a built-in control.

If this property is set to -1, the width of the list is based on the length of the longest item in the combo box list. If this property is set to 0, the width of the list is based on the width of the control.

list index

Returns or sets the index number of the selected item in the list portion of the command bar combo box control. If nothing is selected in the list, this property returns zero. Read/write.

*Note*  This property fails when applied to controls other than list controls.

Setting the list index property causes the specified control to select the given item and execute the appropriate action in the application.

**Class: command bar control**

**Plural**

**command bar controls**

Represents a command bar control. The properties and commands of the command bar control object are all shared by the command bar button, command bar combobox, and command bar popup classes.

*Note*  To write scripts to work with custom command bar controls, use the command bar button, command bar combobox, and command bar popup classes. To write scripts to work with built-in controls in the container application that cannot be represented by one of those three classes, use the command bar control class.

Use command bar control index, where index is the index number of a control, to return a command bar control object. (The control type property of the control must be control label, expanding grid, split expanding grid, control grid, or control gauge.)

*Note*  Variables declared as command bar control can be assigned command bar button, command bar combobox, and command bar popup values.

**Properties**

*begin group*

**begin group**

*True* if the specified command bar control appears at the beginning of a group of controls on the command bar. Read/write.

**built in**

*True* if the specified command bar or command bar control is a built-in command bar or control of the container application. False if it's a custom command bar or control, or if it's a built-in control whose on action property has been set. Read-only.
control type

Returns the type of command bar control. Read-only.

Can be one of the following:

- control button
- button drop down
- button popup
- control combobox
- control custom
- control dropdown
- control edit
- expanding grid
- control gauge
- generic dropdown
- graphic combo
- graphic popup
- graphic dropdown
- control grid
- OCX dropdown
- msopopup
- split button MRU popup
- split button popup
- split dropdown
- split expanding grid

description text

Returns or sets the description for the specified command bar control. The description is displayed in the status bar of the container application when the user positions the pointer over a command bar control. Read/write.

Not all applications display a status bar.

enabled

True if the specified command bar or command bar control is enabled. Read/write.

For command bars, setting this property to True causes the name of the command bar to appear in the list of available command bars.

For built-in controls, if you set the enabled property to True, the application determines its state. Setting the enabled property to False forces the control to be disabled.

entry index

Returns the index number for the command bar control. Read-only.

height

Returns or sets the height (in pixels) of a command bar or command bar control. Read/write.

An error will occur if you attempt to set the height property for a command bar that isn't in a resizable state (that is, if it's docked or protected from resizing).

help context ID

Returns or sets the Help context Id number for the Help topic attached to the command bar control. Read/write.

To use this property, you must also set the help file property.

help file

Returns or sets the Help file name for the Help topic attached to the command bar control. Read/write.

To use this property, you must also set the help context ID property.
id
Returns the ID for a built-in command bar control. Read-only.
A control's ID determines the built-in action for that control. The value of the id property for all custom controls is 1.

left position
Returns the distance (in pixels) from the left edge of the specified command bar or command bar control to the left edge of the screen. Returns the distance from the left side of the docking area. Read-only.

name
Returns or sets the caption text for a command bar control. Read/write.

on action
Returns or sets the name of a Visual Basic procedure that will run when the user clicks or changes the value of a command bar control. Read/write.

parameter
Returns or sets a string that an application can use to execute a command. Read/write.
If the specified parameter is set for a built-in control, the application can modify its default behavior if it can parse and use the new value. If the parameter is set for custom controls, it can be used to send information to Visual Basic procedures, or it can be used to hold information about the control (similar to a second tag property value).

priority
Returns or sets the priority of a command bar control. A control's priority determines whether the control can be dropped from a docked command bar if the command bar controls can't fit in a single row. Read/write.
Valid priority numbers are 0 (zero) through 7. Special priority numbers are 0 and 1. A priority of 0 indicates an "automatic" value, which means a number is assigned based on the type of control. A priority of 1 means the control cannot be dropped.

tag
Returns or sets information about the command bar control, such as data that can be used as an argument in procedures, or information that identifies the control. Read/write.

tooltip text
Returns or sets the text displayed in a command bar control's ScreenTip. Read/write.

top
Returns or sets the distance (in pixels) from the top edge of the specified command bar or command bar control to the top edge of the screen. For docked command bars, this property returns or sets the distance from the command bar to the top of the docking area. Read-only.

visible
True if a command bar or command bar control is visible. Read/write.
width

Returns or sets the width (in pixels) of the specified command bar or command bar control.
Read/write.

Class: command bar popup

Plural
command bar popups

Elements
command bar control

Represents a pop-up control on a command bar.
Use command bar control index, where index is the number of the control, to return a command bar popup object. (The control type property of the control must be control popup, graphic popup, button popup, split button popup, or split button MRU popup.)

Class: custom document property

Plural
custom document properties

Represents a custom document property of a container document.
Use custom document property index, where index is the name or index number of the custom document property, to return a custom document property object that represents a specific custom document property.

Class: document property

Plural
document properties

Represents a built-in document property of a container document.
Use document property index, where index is the name or index number of the built-in document property, to return a single document property object that represents a specific built-in document property.

The names of all the available built-in document properties are shown on the following list:

<table>
<thead>
<tr>
<th>Title</th>
<th>Creation Date</th>
<th>Manager</th>
</tr>
</thead>
<tbody>
<tr>
<td>Subject</td>
<td>Last Save Time</td>
<td>Company</td>
</tr>
<tr>
<td>Author</td>
<td>Total Editing Time</td>
<td>Number of Bytes</td>
</tr>
<tr>
<td>Keywords</td>
<td>Number of Pages</td>
<td>Number of Lines</td>
</tr>
<tr>
<td>Comments</td>
<td>Number of Words</td>
<td>Number of Paragraphs</td>
</tr>
<tr>
<td>Template</td>
<td>Number of Characters</td>
<td>Number of Slides</td>
</tr>
<tr>
<td>Last Author</td>
<td>Security</td>
<td>Number of Notes</td>
</tr>
<tr>
<td>Revision Number</td>
<td>Category</td>
<td>Number of Hidden Slides</td>
</tr>
<tr>
<td>Application Name</td>
<td>Format</td>
<td>Number of Multimedia Clips</td>
</tr>
<tr>
<td>Last Print Date</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Container applications don't necessarily define a value for every built-in document property. If a given application doesn't define a value for one of the built-in document properties, returning the `value` property for that document property causes an error.

**Properties**

*document property type*

<table>
<thead>
<tr>
<th>Returns or sets the document property type. Read-only for built-in document properties; read/write for custom document properties. Can be one of the following:</th>
</tr>
</thead>
<tbody>
<tr>
<td>boolean</td>
</tr>
<tr>
<td>date</td>
</tr>
<tr>
<td>float</td>
</tr>
<tr>
<td>number</td>
</tr>
<tr>
<td>string</td>
</tr>
</tbody>
</table>

*link source*

Returns or sets the source of a linked custom document property. Read/write.  
This property applies only to custom document properties; you cannot use it with built-in document properties.  
The source of the specified link is defined by the container application.  
Setting the `link source` property sets the `link to context` property to **True**.

*link to context*

**True** if the value of the custom document property is linked to the content of the container document. **False** if the value is static. Read/write.  
This property applies only to custom document properties. For built-in document properties, the value of this property is **False**.  
Use the `link source` property to set the source for the specified linked property. Setting the `link source` property sets the `link to context` property to **True**.

*name*

Returns the name of the specified object. Read-only.

*value*

Returns or sets the value of a document property. Read/write.  
If the container application doesn't define a value for one of the built-in document properties, reading the `value` property for that document property causes an error.
**Class: web page font**

Represents the default font used when documents are saved as Web pages for a particular character set.

Use the web page font object to describe the proportional font, proportional font size, fixed-width font, and fixed-width font size for any available character set. The following character sets are supported:

- Arabic
- Cyrillic
- English Western European Other Latin Script
- Greek
- Hebrew
- Japanese
- Korean
- Multilingual
- Unicode
- Simplified Chinese
- Thai
- Traditional Chinese
- Vietnamese

**Properties**

*fixed width font*

Sets or returns the fixed-width font setting in the host application. Read/write. When you set the fixed width font property, the host application does not check the value for validity.

*fixed width font size*

Sets or returns the fixed-width font size setting in the host application, in points. Read/write. When you set the fixed width font size property, the host application does not check the value for validity. If you enter an invalid value, such as a nonnumber, the host application sets the size to 0 points. You can enter half-point sizes; if you enter other fractional point sizes, they are rounded up or down to the nearest half-point.

*proportional font*

Sets or returns the proportional font setting in the host application. Read/write. When you set the proportional font property, the host application does not check the value for validity.

*proportional font size*

Sets or returns the proportional font size setting (in points) in the host application. Read/write. When you set the proportional font size property, the host application does not check the value for validity. If you enter an invalid value, such as a nonnumber, the host application sets the size to 0 points. You can enter half-point sizes; if you enter other fractional point sizes, they are rounded up or down to the nearest half-point.
Commands

add item to combobox ........................................................................................................... 49
clear combobox .................................................................................................................. 49
execute ................................................................................................................................... 50
get combobox item ............................................................................................................ 50
get count of combobox items ............................................................................................ 51
remove an item from combobox ....................................................................................... 51
reset ...................................................................................................................................... 52
show balloon ....................................................................................................................... 52

Command: add item to combobox

Adding a list item to the specified command bar combo box control. The combo box control must be a custom control, and it must be either a drop-down list box or a combo box.

Note  This command will fail if it's applied to an edit box or a built-in combo box control.

Syntax

add item to combobox command bar combobox   Required. An expression that returns a command bar combobox object

combobox item  Unicode text   Required. The text to be added to the specified control.

[entry index  small integer]   Optional. The position of the specified item in the list. If this argument is omitted, the item is added at the end of the list.

Example

This example adds a combo box control to a command bar. Two items are added to the control, and the number of line items and the width of the combo box are set.

set myBar to make new command bar with properties ¬
    {bar type:normal command bar, name:"New"}
set myControl to make new command bar control at myBar¬
    with properties {control type:control combobox, combobox text:"Monday",¬
    combobox style:combobox style label, drop down lines:2, drop down width:75}
add item to combobox myControl combobox item "Monday" entry index 1
add item to combobox myControl combobox item "Tuesday" entry index 2

Command: clear combobox

Removes all list items from the specified command bar combo box control (drop-down list box or combo box) and clears the text box (edit box or combo box).

Note  This method will fail if it's applied to a built-in command bar control.

Syntax

clear combobox command bar combobox   Required. An expression that returns a command bar combobox object.
Example

This example checks the number of items in the combo box control named "Names" on the command bar named "Custom." If there are more than three items in the list, the example clears the list, adds a new first item to the list, and displays this new item as the default for the combo box control.

```csharp
set myBar to command bar "Custom"
set myControl to command bar control "Names" of myBar
set listItems to get count of combobox items myControl
if listItems > 3 then
  clear combobox myControl
  add item to combobox myControl combobox item "Third Item" entry index 1
  set list index of myControl to 1
End If
```

Command: execute

Runs the procedure associated with a command bar control.

Syntax

```csharp
execute   command bar control   Required. An expression that returns a command bar control, command bar button, or command bar combobox object.
```

Remarks

This method runs the specified script or command if controls are currently enabled in the application and if the enabled property of the specified object is True.

Applying this method to a command bar popup object generates a run-time error.

Example

This example checks the value of the combo box control on the custom command bar named "My Custom Bar." If the index number of the command bar control is 1, the example runs the script specified by the on action property of the command bar control.

```csharp
set myControl to the first command bar combobox of command bar "My Custom Bar"
if entry index of myControl = 1 then
  execute myControl
End If
```

Command: get combobox item

Returns the string at the given index of items within a combo box control.

Syntax

```csharp
get combobox item   command bar combobox   Required. An expression that returns a command bar combobox object.
```

```csharp
entry index   small integer   Required. The position of the specified item in the list. If this argument is omitted, the last item in the list is returned.
```
**Command: get count of combobox items**

Returns the number of items within a combo box control.

**Syntax**

```
get count of combobox items  command bar combobox   Required. An expression that returns a command bar combobox object.
```

**Example**

This example uses the `get count of combobox items` command to check the number of items in the combo box control named "Names" on the command bar named "Custom." If there are more than three items on the list, the example clears the list, adds a new first item to the list, and displays this new item as the default for the combo box control.

```
set myBar to command bar "Custom"
set myControl to command bar control "Names" of myBar
set listItems to get count of combobox items myControl
if listitems > 3 then
  clear combobox myControl
  add item to combobox myControl combobox item "Third Item" entry index 1
  set list index of myControl to 1
end if
```

**Command: remove an item from combobox**

Removes an item from a command bar combo box control.

**Note**  This command fails when applied to controls other than list controls.

**Syntax**

```
remove an item from combobox  command bar combobox   Required. An expression that returns a command bar combobox object.
entry index   small integer   Required. The index number of the item to be removed from the list.
```

**Example**

The following example determines whether there are more than three items in a combo box control. If there are more than three items, the example removes the second item, alters the style, and sets a new value.

```
set myBar to command bar "Custom"
set myControl to command bar combobox 1 of myBar
set listItems to get count of combobox items myControl
if listitems > 3 then
  remove an item from combobox myControl entry index 2
  set combobox style of myControl to combobox style normal
  set combobox text of myControl to "New Default"
end if
```
**Command: reset**

Resets a built-in command bar to its default configuration, or resets a built-in command bar control to its original function and face.

**Syntax**

```
reset command bar/command bar control   Required. An expression that returns a command bar, command bar control, command bar button, command bar popup, or command bar combobox object.
```

**Remarks**

Resetting a built-in control restores the actions originally intended for the control and resets each of the control's properties back to its original state. Resetting a built-in command bar removes custom controls and restores built-in controls.

**Example**

This example resets the **Formatting** toolbar to its default state

```
reset command bar "Formatting"
```

**Command: show balloon**

Displays the specified balloon object. Returns a constant that indicates which balloon the user clicks.

**Syntax**

```
show balloon   balloon   Required. An expression that returns a balloon object.
```

**Remarks**

You can use the return value of the **show balloon** command to display a user's button selection. The **show balloon** command returns one of the following buttons:

- abort button
- cancel button
- ignore button
- no button
- OK button
- retry button
- snooze button
- yes button

- back button
- close button
- next button
- null button
- options button
- search button
- tips button
- yes to all button
Example
This example creates a balloon that contains three choices
set b to make new balloon at office assistant
tell b
    set heading to "This is my heading"
    set balloon text to "Select one of these things"
    repeat 3 times
        make new balloon label at the beginning
    end repeat
    set label text of balloon label 1 to "Choice One"
    set label text of balloon label 2 to "Choice Two"
    set label text of balloon label 3 to "Choice Three"
    show balloon
end tell
Microsoft Word Suite

Classes

add in ................................................................. 56
application ......................................................... 58
auto text entry ..................................................... 64
bookmark .............................................................. 65
border ................................................................. 67
border options .................................................... 71
browser ............................................................... 75
caption label ........................................................ 76
check box ............................................................ 78
custom label ......................................................... 79
data merge .......................................................... 82
data merge data field .......................................... 84
data merge data source ....................................... 84
data merge field .................................................. 86
data merge field name ........................................ 87
default web options .......................................... 88
dialog ................................................................. 91
document ........................................................... 92
document version ............................................... 100
drop cap ............................................................. 100
drop down .......................................................... 101
endnote ............................................................... 102
endnote options ................................................ 103
envelope ............................................................. 104
field ................................................................. 107
file converter ..................................................... 109
find ................................................................. 111
font ................................................................. 114
footnote ............................................................ 119
footnote options ................................................. 119
form field .......................................................... 119
frame ............................................................... 121
header footer ..................................................... 123
heading style .................................................... 125
hyperlink object ................................................ 127
index ............................................................... 129
key binding ....................................................... 131
letter content ..................................................... 133
line numbering ................................................ 136
link format ....................................................... 137
list entry .......................................................... 137
list format ........................................................ 139
list gallery ....................................................... 142
list level ........................................................... 142

Commands

...
Microsoft Word Suite

list template........................................................................................................................... 144
mailing label............................................................................................................................ 145
page number ........................................................................................................................... 146
page number options ............................................................................................................... 147
page setup ............................................................................................................................... 148
pane ......................................................................................................................................... 153
recent file ................................................................................................................................ 155
replacement ............................................................................................................................. 156
revision ..................................................................................................................................... 158
selection object .......................................................................................................................... 159
subdocument ............................................................................................................................ 166
system object ........................................................................................................................... 167
tab stop ..................................................................................................................................... 168
table of authorities .................................................................................................................. 169
table of contents ...................................................................................................................... 171
table of figures ........................................................................................................................ 173
template .................................................................................................................................... 175
text column .............................................................................................................................. 177
text input ................................................................................................................................... 178
text retrieval mode .................................................................................................................... 179
variable ..................................................................................................................................... 180
view .......................................................................................................................................... 181
web options .............................................................................................................................. 185
window ....................................................................................................................................... 188
Word comment ........................................................................................................................ 191
Word list ..................................................................................................................................... 193
Word options ............................................................................................................................ 193
work menu item ....................................................................................................................... 210
zoom .......................................................................................................................................... 210

Class: add in

Represents a single add-in, either installed or not installed. The add ins list contains all the add-ins available to Word, regardless of whether or not they’re currently loaded. The add ins list includes global templates or Word add-in libraries (WLLs) displayed in the Templates and Add-ins dialog box (Tools menu).

Use add in index, where index is the index number of the add-in, to return a single add in object.
You must exactly match the spelling (but not necessarily the capitalization) of the name, as it’s shown in the Templates and Add-Ins dialog box. The following example loads the first add in in the add ins list.

set installed of add in 1 to true

The index number represents the position of the add-in in the list of add-ins in the Templates and Add-ins dialog box. The following instruction displays the path of the first available add-in.

if (count of add ins) ≥ 1 then display dialog (get path of add in 1)
The following example creates a list of installed add-ins at the beginning of a new document. The list contains the name, path, and installed state of each available add-in.

```plaintext
set myDocument to make new document
type text selection text "Name" & tab & "Path" & tab & "Installed" & return
repeat with oAddIn in (get add ins whose installed is true)
type text selection text (name of oAddIn) & tab & (path of oAddIn) & ~
tab & (installed of oAddIn) & return
end repeat
convert to table text object of myDocument separator separate by tabs
```

**Note** Use the `compiled` property to determine whether an `add in` object is a template or a WLL.

**Properties**

- `autoload`
  - **True** if the specified add-in is automatically loaded when Word is started. Add-ins located in the Startup folder in the Word program folder are automatically loaded. Read-only.

- `compiled`
  - **True** if the specified add-in is a Word add-in library (WLL). **False** if the add-in is a template. Read-only.

- `entry index`
  - Returns a number that indicates the position of the add-in in the add-ins list. Read-only.

- `installed`
  - **True** if the specified add-in is installed (loaded). Add-ins that are loaded are selected in the Templates and Add-ins dialog box (Tools menu). Read/write.

  **Note** Uninstalled add-ins are included in the `add ins` list. To remove a template or WLL from the `add ins` list, use the `delete` command with the `add in` object (the add-in name is removed from the Templates and Add-ins dialog box).

- `name`
  - Returns the file name of the add-in. Read-only.

- `path`
  - Returns the path to the specified `add in` object. Read-only.

  **Note** The path doesn’t include a trailing character; for example, Macintosh HD:Users:Shared.
Class: application

Plural
applications

Elements
document
window
recent file
file converter
caption label
add in
command bar
template
key binding
dictionary

work menu item

Represents the Word application. The application object includes properties that return top-level objects. For example, the active document property returns a document object.

The following example displays the user name for Word:

display dialog (user name of application "Microsoft Word") as string

Many of the properties and commands that return the most common user-interface objects — such as the active document (active document property) — can be accessed through the application object. For example, you can write:

tell application "Microsoft Word" to print out active document

You can also write blocks of code without explicitly specifying the application object in each reference. The following example is another way of expressing the example listed above:

tell application "Microsoft Word"
    print out active document.
end tell

Properties

Word 51 menus

Determines whether Microsoft Word 5.1 menus and toolbars are displayed. Read/write.

active document

Returns a document object that represents the active document (the document with the focus). If there are no documents open, an error occurs. Read-only.
**active printer**

Returns the name of the active printer. Read/write.

**active window**

Returns a window object that represents the active window (the window with the focus). If there are no windows open, an error occurs. Read-only.

**application version**

Returns the Microsoft Word version number. Read-only.

**autocorrect object**

Returns an autocorrect object that contains the current AutoCorrect options, entries, and exceptions. Read-only.

**background printing status**

Returns the number of print jobs in the background printing queue. Read-only.

**browse extra file types**

Set this property to "text/html" to allow hyperlinked HTML files to be opened in Word (instead of in the default Internet browser). Read/write.

**browser object**

Returns a browser object that represents the Select Browse Object tool on the vertical scroll bar. Read-only.

**build**

Returns the version and build number of the Word application. Read-only.

**caps lock**

True if the CAPS LOCK key is turned on. Read-only.

**caption**

Returns the caption text for the specified application window. Read-only.

**customization context**

Returns or sets a template or document object that represents the template or document in which changes to menu bars, toolbars, and key bindings are stored. Corresponds to the value of the Save in box on the Commands tab in the Customize dialog box (Tools menu). Read/write.

**default save format**

Returns or sets the default format that will appear in the Format box of the Save As dialog box (File menu). Corresponds to the Save Word files as box in the Save pane of the Preferences dialog box (Word menu). Read/write.
The string used with this property is the file converter class name. The class names for internal Word formats are listed in the following table.

<table>
<thead>
<tr>
<th>Word format</th>
<th>File converter class name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Word Document</td>
<td>&quot;&quot;</td>
</tr>
<tr>
<td>Document Template</td>
<td>&quot;Dot&quot;</td>
</tr>
<tr>
<td>Text Only</td>
<td>&quot;Text&quot;</td>
</tr>
<tr>
<td>Text Only with Line Breaks</td>
<td>&quot;CRTText&quot;</td>
</tr>
<tr>
<td>MS-DOS Text</td>
<td>&quot;8Text&quot;</td>
</tr>
<tr>
<td>MS-DOS Text with Line Breaks</td>
<td>&quot;8CRTText&quot;</td>
</tr>
<tr>
<td>Rich Text Format</td>
<td>&quot;Rtf&quot;</td>
</tr>
<tr>
<td>Unicode Text</td>
<td>&quot;Unicode Text&quot;</td>
</tr>
</tbody>
</table>

Use the class name property with a file converter object to determine the class name of an external file converter.

**default table separator**

Returns or sets the single character used to separate text into cells when text is converted to a table. Read/write.

**Note** The value of the default table separator property is used if the separator argument is omitted from the convert to table command.

**default web options object**

Returns the default web options object that contains global application-level attributes used by Word whenever you save a document as a Web page or open a Web page. Read-only.

**display alerts**

Returns or sets the way certain alerts and messages are handled while a macro is running. Read/write.

Can be one of the following:

<table>
<thead>
<tr>
<th>Constant</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>alerts none</td>
<td>No alerts or message boxes are displayed. If a macro encounters a message box, the default value is chosen and the macro continues.</td>
</tr>
<tr>
<td>alerts message box</td>
<td>Only message boxes are displayed; errors are trapped and returned to the macro.</td>
</tr>
<tr>
<td>alerts all</td>
<td>All message boxes and alerts are displayed; errors are returned to the macro.</td>
</tr>
</tbody>
</table>

**Note** If you set this property to alerts none or alerts message box, Word doesn't set it back to alerts all when your script stops running. You should write your script in such a way that it always sets the display alerts property back to alerts all when it stops running.
Microsoft Word Suite

*display auto complete tips*

**True** if Word displays tips that suggest text for completing words, dates, or phrases as you type. Read/write.

*display recent files*

**True** if the names of recently used files are displayed on the **File** menu. Read/write.

*display screen tips*

**True** if comments, footnotes, endnotes, and hyperlinks are displayed as tips. Text marked as having comments is highlighted. Read/write.

*display scroll bars*

**True** if Word displays a scroll bar in at least one document window. **False** if there are no scroll bars displayed in any window. Read/write.

Setting the *display scroll bars* property to **true** displays horizontal and vertical scroll bars in all windows. Setting this property to **false** turns off all scroll bars in all windows.

Use the *display horizontal scroll bar* and *display vertical scroll bar* properties to display individual scroll bars in the specified window.

*display status bar*

**True** if the status bar is displayed. Read/write.

*do print preview*

**True** if print preview is the current view. Read/write.

*enable cancel key*

Returns or sets the way that Word processes ⌘+period user interruptions. Read/write.

Can be one of the following:

- **cancel disabled** (prevents ⌘+period from interrupting a script)
- **cancel interrupt** (allows a script to be interrupted by ⌘+period)

Use this property very carefully. If you use **cancel disabled**, there's no way to interrupt a runaway loop or other non-self-terminating code. The *enable cancel key* property is reset to **cancel interrupt** when your script stops running.

*font names*

Returns a list that includes the names of all the available fonts. Read-only.

*landscape font names*

Returns a list that includes the names of all the available landscape fonts. Read-only.

*macro container*

Returns a **template** object or **document** object that represents the template or document in which a VB module that contains a running procedure is stored. This property will usually return **missing value**. Read-only.
Microsoft Word Suite

mailing label object

Returns a **mailing label** object that represents a mailing label. Read-only.

name

Returns the name of the application (for example, "Microsoft Word"). Read-only.

normal template

Returns a **template** object that represents the Normal template. Read-only.

num lock

Returns the state of the NUM LOCK key. **True** if the keys on the numeric key pad insert numbers; **false** if the keys move the insertion point. Read-only.

office assistant

Returns an **assistant** object that represents the Office Assistant. Read-only.

path

Returns the path to the specified application. Read-only.

**Note**  The path doesn't include a trailing character; for example, Macintosh HD:Applications:Microsoft Office 2004. To add the character that separates folders and volumes, use the *path separator* property.

path separator

Returns the character used to separate folder names. Read-only.

portrait font names

Returns a list that includes the names of all the available portrait fonts. Read-only.

screen updating

**True** if screen updating is turned on. Read/write.

The *screen updating* property controls most display changes on the monitor while a procedure is running. When screen updating is turned off, toolbars remain visible and Word still allows the procedure to display or retrieve information using status bar prompts, input boxes, dialog boxes, and message boxes. You can increase the speed of some procedures by keeping screen updating turned off. You must set the *screen updating* property to **true** when the procedure finishes or stops after an error.

selection

Returns the **selection object** that represents a selected range or the insertion point. Use the *select* command to set or change the selection. Read-only.

settings

Returns a **Word options** object that represents application settings you can set in Word. Read-only.

**show visual basic editor**

**True** if the Visual Basic Editor window is visible. Read/write.
Microsoft Word Suite

**special mode**

True if Word is in a special mode (for example, CopyText mode or MoveText mode). Read-only.

Word enters a special copy or move mode if you press F2 or SHIFT+F2 while text is selected.

**startup dialog**

True if Word displays the Project Gallery dialog box when the application starts. Read/write.

**startup path**

Returns or sets the complete path of the Startup folder, excluding the final separator.

Read/write.

Note  Templates and add-ins located in the Startup folder are automatically loaded when you start Word.

**status bar**

Displays the specified text in the status bar. Write-only.

**system object**

Returns the system object, which can be used to return system-related information and perform system-related tasks. Read-only.

**usable height**

Returns the height (in points) of the space that can be used by a window in the application window area. Read-only.

Note  If the usable height property returns 1, no space is available for a document window. When no vertical space is available, the horizontal (usable width) value isn't valid. To determine the actual available height, subtract 1 from the usable height value.

**usable width**

Returns the width (in points) of the space that can be used by a window in the application window area. Read-only.

Note  The value of the usable width property is valid only when the value of the usable width property is greater than 1, which indicates that there's space available for a document window.

**user address**

Returns or sets the default address of the contact marked as "This contact is Me" in the Office Address Book. Read/write.

The default address is also the address that appears in the User Information pane of the Preferences dialog box and is used as a return address on envelopes and other types of documents.

**user control**

True if the document or application was created or opened by the user. False if the document or application was created or opened programmatically.

If Word is visible to the user, this property will always return true.
**user initials**

Returns or sets the user’s initials, which Word uses to construct comment marks. Read/write.

**user name**

Returns or sets the name of the contact marked as "This contact is Me" in the Office Address Book. Read/write.

The user name is also the name that appears in the User Information pane of the Preferences dialog box and is used on envelopes and other types of documents and for the Author document property.

**Class: auto text entry**

**Plural**

**auto text entries**

Represents a single AutoText entry. The auto text entries list contains all the AutoText entries in the specified template. The entries are listed on the AutoText tab in the AutoCorrect dialog box (Tools menu).

Use auto text entry index, where index is the AutoText entry name or index number, to return a single auto text entry object. You must exactly match the spelling (but not necessarily the capitalization) of the name, as it’s shown on the AutoText tab in the AutoCorrect dialog box. The following example sets the value of an existing AutoText entry named "cName."

```
set auto text value of auto text entry "cName" of normal template to ¬
    "The Johnson Company"
```

The following example displays the name and value of the first AutoText entry in the template attached to the active document.

```
set myTemp to attached template of active document
display dialog "Name = " & name of auto text entry 1 of myTemp & return & ¬
    "Value = " & auto text value of auto text entry 1 of myTemp
```

The following example inserts the global AutoText entry named "TheWorld" at the insertion point.

```
collapse range text object of selection direction collapse end
insert auto text entry (auto text entry "TheWorld" of normal template) where ¬
    (text object of selection)
```

Use the make command to add an auto text entry object to the auto text entries list. The following example adds an AutoText entry named “Blue” based on the text of the selection.

```
make new auto text entry at normal template with properties ¬
    {name:"Blue", auto text value:text object of selection}
```

**Properties**

**auto text value**

Returns or sets the value of the AutoText entry. Read/write.
Microsoft Word Suite

**entry index**

Returns a number that indicates the position of the AutoText entry in list of AutoText entries. Read-only.

**name**

Returns or sets the name of the AutoText entry. Read/write.

**style name**

Returns the name of the style applied to the specified AutoText entry. Read-only.

**Class: bookmark**

**Plural**

**bookmarks**

Represents a single bookmark. The **bookmarks** list includes all the bookmarks listed in the **Bookmark** dialog box (Insert menu).

Use **bookmark index**, where **index** is the bookmark name or index number, to return a single **bookmark** object. You must exactly match the spelling (but not necessarily the capitalization) of the bookmark name. The following example selects the bookmark named "temp" in the active document.

**select bookmark "temp" of active document**

The index number represents the position of the bookmark in the **selection** or **range** object. For the **document** object, the index number represents the position of the bookmark in the alphabetical list of bookmarks in the **Bookmarks** dialog box (click **Name** to sort the list of bookmarks alphabetically). The following example displays the name of the second bookmark in the **bookmarks** list.

**display dialog (name of bookmark 2 of active document) as string**

Use the **make** command to add a bookmark to a document range. The following example marks the selection by adding a bookmark named "temp."

**make new bookmark at active document with properties ¬
(name:"temp", text object:text object of selection)**

**Remarks**

To return the index number of the **bookmark** object in the **bookmarks** list, use the **bookmark id** property with a range or selection object. The following example displays the index number of the bookmark named "temp" in the active document.

**display dialog (bookmark id of text object of bookmark "temp" of ¬
active document) as string**
Properties

column

True if the specified bookmark is a table column. Read-only.

empty

True if the specified bookmark is empty. An empty bookmark marks a location (a collapsed selection); it doesn't mark any text. Read-only.

Note An error occurs if the specified bookmark doesn't exist.

end of bookmark

Returns or sets the ending character position of the bookmark. Read/write.

Note If this property is set to a value smaller than the start of bookmark property, the start of bookmark property is set to the same value (that is, the start of bookmark and end of bookmark property are equal).

The ending position is the point farthest away from the beginning of the story. This property returns the ending character position relative to the beginning of the story. The main document story type (main text story) begins with character position 0 (zero). You can change the size of a bookmark by setting this property.

name

Returns the name of the bookmark. Read-only.

start of bookmark

Returns or sets the starting character position of a bookmark. Read/write.

Note If this property is set to a value larger than that of the end of bookmark property, the end of bookmark property is set to the same value as that of the start of bookmark property.

The starting position refers to the character position closest to the beginning of the story. This property returns the starting character position relative to the beginning of the story. The main text story type (main text story) begins with character position 0 (zero). You can change the size of a selection, range, or bookmark by setting this property.
story type

Returns the story type for the specified range, selection, or bookmark. Read-only.

Can be one of the following:

- comments story
- endnotes story
- even pages footer story
- even pages header story
- first page footer story
- first page header story
- footnotes story
- main text story
- primary footer story
- primary header story
- text frame story

text object

Returns a text range object that represents the portion of a document that's contained in the specified object. Read-only.

Class: border

Plural

borders

Represents a border of an object.

To return a single border object, use the get border command. The get border command can be used to return one of the following:

- border bottom
- border horizontal
- border left
- border right
- border top
- border vertical

To apply a border line to a border object, use the line style property. The following example applies a double-line border below the first paragraph in the active document.

```
set theBorder to get border paragraph 1 of active document which border bottom
set line style of theBorder to line style double
set line width of theBorder to line width 25 point
```

The following example applies a single-line border around the first character in the selection.

```
set font size of font object of character 1 of selection to 36
set enable borders of border options of character 1 of selection to true
```

Border objects cannot be added to the borders list. The number of members in the borders list is finite and varies depending on the type of object. For example, a table has six elements in the borders list, whereas a paragraph has four.
### Properties

#### art style

Returns or sets the graphical page-border design for a document. Read/write.

Can be one of the following:

<table>
<thead>
<tr>
<th>art style</th>
<th>art style</th>
</tr>
</thead>
<tbody>
<tr>
<td>art apples</td>
<td>art holly</td>
</tr>
<tr>
<td>art arched scallops</td>
<td>art house funky</td>
</tr>
<tr>
<td>art baby pacifier</td>
<td>art hypnotic</td>
</tr>
<tr>
<td>art baby rattle</td>
<td>art ice cream cones</td>
</tr>
<tr>
<td>art balloons3 colors</td>
<td>art light bulb</td>
</tr>
<tr>
<td>art balloons hot air</td>
<td>art lightning1</td>
</tr>
<tr>
<td>art basic black dashes</td>
<td>art lightning2</td>
</tr>
<tr>
<td>art basic black dots</td>
<td>art maple leaf</td>
</tr>
<tr>
<td>art basic black squares</td>
<td>art maple muffins</td>
</tr>
<tr>
<td>art basic thin lines</td>
<td>art map pins</td>
</tr>
<tr>
<td>art basic white dashes</td>
<td>art marquee</td>
</tr>
<tr>
<td>art basic white dots</td>
<td>art marquee toothed</td>
</tr>
<tr>
<td>art basic white squares</td>
<td>art moons</td>
</tr>
<tr>
<td>art basic wide inline</td>
<td>art mosaic</td>
</tr>
<tr>
<td>art basic wide midline</td>
<td>art music notes</td>
</tr>
<tr>
<td>art basic wide outline</td>
<td>art northwest</td>
</tr>
<tr>
<td>art bats</td>
<td>art ovals</td>
</tr>
<tr>
<td>art birds</td>
<td>art packages</td>
</tr>
<tr>
<td>art birds flight</td>
<td>art palms black</td>
</tr>
<tr>
<td>art cabins</td>
<td>art palms color</td>
</tr>
<tr>
<td>art cake slice</td>
<td>art paper clips</td>
</tr>
<tr>
<td>art candy corn</td>
<td>art papyrus</td>
</tr>
<tr>
<td>art celtic knotwork</td>
<td>art party favor</td>
</tr>
<tr>
<td>art certificate banner</td>
<td>art party glass</td>
</tr>
<tr>
<td>art chain link</td>
<td>art pencils</td>
</tr>
<tr>
<td>art champagne bottle</td>
<td>art people</td>
</tr>
<tr>
<td>art checked bar black</td>
<td>art people hats</td>
</tr>
<tr>
<td>art checked bar color</td>
<td>art people waving</td>
</tr>
<tr>
<td>art checkered</td>
<td>art poinsettias</td>
</tr>
<tr>
<td>art christmas tree</td>
<td>art postage stamp</td>
</tr>
<tr>
<td>art circles lines</td>
<td>art pumpkin1</td>
</tr>
<tr>
<td>art circles rectangles</td>
<td>art push pin note1</td>
</tr>
<tr>
<td>art classical wave</td>
<td>art push pin note2</td>
</tr>
<tr>
<td>art clocks</td>
<td>art pyramids</td>
</tr>
<tr>
<td>art compass</td>
<td>art pyramids above</td>
</tr>
<tr>
<td>art confetti</td>
<td>art quadrants</td>
</tr>
<tr>
<td>art confetti grays</td>
<td>art rings</td>
</tr>
<tr>
<td>art confetti outline</td>
<td>art safari</td>
</tr>
<tr>
<td>art confetti streamers</td>
<td>art sawtooth</td>
</tr>
<tr>
<td>art confetti white</td>
<td>art sawtooth gray</td>
</tr>
<tr>
<td>art corner triangles</td>
<td>art scared cat</td>
</tr>
<tr>
<td>art coupon cutout dashes</td>
<td>art seattle</td>
</tr>
<tr>
<td>art coupon cutout dots</td>
<td>art shadowed squares</td>
</tr>
<tr>
<td>art crazy maze</td>
<td>art sharks teeth</td>
</tr>
<tr>
<td>art creatures butterfly</td>
<td>art shorebird tracks</td>
</tr>
<tr>
<td>art creatures fish</td>
<td>art skyrocket</td>
</tr>
<tr>
<td>art creatures insects</td>
<td>art snowflake fancy</td>
</tr>
</tbody>
</table>
art creatures lady bug art snowflakes
art cross stitch art sombrero
art cup art southwest
art deco arch art stars
art deco arch color art stars3D
art deco blocks art stars black
art diamonds gray art stars shadowed
art double d art stars top
art double diamonds art sun
art earth1 art swirligig
art earth2 art torn paper
art eclipsing squares1 art torn paper black
art eclipsing squares2 art trees
art eggs black art triangle party
art fans art triangle
art film art tribal1
art firecrackers art tribal2
art flowers block print art tribal3
art flowers daisies art tribal4
art flowers modern1 art tribal5
art flowers modern2 art tribal6
art flowers pansy art twisted lines1
art flowers red rose art twisted lines2
art flowers roses art vine
art flowers teacup art waveline
art flowers tiny art weaving angles
art gems art weaving braid
art gradient art weaving strips
art handmade1 art white flowers
art handmade2 art woodwork
art heart balloon art x illusions
art heart gray art zany triangles
art hearts art zig zag
art heebie jeebies art zig zag stitch

**art width**

Returns or sets the width (in points) of the specified graphical page border. Read/write.

**color**

Returns or sets the RGB color for the specified border object. Read/write.

**color index**

Returns or sets the color for the specified border or font object. Read/write.

Can be one of the following:

<table>
<thead>
<tr>
<th>auto</th>
<th>dark yellow</th>
<th>red</th>
</tr>
</thead>
<tbody>
<tr>
<td>black</td>
<td>gray25</td>
<td>teal</td>
</tr>
<tr>
<td>blue</td>
<td>gray50</td>
<td>turquoise</td>
</tr>
<tr>
<td>bright green</td>
<td>green</td>
<td>violet</td>
</tr>
<tr>
<td>dark blue</td>
<td>no highlight</td>
<td>white</td>
</tr>
<tr>
<td>dark red</td>
<td>pink</td>
<td>yellow</td>
</tr>
</tbody>
</table>
inside

**True** if an inside border can be applied to the specified object. Read-only.

**line style**

Returns or sets the border line style for the specified object. Read/write.

Can be one of the following:

- line style none
- line style single
- line style dot
- line style dash small gap
- line style dash large gap
- line style dash dot
- line style dash dot dot
- line style double
- line style triple
- line style thin thick small gap
- line style thin thick large gap
- line style thin thick small gap
- line style thin thick med gap
- line style thick thin small gap
- line style thick thin med gap
- line style thick thin large gap
- line style single wavy
- line style double wavy
- line style dash dot stroked
- line style emboss3D
- line style engrave3D
- line style thin thick thin small gap
- line style thin thick med gap
- line style thin thick large gap

Setting the **line style** property for a range that refers to individual characters or words applies a character border.

Setting the **line style** property for a paragraph or range of paragraphs applies a paragraph border. To apply a border between consecutive paragraphs, use the **inside line style** property of the **border options** object.

Setting the **line style** property for a section applies a page border around the pages in the section.

**line width**

Returns or sets the line width of the border of an object. Read/write.

Can be one of the following:

- line width25 point
- line width50 point
- line width75 point
- line width100 point
- line width150 point
- line width225 point
- line width300 point
- line width450 point
- line width600 point

**Note** If the specified line width isn't available for the border's line style, this property generates an error. To determine the line widths available for a particular line style, see the **Borders and Shading** dialog box (Format menu).
visible

True if the specified object is visible. Read/write.

Class: border options

Represents options associated with the border of an object.

Use the border options property to return the border options object. The following example applies the default border around the first paragraph in the active document.

```plaintext
set enable borders of border options of paragraph 1 of active document to true
```

Properties

always in front

True if page borders are displayed in front of the document text. Read/write.

distance from

Returns or sets a value that indicates whether the specified page border is measured from the edge of the page or from the text it surrounds. Read/write.

Can be either of the following:

- border distance from text
- border distance from page edge

distance from bottom

Returns or sets the space (in points) between the text and the bottom border. Read/write.

Note Using this property with a page border, you can set either the space between the text and the bottom page border or the space between the bottom edge of the page and the bottom page border. Where the distance is measured from depends on the value of the distance from property.

distance from left

Returns or sets the space (in points) between the text and the left border. Read/write.

Note Using this property with a page border, you can set either the space between the text and the left page border or the space between the left edge of the page and the left page border. Where the distance is measured from depends on the value of the distance from property.

distance from right

Returns or sets the space (in points) between the right edge of the text and the right border. Read/write.

Note Using this property with a page border, you can set either the space between the text and the right border or the space between the right edge of the page and the right border. Where the distance is measured from depends on the value of the distance from property.
distance from top

Returns or sets the space (in points) between the text and the top border. Read/write.

**Note** Using this property with a page border, you can set either the space between the text and the top page border or the space between the top edge of the page and the top page border. Where the distance is measured from depends on the value of the `distance from` property.

enable borders

Returns or sets border formatting for the specified object. Read/write.

The `enable borders` property applies to all borders for the specified object. **True** sets the line style to the default line style and sets the line width to the default line width.

To remove all the borders from an object, set the `enable borders` property to **false**, as shown in the following example.

```
set enable borders of border options of table 1 of active document to false
```

To remove or apply a single border, use the `get border` command to return a single border, and then set the `line style` property. The following example removes the bottom border from `myRange`.

```
set line style of (get border myRange which border border bottom) to line style none
```

enable first page in section

**True** if page borders are enabled for the first page in the section. Read/write.

enable other pages in section

**True** if page borders are enabled for all pages in the section except for the first page. Read/write.

has horizontal

**True** if a horizontal border can be applied to the object. Read-only.

Horizontal borders can be applied to ranges that contain cells in two or more rows of a table or ranges that contain two or more paragraphs.

has vertical

**True** if a vertical border can be applied to the specified object. Read-only.

Vertical borders can be applied to ranges that contain cells in two or more columns of a table.

inside color

Returns or sets the RGB color of the inside borders. Read/write.

If the `inside line style` property is set to **line style none**, setting this property has no effect.
inside color index

Returns or sets the color of the inside borders. Read/write.

Can be one of the following:

- auto
- black
- blue
- bright green
- dark blue
- dark red
- dark yellow
- gray25
- gray50
- green
- pink
- red
- teal
- turquoise
- violet
- white
- yellow

If the inside line style property is set to line style none, setting this property has no effect.

inside line style

Returns or sets the inside border for the specified object. Read/write.

Can be one of the following:

- line style none
- line style single
- line style dot
- line style dash small gap
- line style dash large gap
- line style dash dot
- line style dash dot dot
- line style double
- line style triple
- line style thin thick small gap
- line style thick thin small gap
- line style thin thick thin small gap
- line style thin thick med gap
- line style thick thin med gap
- line style thick thin large gap
- line style thin thick large gap
- line style single wavy
- line style double wavy
- line style dash dot stroked
- line style emboss3D
- line style engrave3D
- line style thin thick small gap
- line style thick thin small gap
- line style thin thick thin small gap

inside line width

Returns or sets the line width of the inside border of an object. Read/write.

Can be one of the following:

- line width25 point
- line width50 point
- line width75 point
- line width100 point
- line width125 point
- line width150 point
- line width225 point
- line width300 point
- line width450 point
- line width600 point

Note  If the specified line width isn't available for the border's line style, this property generates an error. To determine the line widths available for a particular line style, see the Borders and Shading dialog box (Format menu).
Microsoft Word Suite

join borders

True if vertical borders at the edges of paragraphs and tables are removed so that the horizontal borders can connect to the page border. Read/write.

outside color

Returns or sets the RGB color of the outside borders. Read/write.

If the outside line style property is set to line style none, setting this property has no effect.

outside color index

Returns or sets the color of the outside borders. Read/write.

Can be any of the following:

auto   green
black  pink
blue   red
bright green  teal
dark blue  turquoise
dark red  violet
dark yellow  white
gray25  yellow
gray50

If the outside line style property is set to line style none, setting this property has no effect.

outside line style

Returns or sets the outside border for the specified object. Read/write.

Can be one of the following:

line style none   line style thin thick med gap
line style single  line style thick thin med gap
line style dot     line style thin thick thin med gap
line style dash small gap  line style thin thick large gap
line style dash large gap   line style thick thin large gap
line style dash dot      line style thin thick thin large gap
line style dash dot dot  line style single wavy
line style double      line style double wavy
line style triple      line style dash dot stroked
line style thin thick small gap  line style emboss3D
line style thick thin small gap  line style engrave3D
outside line width

Returns or sets the line width of the outside border of an object. Read/write.

Can be one of the following:

- line width25 point
- line width50 point
- line width75 point
- line width100 point
- line width150 point
- line width225 point
- line width300 point
- line width450 point
- line width600 point

Note If the specified line width isn't available for the border's line style, this property generates an error. To determine the line widths available for a particular line style, see the Borders and Shading dialog box (Format menu).

shadow

True if the specified border is formatted as shadowed. Read/write.

surround footer

True if a page border encompasses the document footer. Read/write.

surround header

True if a page border encompasses the document header. Read/write.

Class: browser

Plural

browsers

Represents the browser tool used to move the insertion point to objects in a document. This tool is comprised of the three buttons at the bottom of the vertical scroll bar.

To return the browser object, use the browser object property of the application class. The following example moves the insertion point to just before the next field in the active document.

set browser target of browser object to browse field
next for browser browser object
The following example moves the insertion point to the previous table and selects it.

```
set browser target of browser object to browse table
previous for browser browser object
if (get selection information selection information type with in table) ¬
  is "True" then
  select table 1 of selection
end if
```

**Properties**

`browser target`

Returns or sets the document item that the `previous for browser` and `next for browser` commands locate. Read/write.

Can be one of the following:

- `browse comment`
- `browse edit`
- `browse endnote`
- `browse field`
- `browse find`
- `browse footnote`
- `browse go to`
- `browse graphic`
- `browse heading`
- `browse page`
- `browse section`
- `browse table`

**Class:** caption label

**Plural:** caption labels

Represents a single caption label. The items in the `caption labels` list are listed in the Label box in the Caption dialog box (Insert menu).

Use `caption label index`, where `index` is the caption label name or index number, to return a single `caption label` object. The following example sets the numbering style for the Figure caption label.

```
set number style of caption label "Figure" to caption number style ¬
  lowercase letter
```

The index number represents the position of the caption label in the `caption labels` list. The following example displays the first caption label.

```
display dialog (name of caption label 1 of active document) as string
```

To add a custom caption label, use the `make` command. The following example adds a caption label named "Photo."

```
make new caption label at active document with properties {name:"Photo"}
```
Properties

*built in*

**True** if the specified object is one of the built-in caption labels in Word. Read-only.

*caption label id*

If the *built in* property of the *caption label* object returns **true**, *caption label id* returns the type for the specified caption label. Read-only.

Can be one of the following:

- caption figure
- caption table
- caption equation

*caption label position*

Returns or sets the position of caption label text. Read/write.

Can be either of the following:

- caption position above
- caption position below

*chapter style level*

Returns or sets the heading style that marks a new chapter when chapter numbers are included with the specified caption label. The number 1 corresponds to Heading 1, 2 corresponds to Heading 2, and so on. Read/write.

**Note** The *include chapter number* property must be set to **true** for chapter numbers to be included with caption labels.

*include chapter number*

**True** if a chapter number is included with page numbers or a caption label. Read/write.

*name*

Returns the name of the caption. Read-only.

*number style*

Returns or sets the number style for the specified caption label. Read/write.

Can be one of the following:

- caption number style arabic
- caption number style lowercase letter
- caption number style lowercase roman
- caption number style uppercase letter
- caption number style chosung
- caption number style kanada
- caption number style hanja read
- caption number style hanja read digit
- caption number style kanji
- caption number style kanji digit
- caption number style kanji traditional
- caption number style number in circle
- caption number style simp chin num2
- caption number style simp chin num3
- caption number style zodiac1
- caption number style zodiac2
- caption number style uppercase roman
**separator**

Returns or sets the character between the chapter number and the sequence number. 
Read/write.

Can be one of the following:

- separator colon
- separator em dash
- separator en dash
- separator hyphen
- separator period

**Class: check box**

Represents a single check box form field.

To return a single *form field* object, use *form field* *index*, where *index* is the index number or the bookmark name associated with the check box. To return a *check box* object, use the *check box* property with the *form field* object. The following example selects the check box form field named "Check1" in the active document.

```plaintext
set check box value of check box of form field "Check1" of active document to true
```

The index number represents the position of the form field in the *form fields* list. The following example checks the type of the first form field; if it’s a check box, the check box is selected.

```plaintext
if form field type of form field 1 of active document is field form check box then
    set check box value of form field 1 of active document to true
end if
```

The following example determines whether the *ffield* object is valid before changing the check box size to 14 points.

```plaintext
set ffield to check box of form field 1 of active document
if valid of ffield is true then
    set auto size of ffield to false
    set checkbox size of ffield to 14
else
    display dialog "The first field is not a check box"
end if
```
To add a check box form field, use the `make` command with the `form field` class. The following example adds a check box at the beginning of the active document, sets the name to "Color", and then selects the check box.

```
set myFF to make new form field at active document with properties ¬
    {text object:text object of selection, form field type:field form ¬
        check box}
set name of myFF to "Color"
set check box value of check box of myFF to true
```

### Properties

**auto size**

- `True` if the check box size is determined by the font size of the surrounding text. `False` if the check box size is determined by the `checkbox size` property. Read/write.

**check box default**

- Returns or sets the default check box value. `True` if the default value is checked. Read/write.

**check box value**

- `True` if the check box is selected. Read/write.

**checkbox size**

- Returns or sets the size (in points) of the specified check box. Setting this property sets the `auto size` property to `false`. Read/write.

**valid**

- `True` if the specified form field object is a valid check box form field, drop-down form field, or text form field. `False` if it isn't valid. Read-only.

### Class: custom label

#### Plural

**custom labels**

Represents a custom mailing label. The `custom labels` list contains all the custom mailing labels listed in the `Label Options` dialog box.

Use `custom label index`, where `index` is the custom label name or index number, to return a single `custom label` object. The following example creates a new document with the custom label layout named "My Labels."

```
set ML to mailing label object
if valid of custom label "My Labels" of ML is true then
    create new mailing label document ML name "My Labels"
else
    display dialog "The My Labels custom label is not available"
end if
```
The index number represents the position of the custom mailing label in the custom labels list. The following example displays the name of the first custom mailing label.

```vba
if (count of custom labels of mailing label object) ≥ 1 then
display dialog name of custom label 1 of mailing label object as string
end if
```

To create a custom label, use the make command. The following example adds a custom mailing label named "My Label" and sets the page size.

```vba
set CL to make new custom label at mailing label object with properties ¬
{name:"My Labels", dot matrix:false}
set page size of CL to custom label A4
```

**Properties**

- **dot matrix**
  
  True if the printer type for the specified custom label is dot matrix. False if the printer type is either laser or ink jet. Read-only.

- **entry index**
  
  Returns a number that indicates the position of the custom label in the custom labels list. Read-only.

- **height**
  
  Returns or sets the height of the specified custom mailing label. Read/write.

- **horizontal pitch**
  
  Returns or sets the horizontal distance (in points) between the left edge of one custom mailing label and the left edge of the next mailing label. Read/write.

  **Note** If this property is changed to a value that isn't valid for the specified mailing label layout, an error occurs.

- **name**
  
  Returns or sets the name of the custom mailing label. Read/write.

- **number across**
  
  Returns or sets the number of custom mailing labels across a page. Read/write.

  **Note** If this property is changed to a value that isn't valid for the specified mailing label layout, an error occurs.

- **number down**
  
  Returns or sets the number of custom mailing labels down the length of a page. Read/write.

  **Note** If this property is changed to a value that isn't valid for the specified mailing label layout, an error occurs.
page size

Returns or sets the page size for the specified custom mailing label. Read/write.

Can be one of the following:

- custom label A4
- custom label A4 landscape
- custom label A5
- custom label A5 landscape
- custom label B5
- custom label fanfold
- custom label letter
- custom label letter landscape
- custom label mini

side margin

Returns or sets the side margin widths (in points) for the specified custom mailing label. Read/write.

Note If this property is changed to a value that isn't valid for the specified mailing label layout, an error occurs.

top margin

Returns or sets the distance (in points) between the top edge of the page and the top boundary of the body text. Read/write.

valid

True if the various properties (for example, height, width, and number down) for the specified custom label work together to produce a valid mailing label.

vertical pitch

Returns or sets the vertical distance between the top of one mailing label and the top of the next mailing label. Read/write.

Note If this property is changed to a value that isn't valid for the specified mailing label layout, an error occurs.

width

Returns or sets the width (in points) of the custom label. Read/write.
**Class: data merge**

**Plural**

data merges

**Elements**

data merge field

Represents the data merge functionality in Word.

To return the *data merge* object, use the *data merge* property of the *document* class. The *data merge* object is always available regardless of whether the data merge operation has begun. To determine the status of the data merge operation, use the *state* property. The following example executes a data merge if the active document is a main document with an attached data source.

```javascript
if state of data merge of active document is main and data source then
    execute data merge data merge of active document
end if
```

The following example merges the main document with the first three data records in the attached data source and then sends the results to the printer.

```javascript
set myMerge to data merge of active document
if (state of myMerge is main and source and header) or (state of myMerge is ~
    main and data source) then
    set first record of data source of myMerge to 1
    set last record of data source of myMerge to 3
end if
set destination of mymerge to send to printer
execute data merge myMerge
```

**Properties**

*data source*

Returns a *data merge data source* object that refers to the data source attached to a data merge main document. Read-only.

*destination*

Returns or sets the destination of the data merge results. Read/write.

Can be one of the following:

- send to email
- send to fax
- send to new document
- send to printer
**mail address field name**

Returns or sets the name of the field that contains electronic mail addresses that are used when the data merge destination is electronic mail. Read/write.

**mail as attachment**

_**True**_ if the merge documents are sent as attachments when the data merge destination is an e-mail message or a fax. Read/write.

**mail subject**

Returns or sets the subject line used when the data merge destination is electronic mail. Read/write.

**main document type**

Returns or sets the data merge main document type. Read/write.

Can be one of the following:

- **document type catalog**
- **document type envelopes**
- **document type form letters**
- **document type mailing labels**
- **not a merge document**

**Note** If you set this property for a document that's already a main document, the attached data source is removed.

**state**

Returns the current state of a data merge operation. Read-only.

Can be one of the following:

- **normal document**
- **main document only**
- **main and data source**
- **main and header**
- **main and source and header**
- **data source**

**suppress blank lines**

_**True**_ if blank lines are suppressed when data merge fields in a mail merge main document are empty. Read/write.

**view data merge field codes**

_**True**_ if merge field names are displayed in a data merge main document. _**False**_ if information from the current data record is displayed. Read/write.

**Note** If the active document isn’t a data merge main document, this property causes an error.
Class: data merge data field

Plural
data merge data fields

Represents a single mail merge field in a data source. The data merge data fields list includes all the data fields in a data merge data source (for example, Name, Address, and City).

Use data merge data field index, where index is the data field name or the index number, to return a single data merge data field object. The index number represents the position of the data field in the data merge data source. The following example retrieves the first value from the FName field in the data source attached to the active document.

get data merge data field value of data merge data field "FName" of data source of data merge of active document

The following example displays the name of the first field in the data source attached to the active document.

display dialog name of data merge data field 1 of data source of data merge of active document as string

You cannot add fields to the data merge data fields list. All data fields in a data source are automatically included in the data merge data fields list.

Properties
data merge data field value
    Returns the contents of the data merge data field for the current record. Use the active record property to set the active record in a data merge data source. Read-only.

entry index
    Returns a number that indicates the position of data merge data field in the data merge data fields list. Read-only.

name
    Returns the name of the data merge field. Read-only.

Class: data merge data source

Plural
data merge data sources

Elements
data merge field name
data merge data field

Represents the data merge data source in a data merge operation.

To return the data merge data source object, use the source property of the data merge class.
The following example displays the name of the data source associated with the active document.

if name of data source of data merge of active document is not "" then
    display dialog name of data source of data merge of active document as string
The following example displays the field names in the data source associated with the active document.

```lisp
repeat with aField in (get data merge field names of data source of -
   data merge of active document)
   display dialog name of aField as string
end repeat
```

The following example opens the data source associated with Form letter.doc and determines whether the FirstName field includes the name "Kate."

```lisp
set dmDoc to data merge of document "Form letter.doc"
edit data source dmDoc
if (find record data source of dmDoc find text "Kate" field name "FirstName") ¬
   is true then
   display dialog "Data was found"
end if
```

### Properties

**active record**

Returns or sets the active data merge data record. Read/write.

Can be either a valid data record number in the query result or one of the following:

- first record
- last record
- next record
- no active record
- previous record

**Note** The active data record number is the position of the record in the query result produced by the current query options; as such, this number isn't necessarily the position of the record in the data source.

**connect string**

Returns the connection string for the specified data merge data source. Read-only.

**first record**

Returns or sets the number of the first data record to be merged in a data merge operation. Read/write.

**header source name**

Returns the path and file name of the header source attached to the specified data merge main document. Read-only.
header source type

Returns a value that indicates the way the header source is being supplied for the data merge operation. Read-only.

Can be one of the following:

- merge info from odbc
- merge info from access dde
- merge info from excel dde
- merge info from msquery dde
- no merge info

last record

Returns or sets the number of the last data record to be merged in a data merge operation. Read/write.

mail merge data source type

Returns the type of data merge data source. Read-only.

Can be one of the following:

- merge info from odbc
- merge info from access dde
- merge info from excel dde
- merge info from msquery dde
- no merge info

name

Returns the full name of the data source document. Read-only.

query string

Returns or sets the query string (SQL statement) used to retrieve a subset of the data in a data merge data source. Read/write.

Class: data merge field

Plural

data merge fields

Represents a single mail merge field in a document. The data merge fields object is a member of the data merge data fields list. The data merge data fields list includes all the data merge related fields in a document.
Use **data merge field** *index*, where *index* is the index number, to return a single **data merge field** object. The following example displays the field code of the first mail merge field in the active document.

```plaintext
display dialog (content of data merge field range of data merge field 1 of ~
    data merge of active document) as string
```

**Remarks**

Several commands are available for adding fields related to a data merge operation, such as **make new data merge ask field** and **make new data merge fill in field**.

**Properties**

**data merge field range**

Returns or sets a **text range** object that represents a data merge field's code. A field's code is everything that's enclosed by the field characters ({} including the leading space and trailing space characters. You can access a field's code without changing the view from field results.

Read/write.

**form field type**

Returns the field type. Can be one of the Word fields. Read-only.

**locked**

**True** if the specified field is locked. When a field is locked, you cannot update the field results.

Read/write.

**next data merge field**

Returns the next **data merge field** in the **data merge fields** list. Read-only.

**previous make merge field**

Returns the previous **data merge field** in the **data merge fields** list. Read-only.

---

**Class: data merge field name**

**Plural**

**data merge field names**

Represents a data merge field name in a data source. The **data merge field names** list includes all the data field names in a data merge data source.

Use **data merge field name** *index*, where *index* is the name or index number of the **data merge field name**, to return a single **data merge field name** object. The index number represents the position of the field in the data merge data source. The following example retrieves the name of the last field in the data source attached to the active document.

```plaintext
set alast to count of data merge field names of data source of data merge of ~
    active document

set afirst to name of data merge field name alast of data source of ~
    data merge of active document

display dialog afirst as string
```

You cannot add fields to the **data merge field names** list. Field names in a data source are automatically included in the **data merge field names** list.
**Properties**

**entry index**

Returns a number that indicates the position of the data merge field name in the data merge field names list. Read-only.

**name**

Returns the name of the data merge field. Read-only.

**Class: default web options**

Contains global application-level attributes used by Word when you save a document as a Web page or open a Web page. You can return or set attributes either at the application (global) level or at the document level. (Note that attribute values can be different from one document to another, depending on the attribute value at the time the document was saved.) Document-level attribute settings override application-level attribute settings. Document-level attributes are contained in the **web options** class.

To return the **default web options** object, use the **default web options object** property of the **application** class. The following example checks to see whether PNG (Portable Network Graphics) is allowed as an image format and sets the **strImageFileType** variable accordingly.

```
set objAppWebOptions to default web options object
if allow png of objAppWebOptions is true then
  set strImageFiletype to "PNG"
else
  set strImageFiletype to "JPG"
end if
```

**Properties**

**allow png**

**True** if PNG (Portable Network Graphics) is allowed as an image format when you save a document as a Web page. **False** if PNG is not allowed as an output format. The default value is **false**. Read/write.

If you save images in the PNG format and if the Web browsers you are targeting support the PNG format, you might improve the image quality or reduce the size of those image files, and therefore decrease the download time.

**always save in default encoding**

**True** if the default encoding is used when you save a Web page or plain text document, independent of the file's original encoding when opened. **False** if the original encoding of the file is used. The default value is **false**. Read/write.

The **encoding** property can be used to set the default encoding.
check if office is htmleditor

True if Word checks to see whether an Office application is the default HTML editor when you start Word. False if Word does not perform this check. The default value is true. Read/write.

This property is used only if the Web browser you are using supports HTML editing and HTML editors.

To use a different HTML editor, you must set this property to false and then register the editor as the default system HTML editor.

check if word is default htmleditor

True if Word checks to see whether it is the default HTML editor when you start Word. False if Word does not perform this check. The default value is true. Read/write.

This property is used only if the Web browser you are using supports HTML editing and HTML editors.

To use a different HTML editor, you must set this property to false and then register the editor as the default system HTML editor.

encoding

Returns or sets the document encoding (code page or character set) to be used by the Web browser when you view the saved document. The default is the system code page. Read/write.

Can be any of the following:

- encoding Thai
- encoding simplified Chinese
- encoding traditional Chinese
- encoding big endian
- encoding Cyrillic
- encoding Greek
- encoding Hebrew
- encoding Baltic
- encoding simplified Chinese auto detect
- encoding traditional Chinese auto detect
- encoding Greek auto detect
- encoding ISO88591 Latin1
- encoding ISO88593 Latin3
- encoding ISO88595 Cyrillic
- encoding ISO88597 Greek
- encoding ISO88599 Turkish
- encoding ISO2022 Japanese
- encoding ISO2022 Japanese JISX02011989
- encoding ISO2022CN traditional Chinese
- encoding Mac Roman
- encoding Mac traditional Chinese Big5
- encoding Mac Greek1
- encoding Mac simplified Chinese GB2312
- encoding Mac Ukraine
- encoding Mac Icelandic
- encoding Japanese ShiftJIS
- encoding Korean
- encoding little endian
- encoding central European
- encoding Western
- encoding Arabic
- encoding Vietnamese
- encoding Japanese auto detect
- encoding Korean auto detect
- encoding Cyrillic auto detect
- encoding Arabic auto detect
- encoding ISO88592 central Europe
- encoding ISO88594 Baltic
- encoding ISO88596 Arabic
- encoding ISO88598 Hebrew
- encoding ISO88595 Latin9
- encoding ISO2022 Japanese JISX02021984
- encoding ISO2022KR
- encoding ISO2022CN simplified Chinese
- encoding Mac Japanese
- encoding Mac Korean
- encoding Mac Cyrillic
- encoding Mac Romania
- encoding Mac Latin2
- encoding Mac Turkish
encoding Mac Croatia
encoding EBCDIC International
encoding EBCDIC Greek modern
encoding EBCDIC Germany
encoding EBCDIC Finland Sweden
encoding EBCDIC Latin America Spain
encoding EBCDIC Japanese
  Katakana extended
encoding EBCDIC Arabic
encoding EBCDIC Hebrew
encoding EBCDIC Thai
encoding EBCDIC Turkish
encoding EBCDIC Serbian Bulgarian
encoding EBCDIC US Canada and Japanese
encoding EBCDIC simplified Chinese
  extended and simplified Chinese
encoding EBCDIC Japanese
  Latin extended and Japanese
encoding OEM Baltic
encoding OEM multilingual LatinII
encoding OEM Turkish
encoding OEM Icelandic
encoding OEM Canadian French
encoding OEM Nordic
encoding OEM modern Greek
encoding EUC Chinese simplified Chinese
encoding EUC Taiwanese traditional Chinese
encoding Bengali
encoding Telugu
encoding Oriya
encoding Malayalam
encoding Punjabi
encoding Arabic transparent ASMO
encoding Taiwan CNS
encoding Taiwan Eten
encoding Taiwan teletext
encoding IA5 German
encoding IA5 Norwegian
encoding T61
encoding Ext alpha lowercase
encoding Europa3
encoding UTF7
encoding EBCDIC US Canada
encoding EBCDIC multilingual
  ROECE Latin2
encoding EBCDIC Turkish Latin5
encoding EBCDIC Denmark Norway
encoding EBCDIC Italy
encoding EBCDIC United Kingdom
encoding EBCDIC France
encoding EBCDIC Greek
encoding EBCDIC Korean extended
encoding EBCDIC Icelandic
encoding EBCDIC Russian
encoding EBCDIC Japanese Katakana
  extended and Japanese
encoding EBCDIC extended and Korean
encoding EBCDIC US Canada
  and traditional Chinese
encoding OEM United States
encoding OEM multilingual LatinI
encoding OEM Cyrillic
encoding OEM Portuguese
encoding OEM Hebrew
encoding OEM Arabic
encoding OEM CyrillicII
encoding EUC Japanese
encoding EUC Korean
encoding Devanagari
encoding Tamil
encoding Assamese
encoding Kannada
encoding Gujarati
encoding Arabic ASMO
encoding Korean Johab
encoding Taiwan TCA
encoding Taiwan IBM5550
encoding Taiwan Wang
encoding IA5 Swedish
encoding US ASCII
encoding ISO6937 nonspacing accent
encoding KOI8U
encoding HZGB simplified Chinese
encoding UTF8
**pixels per inch**

Returns or sets the density (pixels per inch) of graphics images and table cells on a Web page. The range of settings is usually from 19 to 480, and common settings for popular screen sizes are 72, 96, and 120. The default setting is 96. Read/write.

This property determines the size of the images and cells on the specified Web page relative to the size of text whenever you view the saved document in a Web browser. The physical dimensions of the resulting image or cell are the result of the original dimensions (in inches) multiplied by the number of pixels per inch.

To set the optimum screen size for the targeted Web browsers, use the *screen size* property.

**screen size**

Returns or sets the ideal minimum screen size (width by height, in pixels) that you should use when viewing the saved document in a Web browser. The default constant is **resolution** 800x600. Read/write.

Can be one of the following:

- `resolution 544x376`
- `resolution 640x480`
- `resolution 720x512`
- `resolution 800x600`
- `resolution 1024x768`
- `resolution 1152x882`
- `resolution 1152x900`
- `resolution 1280x1024`
- `resolution 1600x1200`
- `resolution 1800x1440`
- `resolution 1920x1200`

**update links on save**

**True** if hyperlinks and paths to all supporting files are automatically updated before you save the document as a Web page, ensuring that the links are up-to-date at the time the document is saved. **False** if the links are not updated. The default value is **True**. Read/write.

You should set this property to **false** if the location where the document is saved is different from the final location on the Web server and the supporting files are not available at the first location.

**use long file names**

**True** if long file names are used when you save the document as a Web page. **False** if long file names are not used and the DOS file name format (8.3) is used. The default value is **true**. Read/write.

---

**Class**: dialog

**Plural**: dialogs

Represents a built-in dialog box. The **dialogs** list contains all the built-in dialog boxes in Word. You cannot create a new built-in dialog box or add one to the **dialogs** list.

To return a single **dialog** object, use the **get dialog** command along with a **dialog type** constant that identifies the dialog box. The following example displays and carries out the actions taken in the built-in **Open** dialog box (**File** menu).

```
show (get dialog dialog file open)
```
Properties

default dialog tab

Returns or sets the active tab when the specified dialog box is displayed. Read/write.

dialog type

Returns the type of built-in Word dialog box. Read-only.

Class: document

Plural
documents

Elements
document property variable
custom document property revision
bookmark table of contents
table table of authorities
footnote window
endnote index
Word comment subdocument
section hyperlink object
paragraph shape
word list template
sentence Word list
character inline shape
field document version
form field spelling error
Word style readability statistic
frame grammatical error
table of figures

Represents a document. The documents list contains all the document objects that are currently open in Word.

Use document index, where index is the document name or index number to return a single document object. The following example closes the document named "Report.doc" without saving changes.

close document "Report.doc" saving no

The index number represents the position of the document in the documents list. The following example activates the first document in the documents list.

activate document 1
Using the active document property

You can use the active document property to refer to the document with the focus. The following example uses the activate command to activate the document named "Document 1." The example also sets the page orientation to landscape mode and then prints the document.

activate document "Document 1"
set orientation of page setup of active document to orient landscape
print out active document

Properties

active theme

Returns the name of the active theme plus the theme formatting options for the specified document. Returns none if the document doesn’t have an active theme. Read-only.

For an explanation of the value returned by this property, see the name argument of the apply theme command. The value returned by this property may not correspond to the theme’s display name as it appears in the Theme dialog box (Theme command, Format menu). To return a theme’s display name, use the active theme display name property.

active theme display name

Returns the display name of the active theme for the specified document. Returns none if the document doesn’t have an active theme. Read-only.

A theme’s display name is the name that appears in the Theme dialog box (Theme command, Format menu). This name may not correspond to the string you would use to set a default theme or to apply a theme to a document.

active window

Returns a window object that represents the active window (the window with the focus). If there are no windows open, an error occurs. Read-only.

attached template

Returns a template object that represents the template attached to the specified document. To set this property, specify either the name of the template or an expression that returns a Template object. Read/write.

auto hyphenation

True if automatic hyphenation is turned on for the specified document. Read/write.

background shape

Returns a shape object that represents the background shape in the specified document. Read-only.

click and type paragraph style

Returns or sets the default paragraph style applied to text by the Click and Type feature in the specified document. To set this property, specify either the local name of the style, an integer, a built-in style, or an object that represents the style. Read/write.

If the in use property for the specified style is set to false, an error occurs.
consecutive hyphens count

Returns or sets the maximum number of consecutive lines that can end with hyphens. Read/write.

Note If this property is set to 0 (zero), any number of consecutive lines can end with hyphens.

data merge

Returns a data merge object that represents the mail merge functionality for the specified document. Read-only.

Note The data merge object is available regardless of whether the specified document is a mail merge main document. To determine the current state of the mail merge operation, use the state property.

default tab stop

Returns or sets the interval (in points) between the default tab stops in the specified document. Read/write.

document_type

Returns the document type (template or document). Read-only.

Can be either of the following:

- type document
- type template

embed true type fonts

True if Word embeds TrueType fonts in a document when it's saved. This allows others to view the document with the same fonts that were used to create it. Read/write.

endnote options

Returns an endnote options object that represents all the options for endnotes in a document. Read-only.

envelope object

Returns an envelope object that represents envelope functionality and the envelope in the specified document. Read-only.

footnote options

Returns a footnote options objects that represents all the options for footnotes in a document. Read-only.

full name

Returns the name of the specified document, including its path on disk. Read-only.

grammar checked

True if a grammar check has been run on the specified range or document. False if some of the specified range or document hasn't been checked for grammar. Read/write.

To recheck the grammar in a range or document, set the grammar checked property to false.
Microsoft Word Suite

**grid distance horizontal**
Returns or sets the amount of horizontal space between the invisible gridlines that Word uses when you draw, move, and resize AutoShapes. Read/write.

**grid distance vertical**
Returns or sets the amount of vertical space between the invisible gridlines that Word uses when you draw, move, and resize AutoShapes. Read/write.

**grid origin from margin**
True if Word starts the character grid from the upper-left corner of the page. Read/write.

**grid origin horizontal**
Returns or sets the point, relative to the left edge of the page, where you want the invisible grid for drawing, moving, and resizing AutoShapes to begin. Read/write.

**grid origin vertical**
Returns or sets the point, relative to the top of the page, where you want the invisible grid for drawing, moving, and resizing AutoShapes to begin. Read/write.

**grid space between horizontal lines**
Returns or sets the interval at which Word displays horizontal character gridlines in page layout view. Read/write.

**grid space between vertical lines**
Returns or sets the interval at which Word displays vertical character gridlines in page layout view. Read/write.

**has password**
True if a password is required to open the specified document. Read-only.

**hyphenate caps**
True if words in all capital letters can be hyphenated. Read/write.

**hyphenation zone**
Returns or sets the width (in points) of the hyphenation zone. The hyphenation zone is the maximum amount of space that Word leaves between the end of the last word in a line and the right margin. Read/write.

**is master document**
True if the specified document is a master document. A master document includes one or more subdocuments. Read-only.

**is subdocument**
True if the specified document is opened in a separate document window as a subdocument of a master document. Read-only.

**letter content**
Returns a **letter content** object that represents the elements of a letter created by the **Letter Wizard**. Read-only.
name
Returns the document’s file name. Read-only.

page setup
Returns a page setup object that's associated with the specified document. Read-only.

password
Sets a password that must be supplied to open the specified document. Write-only.

path
Returns the path to the specified document object. Read-only.

Note  The path doesn’t include a trailing character; for example, Macintosh HD:Users:Shared. To return the file name without the path, use the name property. To return the file name and the path together, use the full name property.

print forms data
True if Word prints onto a preprinted form only the data entered in the corresponding online form. Read/write.

print fractional widths
True if the specified document is formatted to use fractional point spacing to display and print characters on the Macintosh. Read/write.

print post script over text
True if PRINT field instructions (such as PostScript commands) in a document are to be printed on top of text and graphics when a PostScript printer is used. Read/write.

print revisions
True if revision marks are printed with the document. False if revision marks aren't printed (that is, tracked changes are printed as if they’d been accepted). Read/write.

protection type
Returns the protection type for the specified document. Read-only.
Can be one of the following:

- allow only comments
- allow only form fields
- allow only revisions
- no document protection

read only
True if changes to the document cannot be saved to the original document. Read-only.

read only recommended
True if Word displays a message box whenever a user opens the document, suggesting that it be opened as read-only. Read/write.
remove personal information

True if Word removes all user information from comments, revisions, and the Properties dialog box upon saving a document. Read/write.

save format

Returns the file format of the specified document or file converter. Read-only.
Can be a unique number that specifies an external file converter, or one of the following:

- `format custom dictionary`
- `format document`
- `format dostext`
- `format dostext line breaks`
- `format exclude dictionary`
- `format rtf`
- `format stationery`
- `format HTML`
- `format template`
- `format text`
- `format text line breaks`
- `format web archive`
- `format Unicode text`

save forms data

True if Word saves the data entered in a form as a tab-delimited record for use in a database. Read/write.

save subset fonts

True if Word saves a subset of the embedded TrueType fonts with the document. Read/write.

If fewer than 32 characters of a TrueType font are used in a document, Word embeds the subset (only the characters used) in the document. If more than 32 characters are used, Word embeds the entire font.

saved

True if the specified document or template hasn't changed since it was last saved. False if Word displays a prompt to save changes when the document is closed. Read/write.

show Word comments by

Returns or sets the name of the reviewer whose comments are shown in the comments pane.
You can choose to show comments either by a single reviewer or by all reviewers. To view comments by all reviewers, set this property to "All Reviewers". Read/write.

show grammatical errors

True if grammatical errors are marked by a wavy green line in the specified document. Read/write.

Note: To view grammatical errors in your document, you must set the check grammar as you type property of the Word options class to true.
Microsoft Word Suite

**show hidden bookmarks**
True if hidden bookmarks are included in the bookmarks list. This property also controls whether hidden bookmarks are listed in the Bookmarks dialog box (Insert menu). Read/write.

Hidden bookmarks are automatically inserted when cross-references are inserted into the document.

**show revisions**
True if tracked changes in the specified document are shown on the screen. Read/write.

**show spelling errors**
True if Word underlines spelling errors in the document. Read/write.

**Note** To view spelling errors in a document, you must set the check spelling as you type property of the Word options class to **true**.

**show summary**
True if an automatic summary is displayed for the specified document. Read/write.

**snap to grid**
True if AutoShapes are automatically aligned with an invisible grid when they are drawn, moved, or resized. Read/write.

You can temporarily override this setting by pressing OPTION while drawing, moving, or resizing an AutoShape.

**snap to shapes**
True if Word automatically aligns AutoShapes with invisible gridlines that go through the vertical and horizontal edges of other AutoShapes. Read/write.

This property creates additional invisible gridlines for each AutoShape. The snap to shapes property works independently of the snap to grid property.

**spelling checked**
True if spelling has been checked throughout the specified document. False if all or some of the document hasn’t been checked for spelling. Read/write.

To recheck the spelling in a range or document, set the spelling checked property to false.

**subdocuments expanded**
True if the subdocuments in the specified document are expanded. Read/write.

**summary length**
Returns or sets the length of the summary as a percentage of the document length. The larger the number, the more detail is included in the summary. Read/write.

**Note** This property takes effect immediately if the AutoSummarize toolbar is displayed; otherwise, it takes effect the next time the auto summarize command or the summary view mode property is applied to the document.
**summary view mode**

Returns or sets the way a summary is displayed. This property corresponds to **Type of summary** in the **AutoSummarize** dialog box (Tools menu). Read/write.

Can be one of the following:

<table>
<thead>
<tr>
<th>Constant</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>summary mode highlight</td>
<td>Highlights the key points in the specified document and displays the <strong>AutoSummarize</strong> toolbar.</td>
</tr>
<tr>
<td>summary mode insert</td>
<td>Inserts a summary at the beginning of the specified document.</td>
</tr>
<tr>
<td>summary mode create new</td>
<td>Creates a new document and inserts the specified summary.</td>
</tr>
<tr>
<td>summary mode hide all but</td>
<td>Hides everything except the specified summary and displays the <strong>AutoSummarize</strong> toolbar.</td>
</tr>
</tbody>
</table>

**text object**

Returns a **text range** object that represents the main document story. Read-only.

**track revisions**

**True** if changes are tracked in the specified document. Read/write.

**update styles on open**

**True** if the styles in the specified document are updated to match the styles in the attached template each time the document is opened. Read/write.

**web options**

Returns the **web options** object, which contains document-level attributes used by Word when you save a document as a Web page or open a Web page. Read-only.

**write password**

Sets a password for saving changes to the specified document. Write-only.

**write reserved**

**True** if the specified document is protected with a write password. Read-only.
Class: document version

Plural
document versions

Represents a single version of a document. The document versions list includes all the versions of the specified document.

Use document version index, where index is the index number, to return a single document version object. The index number represents the position of the version in the document versions list. The first version added to the document versions list is index number 1. The following example displays the comment, author, and date of the first version of the active document.

if (count of document versions of active document) ≥ 1 then
    set v1 to document version 1 of active document
    display dialog "Comment = " & (comment of v1) & return & ¬
    "Author = " & (saved by of v1) & return & "Date = " & (date value of v1)
end if

Use the save version command to add an item to the document versions list. The following example adds a version of the active document with the specified comment.

save version active document comment "incorporated Judy's revisions"

Properties

comment

Returns the comment associated with the specified version of a document. Read-only.

date value

The date and time that the document version was saved. Read-only.

entry index

Returns a number that indicates the position of a document revision in the document revisions list. Read-only.

saved by

Returns the name of the user who saved the specified version of the document. Read-only.

Class: drop cap

Plural
drop caps

Represents a dropped capital letter at the beginning of a paragraph. Each paragraph object contains only one drop cap object.

To return a drop cap object, use the drop cap property of a paragraph object. The following example sets a dropped capital letter for the first letter in the first paragraph in the active document.

enable drop cap of paragraph 1 of active document
set position of drop cap of paragraph 1 of active document to drop normal
Properties

distance from text

Returns or sets the distance (in points) between the dropped capital letter and the paragraph text. Read/write.

drop position

Returns or sets the position of a dropped capital letter. Read/write.

Can be one of the following:

- drop margin
- drop none
- drop normal

font name

Returns or sets the name of the font for the dropped capital letter. Read/write.

lines to drop

Returns or sets the height (in lines) of the specified dropped capital letter. Read/write.

Class: drop down

Plural

drop downs

Elements

list entry

Represents a drop-down form field that contains a list of items in a form.

To return a single form field object, use form field index, where index is the index number or the bookmark name associated with the drop-down form field. To return a drop down object, use the drop down property with the form field object. The following example selects the first item in the drop-down form field named "DropDown" in the active document.

set drop down value of drop down of form field "dropdown1" of active document to 1

The index number represents the position of the form field in the form fields list. The following example checks the type of the first form field in the active document. If it's a drop-down form field, the second item is selected.

if form field type of form field 1 of active document is field form drop down then

set drop down value of drop down of form field 1 of active document to 2

end if
The following example determines whether the form field represented by ffield is a valid drop-down form field before adding an item to it.

```plaintext
set ffield to drop down of form field 1 of active document
if valid of ffield is true then
    make new list entry at ffield with properties {name:"Hello"}
else
    display dialog "The first field is not a drop down"
end if
```

To add a drop-down form field, use the `make` command with the `form field` class. The following example adds a drop-down form field at the beginning of the active document and then adds items to the form field.

```plaintext
set ffield to make new form field at the beginning of active document with ¬
    properties {text object:(create range active document start 0 ¬
        end 0), form field type:field form drop down}
set name of ffield to "Colors"
make new list entry at drop down of ffield with properties {name:"Blue"}
make new list entry at drop down of ffield with properties {name:"Green"}
make new list entry at drop down of ffield with properties {name:"Red"}
```

**Properties**

- **drop down default**
  Returns or sets the default drop-down item. The first item in a drop-down form field is 1, the second item is 2, and so on. Read/write.

- **drop down value**
  Returns or sets the number of the selected item in a drop-down form field. Read/write.

- **valid**
  True if the specified form field object is a valid check box form field, drop down form field, or text form field. False if it isn't valid. Read-only.

**Class:** endnote

**Plural**

- **endnotes**

  Represents an endnote. The endnotes list represents the endnotes in a selection, text range, or document.
Use `endnote index`, where `index` is the index number, to return a single `endnote` object. The index number represents the position of the endnote in the selection, text range, or document. The following example applies red formatting to the first endnote in the selection.

```vbnet
if (count of endnotes of selection) ≥ 1 then
    set color index of font object of note reference of endnote 1 of selection to red
end if
```

To add an endnote to the `endnotes` list, use the `make` command. The following example adds an endnote immediately after the selection.

```vbnet
collapse range text object of selection direction collapse end
make new endnote at the end of selection
```

### Properties

**entry index**

Returns a number that indicates the position of an endnote in the endnotes list. Read-only.

**note reference**

Returns a `text range` object that represents an endnote mark. Read-only.

**text object**

Returns a `text range` object that represents the portion of a document that's contained in the endnote object. Read-only.

### Class: `endnote options`

Represents options for endnotes in a selection, text range, or document.

To return the `endnote options` object, use the `endnote options` property of the `selection`, `text range`, or `document` object. The following example sets the location of endnotes in the active document.

```vbnet
set endnote location of endnote options of active document to end_of_section
```

### Properties

**endnote continuation notice**

Returns a `text range` object that represents the endnote continuation notice. Read-only.

**endnote continuation separator**

Returns a `text range` object that represents the endnote continuation separator. Read-only.

**endnote location**

Returns or sets the position of all endnotes. Read/write.

Can be either of the following:

- `end_of_section`
- `end_of_document`
endnote number style

Returns or sets the number style for the specified object. Read/write.

Can be one of the following:

- note number style arabic
- note number style lowercase letter
- note number style lowercase roman
- note number style symbol
- note number style uppercase letter
- note number style uppercase roman
- note number style arabic fill width
- note number style hanja read
- note number style hanja read digit
- note number style kanji
- note number style kanji digit
- note number style kanji traditional
- note number style number in circle
- note number style simp chin num1
- note number style simp chin num2
- note number style simp chin num1
- note number style simp chin num2
- note number style trad chin num1
- note number style trad chin num2
- note number style simp chin num1
- note number style simp chin num2
- note number style trad chin num1
- note number style trad chin num2

endnote numbering rule

Returns or sets the way endnotes are numbered after page breaks or section breaks. Read/write.

Can be one of the following:

- restart continuous
- restart section
- restart page

endnote separator

Returns a text range object that represents the endnote separator. Read-only.

endnote starting number

Returns or sets the starting note number. Read/write.

Class: envelope

Plural

envelopes

Represents an envelope. Each document object contains only one envelope object.

To return the envelope object, use the envelope object property. The following example adds an envelope to a new document and sets the distance between the top of the envelope and the address to 2 inches.

set myDoc to make new document
set addr to "Joseph Matthews" & return & "123 Skye St." & return & ¬ "Redmond, WA 98107"
set retaddr to "Wendy Kahn" & return & "456 Erde Lane" & return & ¬ "Redmond, WA 98107"
set myEnv to envelope object of myDoc
insert envelope data myEnv address addr return address retaddr
set address from top of myEnv to (inches to points inches 2.0)
Remarks
The envelope object is available regardless of whether an envelope has been added to the specified document. However, an error occurs if you use one of the following properties when an envelope hasn't been added to the document: address, address from left, address from top, feed source, return address, return address from left, and return address from top.

To add an envelope to the specified document, use the insert envelope data command. To set the properties of an envelope and print it without adding it to the document, use the print out envelope command.

Properties

address
Returns the envelope delivery address as a text range object. Read-only.

address from left
Returns or sets the distance (in points) between the left edge of the envelope and the delivery address. Read/write.

Note If you use this property before an envelope has been added to the document, an error occurs.

address from top
Returns or sets the distance (in points) between the top edge of the envelope and the delivery address. Read/write.

Note If you use this property before an envelope has been added to the document, an error occurs.

address style
Returns a Word style object that represents the delivery address style for the envelope. Read-only

Note If an envelope is added to the document, text formatted with the Envelope Address style is automatically updated.

default face up
True if envelopes are fed face up by default. Read/write.

default height
Returns or sets the default envelope height, in points. Read/write.

Note The default height and default width properties, if set, specify a custom envelope size for printing, as defined in the Envelope Size dialog box (Envelopes command, Tools menu). To set the default size to a predefined size, use the default size property.

default omit return address
True if the return address is omitted from envelopes by default. Read/write.
**default orientation**

Returns or sets the default orientation for feeding envelopes. Read/write.

Can be one of the following:

- center clockwise
- center landscape
- center portrait
- left clockwise
- left landscape
- left portrait
- right clockwise
- right landscape
- right portrait

**default print FIMA**

True to add a Facing Identification Mark (FIM-A) to envelopes by default. Read/write.

**Note**  For U.S. mail only. A FIM-A code is used to presort courtesy reply mail. The default print bar code property must be set to true before this property is set.

**default print bar code**

True if a POSTNET bar code is added to envelopes or mailing labels by default. Read/write.

**Note**  For U.S. mail only. For envelopes, this property must be set to true before the default print FIMA property is set.

**default size**

Returns or sets the default envelope size. Read/write.

**Note**  The string that's returned corresponds to the right-hand side of the string that appears in the Envelope Size box in the Envelope dialog box (Tools menu, Envelopes command, Custom button). If you set either the default height or default width property, the envelope size is automatically changed to Custom Size in the Envelope dialog box, and this property returns "Custom size."

**default width**

Returns or sets the default envelope width, in points. Read/write.

**Note**  If you set the default height or default width property, the envelope size is automatically changed to Custom Size in the Envelopes dialog box (Tools menu). To set the default size to a predefined size, use the default size property.
feed source

Returns or sets the paper tray for the envelope. Read/write.

Can be one of the following:

- printer automatic sheet feed
- printer default bin
- printer envelope feed
- printer form source
- printer large capacity bin
- printer large format bin
- printer lower bin
- printer manual envelope feed
- printer manual feed
- printer middle bin
- printer only bin
- printer paper cassette
- printer small format bin
- printer tractor feed
- printer upper bin

Note If you use this property before an envelope has been added to the document, an error occurs.

return address

Returns a text range object that represents the envelope return address. Read-only.

return address from left

Returns or sets the distance (in points) between the left edge of the envelope and the return address. Read/write.

Note If you use this property before an envelope has been added to the document, an error occurs.

return address from top

Returns or sets the distance (in points) between the top edge of the envelope and the return address. Read/write.

Note If you use this property before an envelope has been added to the document, an error occurs.

return address style

Returns a Word style object that represents the return address style for the envelope. Read-only

Note If an envelope is added to the document, text formatted with the Envelope Return style is automatically updated.

Class: field

Plural

fields

Represents a field. The fields list represents the fields in a selection, text range, or document.

Use field index, where index is the index number, to return a single field object. The index number represents the position of the field in the selection, text range, or document. The following example displays the field code and the result of the first field in the active document.
if (count of fields of active document) ≥ 1 then
    display dialog ("Code = " & content of field code of field 1 of ¬
    active document as string) & return & "Result = " & content of ¬
    result range of field 1 of active document as string
end if

To add a field to the fields list, use the make command. The following example inserts a DATE field
at the beginning of the selection and then displays the result. This example uses the hidden text
range property of the field object to specify the location of the field.

collapse range text object of selection direction collapse start
set myField to make new field at active document with properties ¬
    {text range:text object of selection, field type:field date}
display dialog (content of result range of myField) as string

Properties

   entry index

      Returns a number that indicates the position of the field in the fields list. Read-only.

   field code

      Returns a text range object that represents a field's code. A field's code is everything that's
      enclosed by the field characters ({}), including the leading space and trailing space characters.
      Read/write.

   field kind

      Returns the type of link for a field object. Read-only.

      Can be one of the following:

      | Constant        | Description                                                                 |
      |-----------------|-----------------------------------------------------------------------------|
      | field kind hot  | A field that's automatically updated each time it's displayed or each time   |
      |                 | the page is reformatted, but which can also be manually updated (for example, |
      |                 | INCLUDEPICTURE or FORMDROPDOWN).                                            |
      | field kind warm | A field that can be updated and has a result. This type includes fields       |
      |                 | that are automatically updated when the source changes as well as fields     |
      |                 | that can be manually updated (for example, DATE or INCLUDETEXT).             |
      | field kind none | An invalid field (for example, a pair of field characters with nothing inside).|
      | field kind cold | A field that doesn't have a result (for example, XE (Index Entry) fields,    |
      |                 | TC (Table of Contents Entry) fields, or Private fields).                     |
field text

Returns or sets data in an ADDIN field. Read/write.

Note The data is not visible in the field code or result; it is only accessible by returning the value of the field text property. If the field isn't an ADDIN field, this property will return an error.

field type

Returns the field type. Can be one of the Word fields. Read-only.

inline shape

Returns an inline shape object that represents the picture or OLE object that is the result of an INCLUDEPICTURE or EMBED field. Read-only.

An inline shape object is treated like a character and is positioned as a character within a line of text.

link format

Returns a link format object that represents the link options of the specified field, inline shape, or shape that's linked to a file. Read-only.

locked

True if the specified field is locked. When a field is locked, you cannot update the field results. Read/write.

next field

Returns the next field in the fields list. Read-only.

previous field

Returns the previous field in the fields list. Read-only.

result range

Returns a text range object that represents a field's result. You can access a field result without changing the view from field codes. Read/write.

Note To return text from a text range object, use the content property.

show codes

True if field codes instead of field results are displayed for the specified field. Read/write.

Class: file converter

Plural

file converters

Represents a file converter that's used to open or save files. The file converters list contains all the installed file converters for opening and saving files.

Use file converter index, where index is a class name or index number, to return a single file converter object. The index number represents the position of the file converter in the file converters list. The following example displays the format name of the first file converter.

display dialog format name of file converter 1 as string
You cannot create a new file converter or add one to the **file converters** list. **File converter** objects are added during setup of Microsoft Office or when you install supplemental file converters. To determine whether a **file converter** object can be used to open or save a document, use either the **can save** or **can open** property.

**Remarks**
File converters for saving documents are listed in the **Save As** dialog box. File converters for opening documents appear in a dialog box if the **Confirm conversion at Open** check box is selected in the **General** pane of the **Preferences** dialog box (**Word** menu).

**Properties**
- **can open**
  - *True* if the specified file converter is designed to open files. Read-only.
  - **Note** The **can save** property returns *true* if the specified file converter can be used to save (export) files.
- **can save**
  - *True* if the specified file converter is designed to save files. Read-only.
  - **Note** The **can open** property returns *true* if the specified file converter can be used to open (import) files.
- **class name**
  - Returns a unique name that identifies the file converter. Read-only.
- **extensions**
  - Returns the file name extensions associated with the specified **file converter** object. Read-only.
- **format name**
  - Returns the name of the specified file converter. The format names appear in the **Save as type** box in the **Save As** dialog box (**File** menu). Read-only.
- **name**
  - Returns the file converter's file name. Read-only.
- **open format**
  - Returns the file format of the specified file converter as a unique number that represents the file converter. Read-only.
- **path**
  - Returns the path to the specified **file converter** object. Read-only.
  - **Note** The path doesn't include a trailing character; for example, Macintosh HD:Users:Shared.
- **save format**
  - Returns the file format of the specified document or file converter as a unique number that represents the file converter. Read-only.
Class: find

Plural
finds

Represents the criteria for a find operation. The properties of the find object correspond to the options in the Find and Replace dialog box.

To return a find object, use the find object property of the text range or selection object class. The following example finds and selects the next occurrence of the word "hi."

```plaintext
set myFind to find object of selection
clear formatting myFind set content of myFind to "hi"
execute find myFind with match forward
```

The following example finds all occurrences of the word "hi" in the active document and replaces the word with "hello."

```plaintext
set myRange to text object of active document
execute find (find object of myRange) find text "hi" replace with "hello" ~
    replace replace all
```

Remarks

If you've gotten to the find object from the selection object, the selection is changed when text matching the find criteria is found. The following example selects the next occurrence of the word "blue."

```plaintext
execute find (find object of selection) find text "blue" with match forward
```

Properties

all documents

True if the find operation searches all open documents for the text to find. Read/write.

Note To specify the text to be located in a document, use the content property of the find object or use the find text argument with the execute find command.

content

Returns or sets the text to find or replace in the specified range or selection. Read/write.

font object

Returns or sets a font object that represents the character formatting of the find object. To set this property, specify an expression that returns a font object. Read/write.

format

True if formatting is included in the find operation. Read/write.

forward

True if the find operation searches forward through the document. False if it searches backward through the document. Read/write.
found

True if the search produces a match. Read-only.

frame

Returns a frame object that represents the frame formatting for the specified find-and-replace operation. Read-only.

highlight

True if highlight formatting is included in the find criteria. Read/write.

language id

Returns or sets the language for the specified object. Read/write.

Can be one of the following:

<table>
<thead>
<tr>
<th>Language</th>
<th>Language</th>
</tr>
</thead>
<tbody>
<tr>
<td>afrikaans</td>
<td>language none</td>
</tr>
<tr>
<td>arabic</td>
<td>latvian</td>
</tr>
<tr>
<td>basque</td>
<td>macedonian</td>
</tr>
<tr>
<td>belgian dutch</td>
<td>malaysian</td>
</tr>
<tr>
<td>belgian french</td>
<td>mexicanSpanish</td>
</tr>
<tr>
<td>brazilian portuguese</td>
<td>language no proofing</td>
</tr>
<tr>
<td>bulgarian</td>
<td>norwegian bokmøl</td>
</tr>
<tr>
<td>byelorussian</td>
<td>norwegian nynorsk</td>
</tr>
<tr>
<td>catalan</td>
<td>polish</td>
</tr>
<tr>
<td>croatian</td>
<td>portuguese</td>
</tr>
<tr>
<td>czech</td>
<td>romanian</td>
</tr>
<tr>
<td>danish</td>
<td>russian</td>
</tr>
<tr>
<td>dutch</td>
<td>serbian cyrillic</td>
</tr>
<tr>
<td>english aus</td>
<td>serbian latin</td>
</tr>
<tr>
<td>english canadian</td>
<td>sesotho</td>
</tr>
<tr>
<td>english new zealand</td>
<td>simplified chinese</td>
</tr>
<tr>
<td>english south africa</td>
<td>slovak</td>
</tr>
<tr>
<td>english uk</td>
<td>slovenian</td>
</tr>
<tr>
<td>english us</td>
<td>spanish</td>
</tr>
<tr>
<td>estonian</td>
<td>spanish modern sort</td>
</tr>
<tr>
<td>farsi</td>
<td>swedish</td>
</tr>
<tr>
<td>finnish</td>
<td>swiss french</td>
</tr>
<tr>
<td>french</td>
<td>swiss german</td>
</tr>
<tr>
<td>french canadian</td>
<td>swiss italian</td>
</tr>
<tr>
<td>german</td>
<td>traditional chinese</td>
</tr>
<tr>
<td>greek</td>
<td>tsonga</td>
</tr>
<tr>
<td>hebrew</td>
<td>tswana</td>
</tr>
<tr>
<td>hungarian</td>
<td>turkish</td>
</tr>
<tr>
<td>italian</td>
<td>ukrainian</td>
</tr>
<tr>
<td>icelandic</td>
<td>venda</td>
</tr>
<tr>
<td>japanese</td>
<td>xhosa</td>
</tr>
<tr>
<td>korean</td>
<td>zulu</td>
</tr>
</tbody>
</table>
language id east asian

Returns or sets an East Asian language for the specified object. Can be any of the language id constants listed above. Read/write.

Note This is the recommended way to apply an East Asian language to text in a document created in an East Asian version of Word.

match all word forms

True if all forms of the text to find are found by the find operation (for instance, if the text to find is "sit," "sat" and "sitting" are found as well). Read/write.

Note To specify the text to be searched for in a document, use the content property of the find object or use the find text argument with the execute find command.

match byte

True if Word distinguishes between full-width and half-width letters or characters during a search. Read/write.

match case

True if the find operation is case sensitive. Read/write.

match fuzzy

True if Word uses the nonspecific search options for Japanese text during a search. Read/write.

match sounds like

True if words that sound similar to the text to find are found by the find operation. Read/write.

match whole word

True if the find operation locates only entire words and not text that's part of a larger word. Read/write.

match wildcards

True if the text to find contains search wildcards. Corresponds to the Use wildcards check box in the Find and Replace dialog box (Edit menu). Read/write.

no proofing

True if Word finds or replaces text that the spelling and grammar checker ignores. Read/write.

paragraph format

Returns or sets a paragraph format object that represents the paragraph settings for the find operation. Read/write.

replacement

Returns a replacement object that contains the criteria for a replace operation. Read-only.
**style**

Returns or sets the style for the specified object. To set this property, specify either the local name of the style, an integer, a built-in Word style, or an object that represents the style. Read/write.

**supplemental language id**

Returns or sets the language for the specified object. Can be any of the language id constants listed above. Read/write.

**Note** This is the recommended way to determine the language of a document created in an East Asian version of Word.

**wrap**

Returns or sets what happens if the search begins at a point other than the beginning of the document and the end of the document is reached (or vice versa if forward is set to false) or if the search text isn’t found in the specified selection or range. Read/write.

Can be one of the following:

<table>
<thead>
<tr>
<th>Constant</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>find ask</td>
<td>After searching the selection or range, Word displays a message asking whether to search the remainder of the document.</td>
</tr>
<tr>
<td>find continue</td>
<td>The find operation continues when the beginning or end of the search range is reached.</td>
</tr>
<tr>
<td>find stop</td>
<td>The find operation ends when the beginning or end of the search range is reached.</td>
</tr>
</tbody>
</table>

**Class: font**

**Plural fonts**

Contains font attributes (font name, font size, color, and so on) for an object.

Use the font object property to return the font object. The following instruction applies bold formatting to the selection.

set bold of font object of selection to true

The following example formats the first paragraph in the active document as 24 point Arial and italic.

set myRange to text object of paragraph 1 of active document
set bold of font object of myRange to true
set name of font object of myRange to "Arial"
set size of font object of myRange to 24
Microsoft Word Suite

The following example changes the formatting of the Heading 2 style in the active document to Arial and bold.

```
set H2 to font object of Word style style heading2 of active document
set name of H2 to "Arial"
set bold of H2 to true
```

**Properties**

**all caps**

*True* if the font is formatted as all capital letters. Read/write.

Setting *all caps* to *true* sets *small caps* to *false*, and vice versa.

**animation**

Returns or sets the type of animation applied to the font. Read/write.

Can be one of the following:

- animation blinking background
- animation las vegas lights
- animation marching black ants
- animation none
- animation shimmer
- animation sparkle text
- animation marching red ants

**ascii name**

Returns or sets the font used for characters whose character set numbers are from 0 (zero) through 127. Read/write.

**Note** In the U.S. English version of Microsoft Word, this property defaults to Times New Roman. To change the font that's applied to the text and that appears on the Formatting toolbar, use the *name* property.

**bold**

*True* if the font is formatted as bold. Read/write.

**border options**

Returns a *border options* object that represents options for the borders for the *font* object. Read/write.

**color**

Returns or sets the RGB color for the specified *font* object. Read/write.
color index

Returns or sets the color for the specified border or font object. Read/write.

Can be one of the following:

- auto
- black
- blue
- bright
- green
- dark blue
- dark red
- dark yellow
- gray25
- gray50
- green
- no highlight
- pink
- red
- teal
- turquoise
- violet
- white
- yellow

disable character space grid

True if Word ignores the setting for the number of characters per line in the Document Grid.
Read/write.

double strike through

True if the specified font is formatted as double strikethrough text. Read/write.

Note To set or return single-line strikethrough formatting, use the strike through property. Setting double strike through to true sets strike through to false, and vice versa.

east asian name

Returns or sets an East Asian font name. Read/write.

Note In the U.S. English version of Microsoft Word, this property defaults to Times New Roman. This is the recommended way to apply an East Asian font to text in a document created in an East Asian version of Word.

emboss

True if the specified font is formatted as embossed. Read/write.

Setting emboss to true sets engrave to false, and vice versa.

emphasis mark

Returns or sets the emphasis mark for a character or designated character string. Read/write.

Can be one of the following:

- emphasis mark none
- emphasis mark over comma
- emphasis mark over solid circle
- emphasis mark over white circle
- emphasis mark under solid circle
Microsoft Word Suite

**engrave**

*True* if the font is formatted as engraved. Read/write.

Setting *engrave* to *true* sets *emboss* to *false*, and vice versa.

**font position**

Returns or sets the position of text (in points) relative to the base line. A positive number raises the text, and a negative number lowers it. Read/write.

**font size**

Returns or sets the font size, in points. Read/write.

**hidden**

*True* if the font is formatted as hidden text. Read/write.

To control the display of hidden text, use the *show hidden text* property of the *view* object.

To control whether properties and commands that return *text range* objects include or exclude hidden text when hidden text isn't displayed, use the *include hidden text* property of the *text retrieval mode* object.

**italic**

*True* if the font or range is formatted as italic. Read/write.

**kerning**

Returns or sets the minimum font size for which Word will adjust kerning automatically. Read/write.

**name**

Returns or sets the name of the font. Read/write.

**other name**

Returns or sets the font used for characters whose character set numbers are greater than 127. Read/write.

**Note** In the U.S. English version of Microsoft Word, this property defaults to Times New Roman. To change the font that's applied to the text and that appears on the *Formatting* toolbar, use the *name* property.

**outline**

*True* if the font is formatted as outline. Read/write.

**scaling**

Returns or sets the scaling percentage applied to the font. This property stretches or compresses text horizontally as a percentage of the current size (the scaling range is from 1 through 600). Read/write.

**shading**

Returns a *shading* object that refers to the shading formatting for the specified object. Read-only.
shadow

True if the specified font is formatted as shadowed. Read/write.

small caps

True if the font is formatted as small capital letters. Read/write.

Setting small caps to true sets all caps to false, and vice versa.

spacing

Returns or sets the spacing between characters, in points. Read/write.

strikethrough

True if the font is formatted as strikethrough text. Read/write.

Note To set or return double strikethrough formatting, use the double strike through property.

subscript

True if the font is formatted as subscript. Read/write.

Setting subscript to true sets superscript to false, and vice versa.

superscript

True if the font is formatted as superscript. Read/write.

Setting superscript to true sets subscript to false, and vice versa.

underline

Returns or sets the type of underline applied to the font or range. Read/write.

Can be one of the following:

- underline none
- underline dash
- underline dot dash
- underline dot dot dash
- underline thick
- underline dotted
- underline double
- underline single
- underline words
- underline wavy

underline color

Returns or sets the RGB color of the underline for the specified font object. Read/write.
Class: footnote

Plural
footnotes

Represents a footnote positioned at the bottom of the page or beneath text. The footnotes list represents the footnotes in a selection, text range, or document.

Use footnote index, where index is the index number, to return a single footnote object. The index number represents the position of the footnote in the selection, range, or document. The following example applies red formatting to the first footnote in the selection.

```plaintext
if (count of footnotes of selection) ≥ 1 then
    set color index of font object of note reference of footnote 1 of selection to red
end if
```

To add a footnote to the footnotes list, use the make command. The following example inserts an automatically numbered footnote immediately after the selection.

```plaintext
collapse range text object of selection direction collapse end
set myFootnote to make new footnote at active document with properties ¬
    {text range: text object of selection}
set content of text object of myFootnote to "The Willow Tree, (Lone Creek Press, 1996)."
```

Remarks
Footnotes positioned at the end of a document or section are considered endnotes and are included in the endnotes list.

Properties
entry index

Returns a number that indicates the position of the footnote in the footnotes list. Read-only.

note reference

Returns a text range object that represents a footnote mark. Read-only.

text object

Returns a text range object that represents the portion of a document that's contained in the footnote object. Read-only.

Class: footnote options

Represent all the options for footnotes in a selection, text range, or document.

Use the footnote options property to return the footnote options object. The following example changes all of the footnotes in the active document to endnotes.

```plaintext
swap with endnotes footnote options of active document
```
Properties

footnote continuation notice

Returns a text range object that represents the footnote continuation notice. Read-only.

footnote continuation separator

Returns a text range object that represents the footnote continuation separator. Read-only.

footnote location

Returns or sets the position of all footnotes. Read/write.

Can be either of the following:

- end_of_section
- end_of_document

footnote number style

Returns or sets the number style for the footnote object. Read/write.

Can be one of the following:

- note number style arabic
- note number style lowercase letter
- note number style lowercase roman
- note number style symbol
- note number style uppercase letter
- note number style uppercase roman
- note number style arabic fill width
- note number style hanja read
- note number style hanja read digit

- note number style kanji
- note number style kanji digit
- note number style kanji traditional
- note number style number in circle
- note number style simp chin num1
- note number style simp chin num2
- note number style trad chin num1
- note number style trad chin num2

footnote numbering rule

Returns or sets the way footnotes are numbered after page breaks or section breaks. Read/write.

Can be one of the following:

- restart continuous
- restart section
- restart page

footnote separator

Returns a text range object that represents the footnote separator. Read-only.

footnote starting number

Returns or sets the starting note number. Read/write.
**Class: form field**

**Plural**

form fields

Represents a single form field.

Use `form field` index, where index is a bookmark name or index number, to return a single `form field` object. The following example sets the result of the Text1 form field to "Don Funk."

```
set form field result of form field "Text1" of active document to "Don Funk"
```

The index number represents the position of the form field in the selection, text range, or document. The following example displays the name of the first form field in the active document.

```
if (count of form fields of active document) ≥ 1 then
    display dialog name of form field 1 of active document as string
end if
```

To add a `form field` object, use the `make` command. The following example adds a check box at the beginning of the active document and then selects the check box.

```
set ffield to make new form field at active document with properties ¬
    { text object:(create range active document start 0 end 0), form ¬
      field type:field form check box }
set check box value of check box of ffield to true
```

**Remarks**

To return the `check box`, `drop down`, and `text input` objects, use the `check box`, `drop down`, and `text input` properties with the `form field` object. The following example selects the check box named "Check1."

```
set check box value of check box of form field "Check1" of active document ¬
to true
```

**Properties**

* `calculate on exit`

  * `True` if references to the specified form field are automatically updated whenever the field is exited. Read/write.

  A REF field can be used to reference the contents of a form field. For example, `[REF SubTotal]` references the form field marked by the SubTotal bookmark.

* `check box`

  Returns a `check box` object that represents a check box form field. Read-only.

  If the `check box` property is applied to a `form field` object that isn't a check box form field, the property won't fail, but the `valid` property for the returned object will be `false`. 

121
**Microsoft Word Suite**

*drop down*

Returns a *drop down* object that represents a drop-down form field. Read-only.

If the *drop down* property is applied to a *form field* object that isn't a drop-down form field, the property won't fail, but the *valid* property for the returned object will be *false*.

*enabled*

*True* if a form field is enabled. If a form field is enabled, its contents can be changed as the form is filled in. Read/write.

*entry macro*

Returns or sets an entry macro name for the specified form field (*check box*, *drop down*, or *text input*). The entry macro runs when the form field gets the focus. Read/write.

*exit macro*

Returns or sets an exit macro name for the specified form field (*check box*, *drop down*, or *text input*). The exit macro runs when the form field loses the focus. Read/write.

*form field result*

Returns a string that represents a field's result. You can access a field result without changing the view from field codes. Read/write.

*form field type*

Returns the field type. Can be one of the Word fields. Read-only.

*help text*

Returns or sets the text that's displayed in a message box when the form field has the focus and the user presses COMMAND+/ or HELP. If the *own help* property is set to *true*, *help text* specifies the text string value. If the *own help* property is set to *false*, *help text* specifies the name of an AutoText entry that contains help text for the form field. Read/write.

*name*

Returns or sets the name of the form field. Read/write.

*next form field*

Returns the next form field in the form fields list. Read-only.

*own help*

Specifies the source of the text that's displayed in a message box when a form field has the focus and the user presses COMMAND+/ or HELP. If *true*, the text specified by the *help text* property is displayed. If *false*, the text in the AutoText entry specified by the *help text* property is displayed. Read/write.

*own status*

Specifies the source of the text that's displayed in the status bar when a form field has the focus. If *true*, the text specified by the *status text* property is displayed. If *false*, the text of the AutoText entry specified by the *status text* property is displayed. Read/write.
previous form field

    Returns the previous form field in the form fields list. Read-only.

status text

    Returns or sets the text that's displayed in the status bar when a form field has the focus. If the own status property is set to true, status text specifies the status bar text. If the own status property is set to false, status text specifies the name of an AutoText entry that contains status bar text for the form field. Read/write.

text input

    Returns a text input object that represents a text form field. Read-only.

    If the text input property is applied to a form field object that isn't a drop-down form field, the property won't fail, but the valid property for the returned object will be false.

    To return or set the contents of a text input object, use the form field result property with the form field object.

text object

    Returns a text object that represents the portion of a document that's contained in the form field object. Read-only.

Class: frame

Plural

frame

    Represents a frame. The frames list includes all frames in a selection, text range, or document.

Use frame index, where index is the index number, to return a single frame object. The index number represents the position of the frame in the selection, range, or document. The following example allows text to wrap around the first frame in the active document.

    set text wrap of frame 1 of active document to true

    To add a frame around a range, use the make command. The following example adds a frame around the first paragraph in the active document.

    make new frame at active document with properties ¬
        { t e x t  o b j e c t : t e x t  o b j e c t  o f  p a r a g r a p h  1  o f  a c t i v e  d o c u m e n t }

Remarks

    You can wrap text around shape objects by using the wrap format property. You can position a shape object by using the top and left properties.

Properties

border options

    Returns a border options object that represents options for the borders of the frame object. Read/write.

height

    Returns or sets the height of the specified frame. Read/write.
**height rule**

Returns or sets the rule for determining the height of the specified frame. Read/write.

Can be one of the following:

<table>
<thead>
<tr>
<th>Constant</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>frame auto</td>
<td>Sets the height according to the width of the item in the frame.</td>
</tr>
<tr>
<td>frame exact</td>
<td>Sets the height to an exact value specified by the height property.</td>
</tr>
<tr>
<td>frame at least</td>
<td>Sets the height to a value equal to or greater than the value specified by the height property.</td>
</tr>
</tbody>
</table>

**horizontal distance from text**

Returns or sets the horizontal distance (in points) between a frame and the surrounding text. Read/write.

**horizontal position**

Returns or sets the horizontal distance between the edge of the frame and the item specified by the relative horizontal position property. Can be a number that indicates a measurement in points. Read/write.

**lock anchor**

True if the specified frame is locked. The frame anchor indicates where the frame will appear in Normal view. You cannot reposition a locked frame anchor. Read/write.

**relative horizontal position**

Specifies what the horizontal position of a frame is relative to. Read/write.

Can be one of the following:

- relative horizontal position column
- relative horizontal position margin
- relative horizontal position page

**relative vertical position**

Specifies what the vertical position of a frame is relative to. Read/write.

Can be one of the following:

- relative horizontal position paragraph
- relative horizontal position margin
- relative horizontal position page

**shading**

Returns a shading object that refers to the shading formatting for the specified object. Read-only.
text object

Returns a text range object that represents the portion of a document that's contained in the frame object. Read-only.

text wrap

True if document text wraps around the specified frame. Read/write.

vertical distance from text

Returns or sets the vertical distance (in points) between a frame and the surrounding text. Read/write.

vertical position

Returns or sets the vertical distance between the edge of the frame and the item specified by the relative vertical position property. Can be a number that indicates a measurement in points. Read/write.

width

Returns or sets the width (in points) of the specified object. Read/write.

width rule

Returns or sets the rule used to determine the width of a frame. Read/write.

Can be one of the following:

<table>
<thead>
<tr>
<th>Constant</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>frame auto</td>
<td>Sets the width according to the width of the item in the frame.</td>
</tr>
<tr>
<td>frame exact</td>
<td>Sets the width to an exact value specified by the width property.</td>
</tr>
<tr>
<td>frame at least</td>
<td>Sets the width to a value equal to or greater than the value specified by the width property.</td>
</tr>
</tbody>
</table>

Class: header footer

Plural

header footers

Represents a single header or footer. The header footers list includes all headers and footers in the specified document section.

To return a single header footer object, use the get header or get footer command. The following example changes the text of both the primary header and the primary footer in the first section of the active document.

set s1 to section 1 of active document
set content of text object of (get header s1 index header footer primary) ¬
to "Header text"
set content of text object of (get footer s1 index header footer primary) ¬
to "Footer text"
You can also return a single header footer object by using the header footer object property with a selection object.

**Note** You cannot add header footer objects to the header footers list.

**Remarks**

To specify a different first page, use the different first page header footer property with the page setup object. The following example inserts text into the first page footer in the active document.

```vbnet
set different first page header footer of page setup of active document to true
insert text "Written by Joe Smith" at text object of (get footer section 1 of active document index header footer first page)
```

To specify different odd and even page headers and footers, use the odd and even pages header footer property with the page setup object. If the odd and even pages header footer property is true, you can return an odd header or footer by using header footer primary, and you can return an even header or footer by using header footer even pages.

To add a page number to a header or footer, use the make command with the page numbers object. The following example adds page numbers to the primary footer in the first section of the active document.

```vbnet
make new page number at (get footer section 1 of active document index header footer primary)
```

**Properties**

**header footer index**

Represents the specified header or footer in the document or section. Read-only.

Can be one of the following:

- header footer primary
- header footer first page
- header footer even pages

**is header**

True if the specified header footer object is a header. Read-only.

**link to previous**

True if the specified header or footer is linked to the corresponding header or footer in the previous section. When a header or footer is linked, its contents are the same as in the previous header or footer. Read/write.

Because the link to previous property is set to true unless you change it, you can add headers, footers, and page numbers to your entire document by working with the headers, footers, and page numbers in the first section.

The link to previous property applies to each header or footer individually. For example, the link to previous property could be set to true for the even-numbered-page header but false for the even-numbered-page footer.
Microsoft Word Suite

page number options

Returns a page number options object that represents options for the page number fields included in the specified header or footer. Read-only.

text object

Returns a text range object that represents the portion of a document that's contained in the header footer object. Read-only.

Class: heading style

Plural

heading styles

Represents a style used to build a table of contents or figures.

Use heading style index, where index is the index number, to return a single heading style object. The index number represents the position of the style in the heading styles list. The following example adds (at the beginning of the active document) a table of figures built from the Title style, and then displays the name of the first style in the heading styles list.

set myTOF to make new table of figures at active document with properties ¬
    {text object:create range active document start 0 end 0, ¬
     use heading styles:true}
make new heading style at myTOF with properties {style:style title, level:2}
display dialog (style of heading style 1 of myTOF) as string

To add a style to the heading styles list, use the make command. The following example adds a table of contents at the beginning of the active document and then adds the Title style to the list of styles used to build a table of contents.

set myTOC to make new table of contents at active document with properties ¬
    {text object:create range active document start 0 end 0, ¬
     use heading styles:true, lower heading level:3, upper heading level:1}
make new heading style at myTOC with properties {style:"Title", level:2}

Properties

level

Returns or sets the level for the heading style in a table of contents or table of figures. Read/write.

style

Returns or sets the style for the specified object. To set this property, specify either the local name of the style, an integer, a built-in Word style, or an object that represents the style. Read/write.
Class: hyperlink object

Plural

hyperlink objects

Represents a hyperlink.

Use the hyperlink property to return a hyperlink object associated with a shape (a shape can have only one hyperlink). The following example activates the hyperlink associated with the first shape in the active document.

follow (hyperlink of shape 1 of active document)

Use hyperlink object index, where index is the index number, to return a single hyperlink object from a document, text range, or selection. The following example activates the first hyperlink in the selection.

if (count of hyperlink objects) of selection ≥ 1 then
   follow (hyperlink object 1 of selection)
end if

Properties

email subject

Returns or sets the text string for the specified hyperlink's subject line. The subject line is appended to the hyperlink's Internet address, or URL. Read/write.

This property is commonly used with e-mail hyperlinks. The value of this property takes precedence over any e-mail subject specified in the hyperlink address property of the same hyperlink object.

extra info required

True if extra information is required to resolve the specified hyperlink. Read-only.

Note You can specify extra information by using the extra info argument with the follow or follow hyperlink command. For example, you can use extra info to specify the coordinates of an image map, the contents of a form, or a file name.

hyperlink address

Returns the address (for example, a file name or URL) of the specified hyperlink. Read-only.

hyperlink type

Returns the hyperlink type. Read-only.

Can be one of the following:

- hyperlink type inline shape
- hyperlink type text range
- hyperlink type shape

name

Returns the friendly name (as it appears in the History folder) of the hyperlink. Read-only.
screen tip

Returns or sets the text that appears as a screen tip when the mouse pointer is positioned over the specified hyperlink. Read/write.

shape

Returns a shape object for the specified hyperlink. If the hyperlink isn’t represented by a shape, an error occurs. Read-only.

sub address

Returns a named location in the destination of the specified hyperlink. Read/write.

Note The named location can be a bookmark in a Word document, a named cell or cell reference in an Excel worksheet, or a slide number in a PowerPoint presentation.

text object

Returns a text range object that represents the portion of a document that’s contained in the specified object. Read-only.

text to display

Returns or sets the specified hyperlink’s visible text in a document. Read/write.

Class: index

Plural

indexes

Represents a single index. The indexes list includes all the indexes in the specified document.

Use index index, where index is the index number, to return a single index object. The index number represents the position of the index object in the document. The following example updates the first index in the active document.

if (count of indexes of active document) ≥ 1 then
  update index 1 of active document
end if

To create an index and add it to the indexes list, use the make command. The following example creates an index at the end of the active document.

set iRange to end key selection move unit a story extend by moving
make new index at active document with properties ¬
  {text object:iRange, index type:index runin}

Properties

accented letters

True if the specified index contains separate headings for accented letters (for example, words that begin with "À" are under one heading and words that begin with "A" are under another). Read/write.
Microsoft Word Suite

**heading separator**

Returns or sets the text between alphabetic groups (entries that start with the same letter) in the index. Corresponds to the \h switch for an INDEX field. Read/write.

Can be one of the following:

- heading separator blank line
- heading separator letter
- heading separator letter full
- heading separator letter low
- heading separator none

**index filter**

Returns or sets a value that specifies how Word classifies the first character of entries in the specified index. Read/write.

Can be any of the following:

- index filter aiueo
- index filter akasatana
- index filter chosung
- index filter low
- index filter medium
- index filter full
- index filter none

**index type**

Returns or sets the index type. Read/write.

Can be either of the following:

- index indent
- index runin

**number of columns**

Sets or returns the number of columns for each page of an index. Read/write.

**Note** Specifying 0 (zero) sets the number of columns in the index to the same number as in the document.

**right align page numbers**

**True** if page numbers are aligned with the right margin in an index, table of contents, or table of figures. Read/write.
sort by

Returns or sets the sorting criteria for the specified index. Can be either of the following: index sort by stroke (sort by the number of strokes in a character) or index sort by syllable (sort phonetically). Read/write.

tab leader

Returns or sets the character between entries and their page numbers in an index, table of authorities, table of contents, or table of figures. Read/write.

Can be one of the following:

- tab leader dashes
- tab leader dots
- tab leader heavy
- tab leader middle dot
- tab leader lines
- tab leader spaces

text object

Returns a text range object that represents the portion of a document that's contained in the index object. Read-only.

Class: key binding

Plural key bindings

Represents a custom key assignment in the current context. Custom key assignments are made in the Customize Keyboard dialog box.

To return a single key binding object, use the find key command. The following example displays the command assigned to the F1 key.

```
set customization context to normal template
display dialog (get command of (find key key code (build key code ¬
key1 f1_key)))
```

Properties

binding context

Returns an object that represents the storage location of the specified key binding. This property can return a document, template or application object. Read-only.

Note Built-in key assignments (for example, CTRL+I for Italic) return the application object as the context. Any key bindings you add will return a document or template object, depending on the customization context in effect when the key binding object was added.
binding key string

Returns the key combination string for the specified keys (for example, CTRL+SHIFT+A). Read-only.

command

Returns the command assigned to the specified key combination. Read-only.

command parameter

Returns the command parameter assigned to the specified shortcut key. Read-only.
To return the command name assigned to the specified shortcut key, use the command property.

key category

Returns the type of item assigned to the specified key binding. Read-only.
Can be one of the following:

- key category auto text
- key category command
- key category disable
- key category font
- key category macro
- key category nil
- key category prefix
- key category style
- key category symbol

key code

Returns a unique number for the first key in the specified key binding. Read-only.

Note You create this number by using the build key code command when you’re adding key bindings.

key_code_2

Returns a unique number for the second key in the specified key binding. Read-only.

protected

True if you cannot change the specified key binding in the Customize Keyboard dialog box (Tools menu, Customize command). Read-only.
Class: letter content

Plural
letter contents

Represents the elements of a letter created by the Letter Wizard.

To return a letter content object, use the letter content property of the document or the create letter content command. The following example retrieves and displays the letter recipient's name from the active document.

set myLetterContent to letter content of active document
display dialog recipient name of myLetterContent as string

The following example uses the create letter content command to create a new letter content object, which is then used with the run letter wizard command.

set LC to create letter content active document ¬
    date format ¬
    "July 31, 1996" include header footer false ¬
    page design ¬
    "" letter style full block ¬
    letterhead true ¬
    letterhead location letter top ¬
    letterhead size 108.0 ¬
    recipient name ¬
    "Tony Madigan" recipient address "432 SE Main St. " & return & ¬
    "Bellevue, WA 98004" salutation ¬
    "Dear Tony" salutation type salutation informal ¬
    recipient reference ¬
    "" mailing instructions ¬
    "" attention line ¬
    "" subject ¬
    "End of Year Report" cc list ¬
    "" return address ¬
    "" sender name ¬
    "" closing ¬
    "Sincerely yours," sender company ¬
    "" sender job title ¬
    "" sender initials ¬
    "" number of enclosures 0
run letter wizard active document letter content LC with wizard mode
Properties

attention line
Returns or sets the attention line text for a letter created by the Letter Wizard. Read/write.

ccc list
Returns or sets the carbon copy (CC) recipients for a letter created by the Letter Wizard. Read/write.

closing
Returns or sets the closing text for a letter created by the Letter Wizard (for example, "Sincerely yours"). Read/write.

date format
Returns or sets the date for a letter created by the Letter Wizard. Read/write.

enclosure count
Returns or sets the number of enclosures for a letter created by the Letter Wizard. Read/write.

include header footer
True if the header and footer from the page design template are included in a letter created by the Letter Wizard. Read/write.
Note Use the page design property to set the name of the template attached to a document created by the Letter Wizard.

letter style
Returns or sets the layout of a letter created by the Letter Wizard. Read/write.
Can be one of the following:

- full block
- modified block
- semi block

letterhead
True if space is reserved for a preprinted letterhead in a letter created by the Letter Wizard. Read/write.
Note The letterhead size property controls the size of the reserved letterhead space.
letterhead location

Returns or sets the location of the preprinted letterhead in a letter created by the Letter Wizard. Read/write.

Can be one of the following:

- letter bottom
- letter left
- letter right
- letter top

letterhead size

Returns or sets the amount of space (in points) to be reserved for a preprinted letterhead in a letter created by the Letter Wizard. Read/write.

mailing instructions

Returns or sets the mailing instruction text for a letter created by the Letter Wizard (for example, "Certified Mail"). Read/write.

page design

Returns or sets the name of the template attached to the document created by the Letter Wizard. Read/write.

recipient address

Returns or sets the address of the person who will be receiving the letter created by the Letter Wizard. Read/write.

recipient name

Returns or sets the name of the person who will be receiving the letter created by the Letter Wizard. Read/write.

recipient reference

Returns or sets the reference line (for example, "In reply to:") for a letter created by the Letter Wizard. Read/write.

return address

Returns or sets the return address for a letter created by the Letter Wizard. Read/write.

salutation

Returns or sets the salutation text for a letter created by the Letter Wizard. Read/write.
Microsoft Word Suite

**salutation type**

Returns or sets the salutation style for a letter created by the Letter Wizard. Read/write.

Can be one of the following:

- **salutation business**
- **salutation formal**
- **salutation informal**
- **salutation other**

**sender city**

Returns the city of the person creating the letter with the Letter Wizard. Read-only.

**sender company**

Returns or sets the company name of the person creating a letter with the Letter Wizard. Read/write.

**sender initials**

Returns or sets the initials of the person creating a letter with the Letter Wizard. Read/write.

**sender job title**

Returns or sets the job title of the person creating a letter with the Letter Wizard. Read/write.

**sender name**

Returns or sets the name of the person creating a letter with the Letter Wizard. Read/write.

**subject**

Returns or sets the subject text of a letter created by the Letter Wizard. Read/write.

**Class: line numbering**

**Plural**

**line numberings**

Represents line numbers in the left margin or to the left of each newspaper-style column.

To return the **line numbering** object, use the **line numbering** property of the **page setup** class. The following example applies line numbering to the text in the first section of the active document.

```plaintext
set LN to line numbering of page setup of section 1 of active document
set active line of LN to true
set count by of LN to 5
set restart mode of LN to restart page
```

The following example applies line numbering to the pages in the current section.

```plaintext
set active line of line numbering of page setup of selection to true
```
Properties

active line

True if line numbering is active for the specified document, section, or sections. Read/write.

count by

Returns or sets the numeric increment for line numbers. For example, if the count by property is set to 5, every fifth line will display the line number. Line numbers are displayed only in page layout view and print preview. Read/write.

This property has no effect unless the active line property of the line numbering object is set to true.

distance from text

Returns or sets the distance (in points) between the right edge of line numbers and the left edge of the document text. Read/write.

restart mode

Returns or sets the way line numbering runs, that is, whether it starts over at the beginning of a new page or section or runs continuously. Read/write.

Can be one of the following:

- restart continuous
- restart page
- restart section

You must be in page layout view to see line numbering.

starting number

Returns or sets the starting note number, line number, or page number. Read/write.

You must be in page layout view to see line numbering.

Class: link format

Plural

link formats

Represents the linking characteristics for a picture.

To return the link format object, use the link format property for a shape, inline shape, or field. The following example breaks the link for the first shape on the active document.

break link link format of shape 1 of active document

Remarks

You can use both the update command and the auto update property to update links. To return or set the full path for a particular link’s source file, use the source full name property.
Properties

auto update

True if the specified link is updated automatically when the container file is opened or when the source file is changed. Read/write.

link type

Returns the link type. Read-only.
Can be one of the following:

- link type import
- link type include
- link type ole
- link type picture
- link type reference
- link type dde
- link type ddeauto
- link type text

locked

True if a field, inline shape, or shape object is locked to prevent automatic updating. If you use this property with a shape object that’s a floating linked picture, an error occurs. Read/write.

save picture with document

True if the specified picture is saved with the document. Read/write.

Note This property works only with shapes and inline shapes that are linked pictures.

source full name

Returns or sets the name and path of the source file for the specified linked picture or field. Read/write.

source name

Returns the name of the source file for the specified linked picture or field. Read-only.

Note This property doesn't return the path for the source file.

source path

Returns the path of the source file for the specified linked picture or field. Read-only.

Note The path doesn't include a trailing character (for example, "Macintosh HD:Users:Shared").
Class: list entry

Plural
list entries

Represents an item in a drop-down form field. The list entries list includes all the items in a drop-down form field.

Use list entry index, where index is the list entry name or the index number, to return a single list entry object. The index number represents the position of the entry in the drop-down form field (the first item is index number 1). The following example deletes the "Blue" entry from the drop-down form field named "Color."

delete list entry "Blue" of drop down of form field "Color" of active document

The following example displays the first item in the drop-down form field named "Color."

display dialog name of list entry 1 of drop down of form field "Color" of active document as string

To add an item to a drop-down form field, use the make command. The following example inserts a drop-down form field and then adds "red," "blue," and "green" to the form field.

set myField to make new form field at active document with properties

{text object:text object of selection, form field type:field form drop down}

make new list entry at drop down of myField with properties {name:"Red"}
make new list entry at drop down of myField with properties {name:"Blue"}
make new list entry at drop down of myField with properties {name:"Green"}

Properties

entry index

Returns a number that indicates the position of the list entry in the list entries list. Read-only.

name

Returns or sets the name of the drop-down form field item. Read/write.

Class: list format

Plural
list formats

Represents the list formatting attributes that can be applied to the paragraphs in a range.

To return the list format object for a text range, use the list format property. The following example applies the default bulleted list format to the selection.

apply bullet default list format of text object of selection
An easy way to apply list formatting is to use the apply bullet default, and apply number default commands, which correspond, respectively, to the first list format (excluding None) on each tab in the Bullets and Numbering dialog box.

To apply a format other than the default format, use the apply list template command, which allows you to specify the list format (list template) you want to apply.

Returning the list or list template

To return the list or list template from the first paragraph in the specified range, use the Word list or list template property.

Remarks

To access the list formatting properties and commands available for the specified range, use the list format property with a text range object. The following example applies the default bullet list format to the second paragraph in the active document.

apply bullet default list format of text object of paragraph 2 of active document

However, if there's already a list defined in your document, you can access a specific Word list object by using the Word lists element of the document. The following example changes the format of the list created in the preceding example to the first number format on the Numbered tab in the Bullets and Numbering dialog box.

apply list template Word list 1 of active document list template list template 1 of list gallery 2

Properties

Word list

Returns a Word list object that represents the first formatted list contained in the specified list format object. Read-only.

If the first paragraph in the range for the list format object is not formatted as a list, the list property returns nothing.

list level number

Returns or sets the list level for the first paragraph in the specified list format object. Read/write.

list string

Returns a string that represents the appearance of the list value of the first paragraph in the range for the specified list format object. For example, the second paragraph in an alphabetical list would return B. Read-only.

For a bulleted list, you will need to apply the correct font in order to see the string. Most bullets use the Symbol or Wingdings font.

To return the numeric value of the paragraph, use the list value property.
Microsoft Word Suite

**list template**

Returns a list template object that represents the list formatting for the specified Word style or list format object. Read-only.

A list template includes all the formatting that defines a particular list. Each of the seven formats (excluding None) found on each of the tabs in the Bullets and Numbering dialog box corresponds to a list template. Documents and templates can also contain list templates.

If the first paragraph in the range for the list format object is not formatted as a list, the list template property returns nothing.

**list type**

Returns the type of lists that are contained in the range for the specified list format object. Read-only.

Can be one of the following:

- list bullet
- list listnum only
- list mixed numbering
- list no numbering
- list outline numbering
- list simple numbering

The constant list listnum only refers to LISTNUM fields, which are fields that can be added within the text of a paragraph.

**list value**

Returns the numeric value of the first paragraph in the range for the specified list format object. For example, the list value property applied to the second paragraph in an alphabetical list would return 2. Read-only.

To return a string that represents the appearance of the list value, use the list string property.

If the list format object is in a bulleted list, list value returns 1.

If the list format object is in an outline-numbered list, list value returns the numeric value of the first paragraph as it occurs in the sequence of paragraphs at the same level. For example, the list value property applied to a paragraph numbered "A.2" would return 2.

This property will not return the value for a LISTNUM field.

**single list**

True if the specified list format object contains only one list. Read-only.

**single list template**

True if the entire list format object uses the same list template. Read-only.
Class: list gallery

Plural
list galleries

Elements
list template

Represents a single gallery of list formats. Each list gallery object represents one of the three tabs in the Bullets and Numbering dialog box.

To return a single list gallery object, use the get list gallery command.

The following example returns the third list format (excluding None) on the Bulleted tab in the Bullets and Numbering dialog box and then applies it to the selection.

set temp3 to list template 3 of (get list gallery bullet gallery)
apply list format template (list format of text object of selection) ¬
    list template temp3

Class: list level

Plural
list levels

Represents a single list level, either the only level for a bulleted or numbered list, or one of the nine levels of an outline numbered list.

Use list level index, where index is a number from 1 through 9, to return a single list level object.

The following example sets list level one of list template one in the active document to start at 4.

set start at of list level 1 of list template 1 of active document to 4

Remarks
The list level object gives you access to all the formatting properties for the specified list level, such as the alignment, font object, number format, number position, number style, and trailing character properties.

To apply a list level, first identify the list or list format, and then use the apply list template or apply list format template command. Each tab at the beginning of the paragraph is translated into a list level. For example, a paragraph that begins with three tabs will become a level three list paragraph after the apply list template or apply list format template command is used.

Properties

entry index

Returns a number that indicates the position of the list level in the list levels list. Read-only.

font object

Returns or sets a font object that represents the character formatting of the specified list level.

To set this property, specify an expression that returns a font object. Read/write.

linked style

Returns or sets the name of the style that's linked to the specified list level object. Read/write.
list level alignment

Returns or sets the alignment for the list level for the list template. Read/write.

Can be one of the following:

- list level align center
- list level align left
- list level align right

number format

Returns or sets the number format for the specified list level. Read/write.

The percent sign (%) followed by any number from 1 through 9 represents the number style from the respective list level. For example, if you want the format for the first level to be "Article I," "Article II," and so on, the string for the number format property would be "Article %1" and the number style property would be set to list number style uppercase roman.

If the number style property is set to list number style bullet, the string for the number format property can contain only one character.

number position

Returns or sets the position (in points) of the number or bullet for the specified list level object. Read/write.

For each list level, you can set the position of the number or bullet, the position of the tab, and the position of the text that wraps.

number style

Returns or sets the number style for the specified object. Read/write.

Can be one of the following:

- list number style arabic
- list number style arabic lz
- list number style bullet
- list number style cardinal text
- list number style legal
- list number style legal lz
- list number style lowercase letter
- list number style lowercase roman
- list number style none
- list number style ordinal
- list number style ordinal text
- list number style uppercase letter
- list number style uppercase roman

reset on higher

True if the specified list level restarts numbering at 1 following a higher list level. False if the numbering continues sequentially each time the list level appears. Read/write.

This feature allows lists to be interleaved, maintaining numeric sequence.

start at

Returns or sets the starting number for the specified list level object. Read/write.
tab position
Returns or sets the tab position for the specified list level object. Read/write.

text position
Returns or sets the position for the second line of wrapping text for the specified list level object. Read/write.

trailing character
Returns or sets the character inserted after the number for the specified list level. Read/write.

Can be one of the following:

- trailing none
- trailing space
- trailing tab

Class: list template

Plural
list templates

Represents a single list template that includes all the formatting that defines a list. Each of the seven formats (excluding None) found on each of the three tabs in the Bullets and Numbering dialog box corresponds to a list template object. These predefined list templates can be accessed from the three list gallery objects. Documents and templates can also contain lists of list templates.

Use list template index, where index is a number from 1 through 7, to return a single list template from a list gallery. The following example returns the third list format (excluding None) on the Numbered tab in the Bullets and Numbering dialog box.

set temp3 to list template 3 of list gallery 2

Note Some properties and commands—convert, for example—won't work with list templates that are accessed from a list gallery. You can modify these list templates, but you cannot change their list gallery type (bullet gallery, number gallery, or outline number gallery).

The following example sets an object variable equal to the list template used in the third list in the active document, and then it applies that list template to the selection.

set myLT to list template 3 of active document
apply list format template (list format of text object of selection) ¬
    list template myLT

To reset formatting to the original list format, use the reset list gallery command for the list gallery object.

Remarks
After you have returned a list template object, use list level index, where index is a number from 1 through 9, to return a single list level object. With a list level object, you have access to all the formatting properties for the specified list level, such as alignment, font object, number format, number position, number style, and trailing character.
Properties

name
Returns or sets an optional list template name that can be used in conjunction with the Name instruction for a LISTNUM field. Read/write.

outline numbered
True if the specified list template object is outline numbered. Read/write.
Setting this property to false converts the list template to a single-level list, using the formatting of the first level.
You cannot set this property for a list template object returned from a list gallery object.

Class: mailing label

Elements

custom labels
Represents a mailing label.
To return the mailing label object, use the mailing label object property. The following example sets default mailing label options.

set default laser tray of mailing label object to printer lower bin
set default print bar code of mailing label object to true
To print a mailing label listed in the Product Number box in the Label Options dialog box, use the print out mailing label object command. The following example prints a page of Avery 5162 standard address labels using the specified address.

set addr to "Katie Jordan" & return & "123 Skye St." & return & ¬
    "OurTown, WA 98107"
print out mailing label name "5162" address addr

Remarks
To format or print a custom mailing label, use the custom labels class. The following example sets the number of labels across and down for the custom label named "MyLabel."

set CL to custom label "MyLabel" of mailing label object
set number across of CL to 2
set number down of CL to 5

Properties

default label name
Returns or sets the name for the default mailing label. Read/write.

Note To find the string for the specified built-in label, select the label in the Label Options dialog box (Tools menu, Labels command, Options button). Then click Details and look at the Label name box, which contains the correct string to use for this property. To set a custom label as the default mailing label, use the label name that appears in the Details dialog box, or use the Name property with a custom label object.
default laser tray

Returns or sets the default paper tray that contains sheets of mailing labels. Read/write.

Can be one of the following:

- printer automatic sheet feed
- printer default bin
- printer envelope feed
- printer form source
- printer large capacity bin
- printer large format bin
- printer lower bin
- printer manual envelope feed
- printer manual feed
- printer middle bin
- printer only bin
- printer paper cassette
- printer small format bin
- printer tractor feed
- printer upper bin

default print bar code

True if a POSTNET bar code is added to mailing labels by default. Read/write.

Class: page number

Plural page numbers

Represents a page number in a header or footer. The page numbers list includes all the page numbers in a single header or footer.

Use page number index, where index is the index number, to return a single page number object. In most cases, a header or footer will contain only one page number, which is index number 1. The following example centers the first page number in the primary header in section one in the active document.

set alignment of page number 1 of (get header section 1 of active document ¬ index header footer primary) to align page number center

To add a page number (a PAGE field) to a header or footer, use the make command. The following example adds a page number to the primary footer in the first section. The page number doesn't appear on the first page.

set myFooter to get footer section 1 of active document ¬ index header footer primary
make new page number at myFooter with properties ¬
{alignment:align page number left}
set show first page number of page number options of myFooter to false
Properties

alignment

Returns or sets the alignment for the page number. Read/write.

Can be one of the following:

- align page number center
- align page number inside
- align page number left
- align page number outside
- align page number right

entry index

Returns a number that indicates the position of the page number in the page numbers list. Read-only.

Class: page number options

Represents the options for page numbers in a single header or footer.

To return the page number options object, use the page number options property of the header footer class. The following example starts page numbering at 3 for the first section in the active document.

```
set starting number of page number options of (get footer section 1 of active document index header footer primary) to 3
```

Properties

chapter page separator

Returns or sets the separator character used between the chapter number and the page number. Read/write.

Can be one of the following:

- separator colon
- separator em dash
- separator en dash
- separator hyphen
- separator period

heading level for chapter

Returns or sets the heading level style that's applied to the chapter titles in the document. Can be a number from 0 (zero) through 8, corresponding to heading levels 1 through 9. Read/write.
**include chapter number**

*True* if a chapter number is included with page numbers or a caption label. Read/write.

**number style**

Returns or sets the number style for the specified object. Read/write.

Can be one of the following:

- caption number style arabic
- caption number style lowercase letter
- caption number style lowercase roman
- caption number style uppercase letter
- caption number style chosung
- caption number style hanja read
- caption number style hanja read digit
- caption number style kanji
- caption number style kanji digit
- caption number style kanji traditional
- caption number style number in circle
- caption number style simp chin num2
- caption number style simp chin num3
- caption number style zodiac1
- caption number style zodiac2
- caption number style uppercase roman
- caption number style ganada

**restart numbering at section**

*True* if page numbering starts at 1 again at the beginning of the specified section. Read/write.

If set to *false*, the *restart numbering at section* property overrides the *starting number* property so that page numbering can continue from the previous section.

**show first page number**

*True* if the page number appears on the first page in the section. Read/write.

Setting this property to *true* automatically adds page numbers to a section.

**starting number**

Returns or sets the starting page number. Read/write.

This property returns or sets the beginning page number for the specified *header footer* object. This number may or may not be visible on the first page, depending on the setting of the *show first page number* property. The *restart numbering at section* property, if set to *false*, overrides the *starting number* property so that page numbering can continue from the previous section.

**Class: page setup**

**Plural**

page setups

**Elements**

text column

Represents the page setup description. The *page setup* object contains all the page setup attributes of a document (left margin, bottom margin, paper size, and so on) as properties.
To return the page setup object, use the page setup property. The following example sets the first section in the active document to landscape orientation and then prints the document.

```plaintext
set orientation of page setup of section 1 of active document to orient landscape
print out active document
```

The following example sets all the margins for the document named "Sales.doc."

```plaintext
set pmSales to page setup of document "Sales.doc"
set left margin of pmSales to (inches to points inches 0.75)
set right margin of pmSales to (inches to points inches 0.75)
set top margin of pmSales to (inches to points inches 1.5)
set bottom margin of pmSales to (inches to points inches 1)
```

**Properties**

- **bottom margin**
  Returns or sets the distance (in points) between the bottom edge of the page and the bottom boundary of the body text. Read/write.

- **chars line**
  Returns or sets the number of characters per line in the document grid. Read/write.

- **different first page header footer**
  True if a different header or footer is used on the first page. Read/write.

- **first page tray**
  Returns or sets the paper tray to use for the first page of a document or section. Read/write.
  Can be one of the following:

  - printer automatic sheet feed
  - printer default bin
  - printer envelope feed
  - printer form source
  - printer large capacity bin
  - printer large format bin
  - printer lower bin
  - printer manual envelope feed
  - printer manual feed
  - printer middle bin
  - printer only bin
  - printer paper cassette
  - printer small format bin
  - printer tractor feed
  - printer upper bin

- **footer distance**
  Returns or sets the distance (in points) between the footer and the bottom of the page. Read/write.

- **gutter**
  Returns or sets the amount (in points) of extra margin space added to each page in a document or section for binding. Read/write.

  If the mirror margins property is set to true, the gutter property adds the extra space to the inside margins. Otherwise, the extra space is added to the left margin.
Microsoft Word Suite

gutter position

Returns or sets on which side the gutter appears in a document. Read/write.
Can be one of the following:

- gutter position left
- gutter position center
- gutter position right

header distance

Returns or sets the distance (in points) between the header and the top of the page. Read/write.

layout mode

Returns or sets the layout mode for the current document. Read/write.
Can be one of the following:

<table>
<thead>
<tr>
<th>Constant</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>layout mode</td>
<td></td>
</tr>
<tr>
<td>default</td>
<td>No grid is used to lay out text.</td>
</tr>
<tr>
<td>genko</td>
<td>Text is laid out on a grid; the user specifies the number of lines and the number of characters per line. As the user types, Word automatically aligns characters with gridlines.</td>
</tr>
<tr>
<td>grid</td>
<td>Text is laid out on a grid; the user specifies the number of lines and the number of characters per line. As the user types, Word doesn't automatically align characters with gridlines.</td>
</tr>
<tr>
<td>line grid</td>
<td>Text is laid out on a grid; the user specifies the number of lines, but not the number of characters per line.</td>
</tr>
</tbody>
</table>

left margin

Returns or sets the distance (in points) between the left edge of the page and the left boundary of the body text. Read/write.
If the mirror margins property is set to true, the left margin property controls the setting for inside margins and the right margin property controls the setting for outside margins.

line between text columns

True if vertical lines appear between all the columns in the text columns list. Read/write.

line numbering

Returns or sets the line numbering object that represents the line numbers for the specified page setup object. Read/write.
You must be in page layout view to see line numbering.
Microsoft Word Suite

*lines page*

Returns or sets the number of lines per page in the document grid. Read/write.

*mirror margins*

True if the inside margins of facing pages are the same width and the outside margins are the same width. Read/write.

If the *mirror margins* property is set to true, the *left margin* property controls the setting for inside margins and the *right margin* property controls the setting for outside margins.

*odd and even pages header footer*

True if the specified page setup object has different headers and footers for odd-numbered pages and even-numbered pages. Read/write.

*orientation*

Returns or sets the orientation of the page. Read/write.

Can be either of the following:

- orient landscape
- orient portrait

*other pages tray*

Returns or sets the paper tray to be used for all but the first page of a document or section. Read/write.

Can be one of the following:

- printer automatic sheet feed
- printer default bin
- printer envelope feed
- printer form source
- printer large capacity bin
- printer large format bin
- printer lower bin
- printer manual envelope feed
- printer manual feed
- printer middle bin
- printer only bin
- printer paper cassette
- printer small format bin
- printer tractor feed
- printer upper bin

*page height*

Returns or sets the height (in points) of the page. Read/write.

Setting the page height property changes the paper size property to paper custom.

Use the paper size property to set the page height and page width to those of a predefined paper size, such as Letter or A4.

*page width*

Returns or sets the width (in points) of the page. Read/write.

Setting the page width property changes the paper size property to paper custom.

Use the paper size property to set the page height and page width to those of a predefined paper size, such as Letter or A4.
**paper size**

Returns or sets the paper size. Read/write.

Can be one of the following:

- `paper ten X fourteen`
- `paper eleven X seventeen`
- `paper a3`
- `paper a4`
- `paper a4 small`
- `paper a5`
- `paper b4`
- `paper b5`
- `paper csheet`
- `paper custom`
- `paper dsheet`
- `paper envelope10`
- `paper envelope11`
- `paper envelope12`
- `paper envelope14`
- `paper envelope9`
- `paper envelope b4`
- `paper envelope b5`
- `paper envelope b6`
- `paper envelope c3`
- `paper envelope c4`
- `paper envelope e5`
- `paper envelope e6`
- `paper envelope ds5`
- `paper envelope dl`
- `paper envelope italy`
- `paper envelope monarch`
- `paper envelope personal`
- `paper envelope esheet`
- `paper executive`
- `paper fanfold legal german`
- `paper fanfold us`
- `paper folio`
- `paper ledger`
- `paper legal`
- `paper letter`
- `paper letter small`
- `paper note`
- `paper quarto`
- `paper statement`
- `paper tabloid`

**right margin**

Returns or sets the distance (in points) between the right edge of the page and the right boundary of the body text. Read/write.

If the `mirror margins` property is set to `true`, the `right margin` property controls the setting for outside margins and the `left margin` property controls the setting for inside margins.

**section start**

Returns or sets the type of section break for the specified `page setup` object. Read/write.

Can be one of the following:

- `section continuous`
- `section even page`
- `section new column`
- `section new page`
- `section odd page`

**show grid**

Determines whether to show the grid. Read/write.
spacing between text columns

Returns or sets the distance (in points) between text in adjacent columns of the specified row or rows. Read/write.

suppress endnotes

True if endnotes are printed at the end of the next section that doesn't suppress endnotes. Suppressed endnotes are printed before the endnotes in that section. Read/write.

Note This property takes effect only if the endnotes' location property is set to end of section.

text columns evenly spaced

True if text columns are evenly spaced. Read/write.

If you set the space after or width property of the text column object, the text columns evenly spaced property is automatically set to true. Also, setting the text columns evenly spaced property may change the settings for the space after and width properties of the text column object.

top margin

Returns or sets the distance (in points) between the top edge of the page and the top boundary of the body text. Read/write.

vertical alignment

Returns or sets the vertical alignment of text on each page in a document or section. Read/write.

Can be one of the following:

- align vertical bottom
- align vertical center
- align vertical justify
- align vertical top

width of text columns

Returns or sets the width (in points) of the text columns. Read/write.

Class: pane

Plural panes

Represents a window pane. The panes list includes all the window panes for a single window.

Use pane index, where index is the index number, to return a single pane object. The following example closes the active pane.

if (count of panes of active window) ≥ 2 then close active pane of ¬ active window
To add a window pane, use the `split window` property. The following example splits the active window at 20 percent of the current window size.

```plaintext
set split window of active window to true
set split vertical of active window to 20
```

**Remarks**

A window has more than one pane if the window is split or the view is not page layout view and information such as footnotes or comments are displayed. The following example displays the comments pane in normal view and then prompts to close the pane.

```plaintext
set view type of view of active window to normal view
if (count of Word comments of active document) > 1 then
   set split special of view of active window to pane comments
   set response to button returned of (display dialog "Do you want to close the comments pane?" buttons {"Yes", "No"})
   if response is "Yes" then close active pane of active window
end if
```

**Properties**

- **browse to window**
  - True if lines wrap at the right edge of the pane rather than at the right margin of the page. Read/write.

- **browse width**
  - Returns the width (in points) of the area in which text wraps in the specified pane. Read-only.

  **Note** This property works only in online layout view.

- **display rulers**
  - True if rulers are displayed for the specified window or pane. Equivalent to the Ruler command on the View menu. Read/write.

  **Note** If the display rulers property is false, the horizontal and vertical rulers won't be displayed, regardless of the state of the display vertical ruler property.

- **display vertical ruler**
  - True if a vertical ruler is displayed for the specified window or pane. Read/write.

  **Note** A vertical ruler appears only in page layout view, and only if the display rulers property is set to true.

- **document**
  - Returns a document object associated with the specified pane. Read-only.

- **entry index**
  - Returns a number that indicates the position of the pane in the panes list. Read-only.
horizontal percent scrolled
Returns or sets the horizontal scroll position as a percentage of the document width. Read/write.

minimum font size
Returns or sets the minimum font size (in points) displayed for the specified pane. Read/write.
Note This property affects only the text as shown in online layout view. The point sizes that are displayed on the Formatting command bar and used for printing aren’t changed.

next pane
Returns the next pane. Read-only.

previous pane
Returns the previous pane. Read-only.

selection
Returns the selection object that represents a selected range or the insertion point. Read-only.

vertical percent scrolled
Returns or sets the vertical scroll position as a percentage of the document length. Read/write.

view
Returns a view object that represents the view for the specified pane. Read-only.

Class: recent file

Plural
recent files
Represents a recently used file. The recent files list includes all the files that have been used recently. The items in the recent files list are displayed at the bottom of the File menu.

Use recent file index, where index is the index number, to return a single recent file object. The index number represents the position of the file on the File menu. The following example opens the first document in the recent files list.
if (count of recent files) ≠ 1 then open recent file recent file 1

Remarks
The save as and open commands include an add to recent files argument that controls whether or not a file is added to the recently-used-files list when the file is opened or saved.

Properties
entry index
Returns a number that indicates the position of the file in the recent files list. Read-only.

name
Returns the object’s file name. Read-only.
**path**

Returns the path to the specified recent file object. Read-only.

**Note** The path doesn't include a trailing character; for example, Macintosh HD:Users:Shared.

**read only**

True if changes to the document cannot be saved to the original document. Read/write.

**Class:** replacement

**Plural** replacements

Represents the replace criteria for a find-and-replace operation. The properties and commands of the replacement object correspond to the options in the Find and Replace dialog box.

To return a replacement object, use the replacement property. The following example replaces the next occurrence of the word "hi" with the word "hello."

```wont
set myFind to find object of selection
set content of myFind to "hi"
clear formatting myFind
set content of replacement of myFind to "hello"
clear formatting replacement of myFind
execute find myFind replace replace one with match forward
```

To find and replace formatting, set both the find text and the replace text to empty strings (""), and set the find format argument of the execute find command to true. The following example removes all the bold formatting in the active document. The bold property is true for the find object and false for the replacement object.

```wont
set myFind to find object of text object of active document
clear formatting myFind
set bold of font object of myFind to true
set content of myFind to ""
clear formatting replacement of myFind
set bold of font object of replacement of myFind to false
set content of replacement of myFind to ""
execute find myFind replace replace all
```

**Properties**

**content**

Returns or sets the text to replace in the specified text range or selection. Read/write.

**font object**

Returns or sets a font object that represents the character formatting of the replacement object. To set this property, specify an expression that returns a font object. Read/write.
frame

Returns a frame object associated with the replacement object. Read-only.

highlight

True if highlight formatting is applied to the replacement object. Read/write.

language id

Returns or sets the language for the specified object. Read/write.

Can be one of the following:

<table>
<thead>
<tr>
<th>Language</th>
<th>Language ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>afrikaans</td>
<td>language none</td>
</tr>
<tr>
<td>arabic</td>
<td>latvian</td>
</tr>
<tr>
<td>basque</td>
<td>macedonian</td>
</tr>
<tr>
<td>belgian dutch</td>
<td>malaysian</td>
</tr>
<tr>
<td>belgian french</td>
<td>mexicanSpanish</td>
</tr>
<tr>
<td>brazilian portuguese</td>
<td>language no proofing</td>
</tr>
<tr>
<td>bulgarian</td>
<td>norwegian bokmol</td>
</tr>
<tr>
<td>byelorussian</td>
<td>norwegian nynorsk</td>
</tr>
<tr>
<td>catalan</td>
<td>polish</td>
</tr>
<tr>
<td>croatian</td>
<td>portuguese</td>
</tr>
<tr>
<td>czech</td>
<td>romanian</td>
</tr>
<tr>
<td>danish</td>
<td>russian</td>
</tr>
<tr>
<td>dutch</td>
<td>serbian cyrillic</td>
</tr>
<tr>
<td>english aus</td>
<td>serbian latin</td>
</tr>
<tr>
<td>english canadian</td>
<td>sesotho</td>
</tr>
<tr>
<td>english new zealand</td>
<td>simplified chinese</td>
</tr>
<tr>
<td>english south africa</td>
<td>slovak</td>
</tr>
<tr>
<td>english uk</td>
<td>slovenian</td>
</tr>
<tr>
<td>english us</td>
<td>spanish</td>
</tr>
<tr>
<td>estonian</td>
<td>spanish modern sort</td>
</tr>
<tr>
<td>farsi</td>
<td>swedish</td>
</tr>
<tr>
<td>finnish</td>
<td>swiss french</td>
</tr>
<tr>
<td>french</td>
<td>swiss german</td>
</tr>
<tr>
<td>french canadian</td>
<td>swiss italian</td>
</tr>
<tr>
<td>german</td>
<td>traditional chinese</td>
</tr>
<tr>
<td>greek</td>
<td>tswana</td>
</tr>
<tr>
<td>hebrew</td>
<td>tswana</td>
</tr>
<tr>
<td>hungarian</td>
<td>turkish</td>
</tr>
<tr>
<td>italian</td>
<td>ukrainian</td>
</tr>
<tr>
<td>icelandic</td>
<td>venda</td>
</tr>
<tr>
<td>japanese</td>
<td>xhosa</td>
</tr>
<tr>
<td>korean</td>
<td>zulu</td>
</tr>
</tbody>
</table>

language id east asian

Returns or sets an East Asian language for the specified object. Can be any of the language id constants listed above. Read/write.

Note  This is the recommended way to apply an East Asian language to text in a document created in an East Asian version of Word.
Microsoft Word Suite

no proofing

True if Word finds or replaces text that the spelling and grammar checker ignores. Read/write.

paragraph format

Returns or sets a paragraph format object that represents the paragraph settings for the replacement object. Read/write.

style

Returns or sets the style for the specified replacement object. To set this property, specify either the local name of the style, an integer, a built-in Word style, or an object that represents the style. Read/write.

Class: revision

Plural

revisions

Represents a change marked with a revision mark. The revisions list includes all the revision marks in a range or document.

Use revision index, where index is the index number, to return a single revision object. The index number represents the position of the revision in the range or document. The following example displays the author name for the first revision in section one in the active document.

display dialog author of revision 1 of text object of section 1 of active document as string

Revision objects are added when change tracking is enabled. To track revisions made to the document text, set the track revisions property to true. The following example enables revision tracking and then inserts "Action " before the selection.

set track revisions of active document to true
insert text "Action " at text object of selection

Properties

author

Returns the name of the user who made the specified tracked change. Read-only.

date value

The date and time that the tracked change was made. Read-only.

entry index

Returns a number that indicates the position of an item in a list. Read-only.
revision type

Returns the revision type. Read-only.

Can be one of the following:

- no revision
- revision delete
- revision display field
- revision insert
- revision paragraph number
- revision property
- revision reconcile
- revision replace
- revision paragraph property
- revision table property
- revision section property
- revision style definition
- revision conflict
- revision style

text object

Returns a text range object that represents the portion of a document that's contained in the specified object. Read-only.

Class: selection object

Plural

selection objects

Elements

table word sentence character footnote endnote Word comment cell section paragraph
field form filed frame bookmark hyperlink object column row inline shape shape

Represents the selection in a window pane. The selection can either encompass an area in the document or be collapsed to an insertion point.

Note: There can be only one selection object per document window pane and only one selection object can be active.

Use the selection property to return the selection object. The following example collapses the selection (if necessary) and moves the insertion point to the end of the current line.

end key selection move unit a line extend by moving

The following example updates the results of the fields in the selection.

if (count of (get fields of selection)) > 1 ~
then update field fields of selection
To return the selection type (for example, a block selection or an insertion point), use the *selection type* property. The following example selects the current paragraph if the selection is an insertion point.

```vbnet
if selection type of selection is selection ip then
    select text object of paragraph 1 of selection
end if
```

Use the *get selection information* command to return information about the selection. If the selection is in a table, the following example displays the number of rows and columns in the table.

```vbnet
if (get selection information selection information type with in table) is "True" then
    display dialog "Columns = " & (get selection information selection information type maximum number of columns) & return & "Rows = " & (get selection information selection information type maximum number of rows)
end if
```

To select an item in a document, use the *select* command. The following example selects the first bookmark in the active document and formats it to appear in red.

```vbnet
if (count of bookmarks of active document) >= 1 then
    select bookmark 1 of active document
    set color index of font object of selection to red
end if
```

**Remarks**

To return a *text range* object from the *selection* object, use the *text object* property. The following example defines the variable myRange as the selected range.

```vbnet
set myRange to text object of selection
```

There can be only one *selection* object per window pane; however, you can have multiple *text range* objects defined in a single document. A *text range* object represents a document area that may or may not be selected. Working with *text range* objects, you can manipulate a document with minimal screen updates.

**Properties**

*IP at end of line*

- **True** if the insertion point is at the end of a line that wraps to the next line. **False** if the selection isn't collapsed, if the insertion point isn't at the end of a line, or if the insertion point is positioned before a paragraph mark. Read-only.

*bookmark id*

Returns the number of the bookmark that encloses the beginning of the specified selection or range; returns 0 (zero) if there's no corresponding bookmark. The number corresponds to the position of the bookmark in the document: 1 for the first bookmark, 2 for the second one, and so on. Read-only.
border options

Returns a border options object that represents options for the borders for the selection. Read/write.

column options

Returns a column options object that represents options for the table columns in the selection. Read-only.

column select mode

True if column selection mode is active. When this mode is active, the letters "COL" appear on the status bar. Read/write.

content

Returns the text contained in the selection. Read-only.

document

Returns a document object associated with the specified selection. Read-only.

endnote options

Returns an endnote options object that represents options for the endnotes in a selection. Read-only.

extend mode

True if extend selection mode is active. When this mode is active, the letters "EXT" appear on the status bar. Read/write.

find object

Returns a find object that contains the criteria for a find operation. Read-only.

The selection is changed if the find operation is successful.

fit text width

Returns or sets the width (in the current measurement units) in which Word fits the text in the current selection or range. Read/write.

font object

Returns a font object that represents the character formatting of the selection. Read/write.

footnote options

Returns a footnote options object that represents options for the footnotes in a selection. Read-only.

formatted text

Returns or sets a text range object that includes the formatted text in the specified selection. Read/write.

This property returns a text range object with the character formatting and text from the specified selection. Paragraph formatting is included in the text range object if there's a paragraph mark in the selection.

When you set this property, the text in the text range is replaced with formatted text. If you don't want to replace the existing text, use the collapse range command before using this property.
**header footer object**

Returns a **header footer** object for the specified selection. Read-only.

**is end of row mark**

True if the specified selection is collapsed and is located at the end-of-row mark in a table. Read-only.

**language id**

Returns or sets the language for the specified object. Read/write.

Can be one of the following:

- **afrikaans**
- **arabic**
- **basque**
- **belgian dutch**
- **belgian french**
- **brazilian portuguese**
- **bulgarian**
- **byelorussian**
- **catalan**
- **croatian**
- **czech**
- **danish**
- **dutch**
- **english aus**
- **english canadian**
- **english new zealand**
- **english south africa**
- **english uk**
- **english us**
- **estonian**
- **farsi**
- **finnish**
- **french**
- **french canadian**
- **german**
- **greek**
- **hebrew**
- **hungarian**
- **italian**
- **icelandic**
- **japanese**
- **korean**
- **language none**
- **latvian**
- **macedonian**
- **malaysian**
- **mexicanSpanish**
- **language no proofing**
- **norwegian bokmol**
- **norwegian nynorsk**
- **polish**
- **portuguese**
- **romanian**
- **russian**
- **serbian cyrillic**
- **serbian latin**
- **sesotho**
- **simplified chinese**
- **spanish**
- **spanish modern sort**
- **swedish**
- **swiss french**
- **swiss german**
- **swiss italian**
- **traditional chinese**
- **tsonga**
- **tswana**
- **turkish**
- **ukrainian**
- **venda**
- **xhosa**
- **zulu**

**language id east asian**

Returns or sets an East Asian language for the specified object. Can be any of the **language id** constants listed above. Read/write.

**Note** This is the recommended way to apply an East Asian language to text in a document created in an East Asian version of Word.
Microsoft Word Suite

no proofing

**True** if the spelling and grammar checker ignores the specified text. Returns **true** if the *no proofing* property is set to *true* for only some of the specified text. Read/write.

orientation

Returns or sets the orientation of the text in the range or selection when the Text Direction feature is enabled. Read/write.

Can be one of the following:

- **text orientation downward**
- **text orientation horizontal**
- **text orientation vertical east asian**
- **text orientation rotated east asian**
- **text orientation upward**

page setup

Returns a **page setup** object that's associated with the specified selection. Read-only.

paragraph format

Returns or sets a **paragraph format** object that represents the paragraph settings for the specified selection. Read/write.

previous bookmark id

Returns the number of the last bookmark that starts before or at the same place as the specified selection or range; returns 0 (zero) if there's no corresponding bookmark. Read-only.

row options

Returns a **row options** object that represents options for the table rows in selection. Read-only.

selection end

Returns or sets the ending character position of a selection. Read/write.

**Note**  If this property is set to a value smaller than the **selection start** property, the **selection start** property is set to the same value (that is, the **selection start** and **selection end** property are equal).

The **selection** object has a starting position and an ending position. The ending position is the point farthest away from the beginning of the story.

This property returns the ending character position relative to the beginning of the story. The main document story type (**main text story**) begins with character position 0 (zero). You can change the size of a selection by setting this property.
**selection flags**

Returns or sets properties of the selection. Read/write.

Can be set to one of the following:

- selection active
- selection at eol
- selection overtype
- selection replace
- selection start active end at eol
- selection start active and overtype
- selection start active and active
- selection start active and replace
- selection start active

The return value of the *selection flags* property is the sum of the constants that apply to the selection.

**Note**  The *selection at eol* constant is functionally equivalent to the *IP at end of line* property.

**selection is active**

*True* if the selection in the specified window or pane is active. Read-only.

**selection start**

Returns or sets the starting character position of a selection, range, or bookmark. Read/write.

**Note**  If this property is set to a value larger than that of the *selection end* property, the *selection end* property is set to the same value as that of *selection start* property.

*Selection* objects have starting and ending character positions. The starting position refers to the character position closest to the beginning of the story.

This property returns the starting character position relative to the beginning of the story. The main *document story type* (*main text story*) begins with character position 0 (zero). You can change the size of a selection by setting this property.

**selection type**

Returns the selection type. Read-only.

Can be one of the following:

- no selection
- selection block
- selection column
- selection frame
- selection inline shape
- selection ip
- selection normal
- selection row
- selection shape
shading

Returns a shading object that refers to the shading formatting for the selection. Read-only.

show Word comments by

Returns or sets the name of the reviewer whose comments are shown in the comments pane. You can choose to show comments either by a single reviewer or by all reviewers. To view the comments by all reviewers, set this property to “All Reviewers”. Read/write.

show hidden bookmarks

Returns or sets whether hidden bookmarks are included in the elements of the selection. Read/write.

start is active

True if the beginning of the selection is active. The active end of the selection moves when you press SHIFT+an arrow key. Read/write.

story length

Returns the number of characters in the story that contains the specified range or selection. Read-only.

story type

Returns the story type for the specified range, selection, or bookmark. Read-only.

Can be one of the following:

- comments story
- endnotes story
- even pages footer story
- even pages header story
- first page footer story
- first page header story
- footnotes story
- main text story
- primary footer story
- primary header story
- text frame story

style

Returns or sets the style for the specified object. To set this property, specify either the local name of the style, an integer, a built-in Word style, or an object that represents the style. Read/write.

supplemental language id

Returns or sets the language for the specified object. Can be any of the language id constants listed above. Read/write.

Note This is the recommended way to determine the language of a document created in an East Asian version of Word.

text object

Returns a text range object that represents the portion of a document that's contained in the selection. Read-only.
Class: subdocument

Plural
subdocuments

Represents a subdocument within a document or range. The subdocument list includes all the subdocuments in a range or document.

Use subdocument index, where index is the index number, to return a single subdocument object. The following example displays the path and file name of the first subdocument in the active document.

```plaintext
if has file of subdocument 1 of active document is true then
    display dialog (path of subdocument 1 of active document & ~
    path separator & name of subdocument 1 of active document) as string
end if
```

Properties

has file

True if the specified subdocument has been saved to a file. Read-only.

level

Returns the heading level used to create the subdocument. Read-only.

locked

True if a subdocument in a master document is locked. Read/write.

name

Returns the subdocument's file name. Read-only.

path

Returns the path to the specified subdocument object. Read-only.

Note: The path doesn't include a trailing character; for example, Macintosh HD:Users:Shared.

text object

Returns a text range object that represents the portion of a document that's contained in the specified object. Read-only.
Class: system object

Contains information about the computer system.

To return the system object, use the system_object property of the application class.

The following example displays the current screen resolution (for example, "1024 x 768").

set horz to horizontal resolution of system_object
set vert to vertical resolution of system_object
display dialog "Resolution = " & horz & " x " & vert

Properties

country

Returns the country/region designation of the system. Read-only.

cursor

Returns or sets the state (shape) of the pointer. Read/write.

Can be one of the following:

- cursor ibeam
- cursor normal
- cursor northwest arrow
- cursor wait

horizontal resolution

Returns the horizontal display resolution, in pixels. Read-only.

operating system

Returns the name of the current operating system. Read-only.

processor type

Returns the type of processor that the system is using. Read-only.

system version

Returns the version number of the processor. Read-only.

vertical resolution

Returns the vertical screen resolution, in pixels. Read-only.
Class: tab stop

Plural
tab stops

Represents a single tab stop. The tab stops list represents all the custom and default tab stops in a paragraph or group of paragraphs.

Use tab stop index, where index is the location of the tab stop (in points) or the index number, to return a single tab stop object. Tab stops are indexed numerically from left to right along the ruler. The following example removes the first custom tab stop from the first paragraph of the selection.
clear tab stop 1 of paragraph 1 of selection

The following example adds a right-aligned tab stop positioned at 2 inches to the first selected paragraphs.
make new tab stop at paragraph 1 of selection with properties ¬
   {tab stop position:(inches to points inches 2), ¬
    alignment:align tab right}

To add a tab stop, use the make command. The following example adds two tab stops to the first selected paragraph. The first tab stop is a left-aligned tab with a dotted tab leader positioned at 1 inch (72 points). The second tab stop is centered and is positioned at 2 inches.
make new tab stop at paragraph 1 of selection with properties ¬
   {tab stop position:(inches to points inches 1), ¬
    tab leader:tab leader dots, alignment:align tab left}
make new tab stop at paragraph 1 of selection with properties ¬
   {tab stop position:(inches to points inches 2), ¬
    alignment:align tab center}

Note  To adjust the spacing of default tab stops, set the default tab stop property of a document object.

Properties

alignment

Returns or sets the alignment for the specified tab stop. Read/write.

Can be one of the following:

- align tab bar
- align tab center
- align tab decimal
- align tab left
- align tab list
- align tab right
**custom tab**

*True* if the specified tab stop is a custom tab stop. Read-only.

**next tab stop**

Returns the next tab stop in the list. Read-only.

**previous tab stop**

Returns the previous tab stop in the list. Read-only.

**tab leader**

Returns or sets the character between entries and their page numbers in a table of contents. Read/write.

Can be one of the following:

- **tab leader dashes**
- **tab leader dots**
- **tab leader heavy**
- **tab leader lines**
- **tab leader middle dot**
- **tab leader spaces**

**tab stop position**

Returns or sets the position of a tab stop relative to the left margin. Read/write.

---

**Class: table of authorities**

**Plural**

**tables of authorities**

Represents a single table of authorities in a document (a TOA field). The **tables of authorities** list includes all the tables of authorities in a document.

Use **table of authorities index**, where *index* is the index number, to return a single **table of authorities** object. The index number represents the position of the table of authorities in the document. The following example includes category headers in the first table of authorities in the active document and then updates the table.

```plaintext
set myTOA to table of authorities 1 of active document
set include category header of myTOA to true
update myTOA
```

To add a table of authorities to a document, use the **make** command. The following example adds a table of authorities that includes all categories at the beginning of the active document.

```plaintext
set myRange to set range text object of active document start 0 end 0
make new table of authorities at active document with properties ¬
    {text object:myRange, passim:true, category:0, entry separator:":", "}
Properties

category
Returns or sets the category of entries to be included in a table of authorities. Corresponds to the \c switch for a TOA field. Values 1 through 16 correspond to the items in the Category list on the Table of Authorities tab in the Index and Tables dialog box. Read/write.

Note  The number 0 (zero), which corresponds to all categories, can only be used with this property when you are creating a new table of authorities. The following example inserts a table of authorities for all categories.

make new table of authorities at active document with properties ¬
{text object:text object of selection, category:0}

entry separator
Returns or sets the characters (up to five) that separate a table of authorities entry and its page number. The default is a tab character with a dotted leader. Corresponds to the \e switch for a TOA (Table of Authorities) field. Read/write.

include category header
True if the category name for a group of entries appears in the table of authorities (for example, Cases). Corresponds to the \h switch for a TOA (Table of Authorities) field. Read/write.

include sequence name
Returns or sets the SEQ (Sequence) field identifier for a table of authorities. Corresponds to the \s switch for a TOA field. Read/write.

keep entry formatting
True if formatting from table of authorities entries is applied to the entries in the specified table of authorities. Corresponds to the \f switch for a TOA (Table of Authorities) field. Read/write.

page number separator
Returns of sets the characters (up to five) that separate individual page references in a table of authorities. The default is a comma and a space. Corresponds to the \l switch for a TOA (Table of Authorities) field. Read/write.

passim
True if five or more page references to the same authority are replaced with "Passim." Corresponds to the \p switch for a TOA (Table of Authorities) field. Read/write.

separator
Returns or sets the characters (up to five) between the sequence number and the page number. A hyphen (-) is the default character. This property corresponds to the \d switch for a TOA field. Read/write.
tab leader

Returns or sets the character between entries and their page numbers in an index, table of authorities, table of contents, or table of figures. Read/write.

Can be one of the following:

- tab leader dashes
- tab leader dots
- tab leader heavy
- tab leader lines
- tab leader middle dot
- tab leader spaces

table of authorities bookmark

Returns or sets the name of the bookmark from which to collect table of authorities entries. Read/write.

The table of authorities bookmark property corresponds to the \b switch for a TOA (Table of Authorities) field.

text object

Returns a text range object that represents the portion of a document that's contained in the table of authorities object. Read-only.

Class: table of contents

Plural

tables of contents

Elements

heading style

Represents a single table of contents in a document. The tables of contents list includes all the tables of contents in a document.

Use table of contents index, where index is the index number, to return a single table of contents object. The index number represents the position of the table of contents in the document. The following example updates the page numbers of the items in the first table of figures in the active document.

update page numbers table of contents 1 of active document

To add a table of contents to a document, use the make command. The following example adds a table of contents at the beginning of the active document. The example builds the table of contents from all paragraphs styled as either Heading 1, Heading 2, or Heading 3.

set myRange to set range text object of active document start 0 end 0
make new table of contents at active document with properties ¬
{text object:myRange, use fields:false, use heading styles:true, ¬
lower heading level:3, upper heading level:1}
Properties

include page numbers

True if page numbers are included in the table of contents. Read/write.

lower heading level

Returns or sets the ending heading level for a table of contents. Corresponds to the ending value used with the \o switch for a TOC (Table of Contents) field. Read/write.

To set the starting heading level, use the upper heading level property. For example, to set the TOC field syntax \o "1-3", set the lower heading level property to 3 and the upper heading level property to 1.

right align page numbers

True if page numbers are aligned with the right margin in a table of contents. Read/write.

tab leader

Returns or sets the character between entries and their page numbers in a table of contents. Read/write.

Can be one of the following:

- tab leader dashes
- tab leader dots
- tab leader heavy
- tab leader lines
- tab leader middle dot
- tab leader spaces

table id

Returns or sets a one-letter identifier that's used to build a table of contents from TC (Table of Contents Entry) fields. Corresponds to the \f switch for a TOC field. For example, "T" builds a table of contents from TC fields using the table identifier T. Read/write.

text object

Returns a text range object that represents the portion of a document that's contained in the table of contents object. Read-only.

upper heading level

Returns or sets the starting heading level for a table of contents. Corresponds to the starting value used with the \o switch for a TOC (Table of Contents) field. Read/write.

To set the ending heading level, use the lower heading level property. For example, to set the TOC field syntax \o "1-3", set the lower heading level property to 3 and the upper heading level property to 1.
use fields

**True** if TC (Table of Contents Entry) fields are used to create a table of contents. Read/write.

use heading styles

**True** if built-in heading styles are used to create a table of contents. Read/write.

**Class: table of figures**

**Plural**

tables of figures

**Elements**

heading style

Represents a single table of figures in a document. The tables of figures list includes all the tables of figures in a document.

Use **table of figures** (index), where index is the index number, to return a single table of figures object. The index number represents the position of the table of figures in the document. The following example updates the page numbers of the items in the first table of figures in the active document.

```plaintext
update page numbers table of figures 1 of active document
```

To add a table of figures to a document, use the **make** command. A table of figures lists figure captions in the order in which they appear in the document. The following example replaces the selection in the active document with a table of figures that includes caption labels and page numbers.

```plaintext
make new table of figures at active document with properties ¬
    { text object: text object of selection, include label: true, ¬
      include page numbers: true }
```

**Properties**

caption

Returns or sets the label that identifies the items to be included in a table of figures. Corresponds to the \c switch for a TOC (Table of Contents) field. Read/write.

include label

**True** if the caption label and caption number are included in a table of figures. Read/write.

include page numbers

**True** if page numbers are included in the table of figures. Read/write.

lower heading level

Returns or sets the ending heading level for a table of figures. Corresponds to the ending value used with the \o switch for a TOC (Table of Contents) field. Read/write.

To set the starting heading level, use the upper heading level property. For example, to set the TOC field syntax (TOC \o "1-3"), set the lower heading level property to 3 and the upper heading level property to 1.
right align page numbers

True if page numbers are aligned with the right margin in a table of figures. Read/write.

tab leader

Returns or sets the character between entries and their page numbers in a table of figures. Read/write.

Can be one of the following:

- tab leader dashes
- tab leader dots
- tab leader heavy
- tab leader lines
- tab leader middle dot
- tab leader spaces

table id

Returns or sets a one-letter identifier that's used to build a table of figures from TC (Table of Contents Entry) fields. Corresponds to the \f switch for a TOC (Table of Contents) field. For example, "T" builds a table of contents from TC fields using the table identifier T. Read/write.

text object

Returns a text range object that represents the portion of a document that's contained in the specified object. Read-only.

upper heading level

Returns or sets the starting heading level for a table of figures. Corresponds to the starting value used with the \o switch for a TOC (Table of Contents) field. Read/write.

To set the ending heading level, use the lower heading level property. For example, to set the TOC field syntax {TOC \o "1-3"}, set the lower heading level property to 3 and the upper heading level property to 1.

use fields

True if TC (Table of Contents Entry) fields are used to create a table of figures. Read/write.

use heading styles

True if built-in heading styles are used to create a table of figures. Read/write.
Class: template

Plural
templates

Elements
auto text entry
document property
custom document property
list template

Represents a document template. The templates list includes all the available template objects.

Use template index, where index is the template name or the index number, to return a single template object. The index number represents the position of the template in the templates list. The following example opens the first template in the templates list.

open as document template 1

The make command isn’t available for the templates list. Instead, you can add a template to the templates list by doing any of the following:

- Using the open command with a document object to open a document based on a template or a template
- Using the attached template property with a document object to attach a template to a document

Remarks

To return a template object that refers to the Normal template, use the normal template property of the application class. To return the template attached to the specified document, use the attached template property of a document object.

To return or set the location of user or workgroup templates (that is, the folder where you want to store these templates), use the get default file path or set default file path command. The following example displays the user template folder from the File Locations pane of the Preferences dialog box (Word menu).

display dialog (get default file path file path type user templates path)

Properties

east asian line break

True if Word applies East Asian line-breaking rules to the specified paragraphs. Read/write.

full name

Returns the name of the specified document or template, including its path on disk. Read-only.

Using this property is equivalent to using the path, path separator, and name properties, in sequence.
**language id**

Returns or sets the language for the specified object. Read/write.

Can be one of the following:

<table>
<thead>
<tr>
<th>Language</th>
<th>Language</th>
</tr>
</thead>
<tbody>
<tr>
<td>afrikaans</td>
<td>language none</td>
</tr>
<tr>
<td>arabic</td>
<td>latvian</td>
</tr>
<tr>
<td>basque</td>
<td>macedonian</td>
</tr>
<tr>
<td>belgian dutch</td>
<td>malaysian</td>
</tr>
<tr>
<td>belgian french</td>
<td>language no proofing</td>
</tr>
<tr>
<td>brazilian portuguese</td>
<td>norwegian bokmol</td>
</tr>
<tr>
<td>bulgarian</td>
<td>nynorsk</td>
</tr>
<tr>
<td>byelorussian</td>
<td>polish</td>
</tr>
<tr>
<td>catalan</td>
<td>portuguese</td>
</tr>
<tr>
<td>croatian</td>
<td>romanian</td>
</tr>
<tr>
<td>czech</td>
<td>russian</td>
</tr>
<tr>
<td>danish</td>
<td>serbian cyrillic</td>
</tr>
<tr>
<td>dutch</td>
<td>serbian latin</td>
</tr>
<tr>
<td>english aus</td>
<td>sesotho</td>
</tr>
<tr>
<td>english canadian</td>
<td>simplified chinese</td>
</tr>
<tr>
<td>english new zealand</td>
<td>slovak</td>
</tr>
<tr>
<td>english south africa</td>
<td>slovenian</td>
</tr>
<tr>
<td>english uk</td>
<td>spanish modern sort</td>
</tr>
<tr>
<td>english us</td>
<td>swedish</td>
</tr>
<tr>
<td>estonian</td>
<td>swiss french</td>
</tr>
<tr>
<td>farsi</td>
<td>swiss german</td>
</tr>
<tr>
<td>finnish</td>
<td>swiss italian</td>
</tr>
<tr>
<td>french</td>
<td>traditional chinese</td>
</tr>
<tr>
<td>french canadian</td>
<td>tsonga</td>
</tr>
<tr>
<td>german</td>
<td>tswana</td>
</tr>
<tr>
<td>greek</td>
<td>turkish</td>
</tr>
<tr>
<td>hebrew</td>
<td>ukrainian</td>
</tr>
<tr>
<td>hungarian</td>
<td>venda</td>
</tr>
<tr>
<td>italian</td>
<td>xhosa</td>
</tr>
<tr>
<td>icelandic</td>
<td>zulu</td>
</tr>
</tbody>
</table>

**language id east asian**

Returns or sets an East Asian language for the specified object. Can be any of the `language id` constants listed above. Read/write.

**Note**  This is the recommended way to apply an East Asian language to text in a document created in an East Asian version of Word.

**name**

Returns the template's file name. Read-only.
Microsoft Word Suite

no proofing

True if the spelling and grammar checker ignores documents based on this template.
Read/write.

path

Returns the path to the specified template object. Read-only.

Note The path doesn't include a trailing character; for example, Macintosh HD:Users:Shared.
To add the character that separates folders and drive letters, use the path separator property of the application class. To return the file name without the path and use the full name property to return the file name and the path together, use the name property.

saved

True if the specified template hasn't changed since it was last saved. False if Word displays a prompt to save changes when the document is closed. Read/write.

template type

Returns the template type. Read-only.
Can be one of the following:

- attached template
- global template
- normal template

Class: text column

Plural

text columns

Represents a single text column. The text columns list includes all the columns in a document or section of a document.

Use text column index, where index is the index number, to return a single text column object. The index number represents the position of the column in the text columns list (counting from left to right).

The following example sets the space after the first text column in the active document to 0.5 inch.

set space after of text column 1 of page setup of active document to inches to points inches 0.5

To add a column to the list of columns, use the make command. By default, there's one text column in the text columns list. The following example adds a 2.5-inch-wide column to the active document.

make new text column at page setup of active document with properties ¬
{width:inches to points inches 2.5, space after:inches to points ¬
inches 0.5}
Properties

Space after

Returns or sets the amount of spacing (in points) after the specified paragraph or text column.
Read/write.

Width

Returns or sets the width (in points) of the specified object. Read/write.

Class: text input

Plural

text inputs

Represents a single text form field.

Use form field index, where index is either the bookmark name associated with the text form field or the index number, to return a form field object. To return a text input object, use the text input property with the form field object. The following example deletes the contents of the text form field named "Text1" in the active document.

clear text input of form field "Text1" of active document

The index number represents the position of the form field in the form fields list. The following example checks the type of the first form field in the active document. If the form field is a text form field, the example sets "Mission Critical" as the value of the field.

if type of form field 1 of active document is field form text input then
    set form field result of form field 1 of active document to "Mission Critical"
end if

The following example determines whether the ffield variable represents a valid text form field in the active document before it sets the default text.

set ffield to text input of form field 1 of active document

if valid of ffield is true then
    set default text input of ffield to "Type your name here"
else
    display dialog "The first field is not a text box"
end if

To add a text form field, use the make command with the form field object. The following example adds a text form field at the beginning of the active document and then sets the name of the form field to "FirstName."

set ffield to make new form field at active document with properties
    {text object: set range text object of active document start 0 end 0, form field type: field form text input}

set name of ffield to "FirstName"
Properties

default text input

Returns or sets the text that represents the default text box contents. Read/write.

format

Returns the text formatting for the specified text box. Read-only.
To set the text formatting, use the edit type command.

text input field type

Returns the type of text form field. Read-only.
Can be one of the following:

- calculation text
- current date text
- current time text
- date text
- number text
- regular text

valid

True if the specified form field object is a valid text form field. False if it isn't valid. Read-only.

width

Returns or sets the width of the text input object, in points. Read/write.

Class: text retrieval mode

Plural
text retrieval modes

Represents options that control how text is retrieved from a text range object.

To return a text retrieval mode object, use the text retrieval mode property of a text range. The following example displays the text of the first sentence in the active document, excluding field codes and hidden text.

set trml to text retrieval mode of sentence 1 of active document
set include hidden text of trml to false
set include field codes of trml to false
display dialog (get content of sentence 1 of active document)

Remarks

Changing the view type, include hidden text, or include field codes property of the text retrieval mode object doesn't change the screen display. Instead, changing one of these properties determines what text is retrieved from a text range object when the content property is used.
Properties

include field codes

True if the text retrieved from the specified range includes field codes. Read/write.

Note  The default value is the same as the setting of the Field codes option in the View pane of the Preferences dialog box (Word menu) until this property has been set. To retrieve text from the specified range, use the content property with a text range object.

include hidden text

True if the text retrieved from the specified range includes hidden text. Read/write.

Note  The default value is the same as the current setting of the Hidden text option in the View pane of the Preferences dialog box (Word menu) until this property has been set. To retrieve text from the specified range, use the content property with a text range object.

view type

Returns or sets the view type for the text retrieval mode object. Read/write.

Can be one of the following:

- master view
- normal view
- online view
- outline view
- page view
- WordNote view
- print preview view

Class: variable

Plural

variables

Represents a variable stored as part of a document. Document variables are used to preserve macro settings in between macro sessions. The variables list includes all the document variables in a document or template. You can use a variable object to transfer values between Visual Basic macros and AppleScript.

Use variable index, where index is the document variable name or the index number, to return a single variable object. The following example displays the value of the Temp document variable in the active document.

display dialog (get variable value of variable "temp" of active document)

The index number represents the position of the document variable in the variables list. The last variable added to the variables list is index number 1; the second-to-last variable added to the list is index number 2, and so on. The following example displays the name of the first document variable in the active document.

display dialog (get name of variable 1 of active document)
Microsoft Word Suite

To add a variable to a document, use the **make** command. The following example adds a document variable named "Temp" with a value of 12 to the active document.

```plaintext
make new variable at active document with properties ¬
   {name:"temp", variable value:"12"}
```

**Properties**

- **entry index**
  Returns a number that indicates the position of the variable in the variables list. Read-only.

- **name**
  Returns the document variable name. Read-only.

- **variable value**
  Returns or sets the value of the document variable. Read/write.

**Class:** view

**Plural**

views
Contains the view attributes (show all, field shading, table gridlines, and so on) for a window or pane.

To return the **view** object, use the **view** property of a **window** object. The following example sets view options for the active window.

```plaintext
set show all of view of active window to true
set table gridlines of view of active window to true
set wrap to window of view of active window to false
```

**Remarks**

To change the view, use the **view type** property. The following example switches the active window to normal view.

```plaintext
set view type of view of active window to normal view
```

To change the size of the text on-screen, use the **percentage** property of the **zoom** class. The following example enlarges the on-screen text to 120 percent.

```plaintext
set percentage of zoom of view of active window to 120
```

To view comments, endnotes, footnotes, or the document header or footer, use the **seek view** property. The following example displays the current footer in the active window in page layout view.

```plaintext
set view type of view of active window to page view
set seek view of view of active window to seek current page footer
```
Microsoft Word Suite

**Properties**

*browse to window*

- **True** if lines wrap at the right edge of the window rather than at the right margin of the page.
  - Read/write.
  - **Note** This property works only in online layout view.

*data merge data view*

- **True** if data merge data is displayed instead of data merge fields in the specified window.
  - Read/write.
  - **Note** If the specified window isn't a main document, an error occurs.

*draft*

- **True** if all the text in a window is displayed in the same sans-serif font with minimal formatting to speed up display.
  - Read/write.

*enlarge fonts less than*

- Returns or sets the point size below which screen fonts are automatically scaled to the larger size.
  - Read/write.
  - **Note** This property only affects the text as shown in online layout view. The point sizes that are displayed on the **Formatting** toolbar and used for printing aren't changed.

*field shading*

- Returns or sets on-screen shading for form fields.
  - Read/write.
  - Can be one of the following:
    - **field shading always**
    - **field shading never**
    - **field shading when selected**

*full screen*

- **True** if the window is in full-screen view.
  - Read/write.

*magnifier*

- **True** if the pointer is displayed as a magnifying glass in print preview, indicating that the user can click to zoom in on a particular area of the page or zoom out to see an entire page or spread of pages.
  - Read/write.
  - **Note** This property generates an error if the view is not print preview.
Microsoft Word Suite

seek view

Returns or sets the document element displayed in page layout view. Read/write.

Can be one of the following:

seek current page footer  seek first page header
seek current page header seek footnotes
seek endnotes seek main document
seek even pages footer seek primary footer
seek even pages header seek primary header
seek first page footer

Note  This property generates an error if the view is not page layout view.

show all

True if all nonprinting characters—such as hidden text, tab marks, space marks, and paragraph marks—are displayed. Read/write.

show animation

True if text animation is displayed. Read/write.

show bookmarks

True if square brackets are displayed at the beginning and end of each bookmark. Read/write.

show drawings

True if objects created with the drawing tools are displayed in page layout view. Read/write.

show field codes

True if field codes are displayed. Read/write.

show first line only

True if only the first line of body text is shown in outline view. Read/write.

Note  This property generates an error if the view isn't outline view or master document view.

show format

True if character formatting is visible in outline view. Read/write.

Note  This property generates an error if the view isn't outline view or master document view.

show hidden text

True if text formatted as hidden text is displayed. Read/write.

show highlight

True if highlight formatting is displayed and printed with a document. Read/write.

show hyphens

True if optional hyphens are displayed. An optional hyphen indicates where to break a word when it falls at the end of a line. Read/write.
show main text layer

True if the text in the specified document is visible when the header and footer areas are displayed. This property is equivalent to the Show/Hide Document Text button on the Header and Footer toolbar. Read/write.

show object anchors

True if object anchors are displayed next to items that can be positioned in page layout view. Read/write.

show optional breaks

True if Word displays optional line breaks. Read/write.

show paragraphs

True if paragraph marks are displayed. Read/write.

show picture place holders

True if blank boxes are displayed as placeholders for pictures. Read/write.

show spaces

True if space characters are displayed. Read/write.

show tabs

True if tab characters are displayed. Read/write.

show text boundaries

True if dotted lines are displayed around page margins, text columns, objects, and frames in page layout view. Read/write.

split special

Returns or sets the active window pane. Read/write.

Can be one of the following:

- pane comments
- pane current page footer
- pane current page header
- pane endnote continuation notice
- pane endnote continuation separator
- pane endnotes
- pane endnote separator
- pane even pages footer
- pane even pages header
- pane first page footer
- pane first page header
- pane footnote continuation notice
- pane footnote continuation separator
- pane footnotes
- pane footnote separator
- pane none
- pane primary footer
- pane primary header

table gridlines

True if table gridlines are displayed. Read/write.
**Microsoft Word Suite**

*view type*

Returns or sets the view type. Read/write.

Can be one of the following:

- master view
- normal view
- online view
- outline view
- page view
- print preview view

*wrap to window*

True if lines wrap at the right edge of the document window rather than at the right margin or the right column boundary. Read/write.

**Note** This property has no effect in page layout or online layout view.

*zoom*

Returns a zoom object that represents the magnification for the specified view. Read-only.

**Class: web options**

Contains document-level attributes used by Word when you save a document as a Web page or open a Web page. You can return or set attributes either at the application (global) level or at the document level. (Note that attribute values can be different from one document to another, depending on the attribute value at the time the document was saved.) Document-level attribute settings override application-level attribute settings. Application-level attributes are contained in the default web options object.

Use the web options property to return the web options object. The following example checks to see whether PNG (Portable Network Graphics) is allowed as an image format and then sets the strImageFileType variable accordingly.

```vba
set objDocWebOptions to web options of active document
if allow png of objDocWebOptions is true then
    set strImageFileType to "PNG"
else
    set strImageFileType to "JPG"
end if
```
Properties

allow png

True if PNG (Portable Network Graphics) is allowed as an image format when you save a document as a Web page. False if PNG is not allowed as an output format. The default value is false. Read/write.

If you save images in the PNG format and if the Web browsers you are targeting support the PNG format, you might improve the image quality or reduce the size of those image files, and therefore decrease the download time.

doc keywords

Determines the keywords associated with a document. Read/write.

doc title

Determines the title for a Web document. Read-only.

encoding

Returns or sets the document encoding (code page or character set) to be used by the Web browser when you view the saved document. The default is the system code page. Read/write. Can be any of the following:

encoding Thai
encoding simplified Chinese
encoding traditional Chinese
encoding big endian
encoding Cyrillic
encoding Greek
encoding Hebrew
encoding Baltic
encoding simplified Chinese auto detect
encoding traditional Chinese auto detect
encoding Greek auto detect
encoding ISO88591 Latin1
encoding ISO88593 Latin3
encoding ISO88595 Cyrillic
encoding ISO88597 Greek
encoding ISO88599 Turkish
encoding ISO2022 Japanese
no half width Katakana
encoding ISO2022 Japanese JISX02011989
encoding ISO2022CN traditional Chinese
encoding Mac Roman
encoding Mac traditional Chinese Big5
encoding Mac Greek1
encoding Mac simplified Chinese GB2312
encoding Mac Ukraine
encoding Mac Icelandic
encoding Mac Croatia
encoding EBCDIC International
encoding EBCDIC Greek modern
encoding EBCDIC Germany
encoding EBCDIC Finland Sweden
encoding Japanese ShiftJIS
encoding Korean
encoding little endian
encoding central European
encoding Cyrillic auto detect
encoding Arabic auto detect
encoding ISO88592 central Europe
encoding ISO88594 Baltic
encoding ISO88596 Arabic
encoding ISO88598 Hebrew
encoding ISO885915 Latin9
encoding ISO2022 Japanese JISX02021984
encoding ISO2022KR
encoding ISO2022CN simplified Chinese
encoding Mac Japanese
encoding Mac Roman
encoding Mac Cyrillic
encoding Mac Latin2
encoding Mac Turkish
encoding EBCDIC US Canada
encoding EBCDIC multilingual
encoding ROECE Latin2
encoding EBCDIC Turkish Latin5
encoding EBCDIC Denmark Norway
Returns or sets the density (pixels per inch) of graphics images and table cells on a Web page. The range of settings is usually from 19 to 480, and common settings for popular screen sizes are 72, 96, and 120. The default setting is 96. Read/write.

This property determines the size of the images and cells on the specified Web page relative to the size of text whenever you view the saved document in a Web browser. The physical dimensions of the resulting image or cell are the result of the original dimensions (in inches) multiplied by the number of pixels per inch.

You use the screen size property to set the optimum screen size for the targeted Web browsers.
round trip html

Determines whether to save an HTML document with information that is specific to Word. Setting this property to true allows you to preserve all Word settings in an HTML document. Read/write.

screen size

Returns or sets the ideal minimum screen size (width by height, in pixels) that you should use when viewing the saved document in a Web browser. Can be one of the values listed below. The default constant is resolution 800x600. Read/write.

<table>
<thead>
<tr>
<th>resolution</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>544x376</td>
<td>1152x900</td>
</tr>
<tr>
<td>640x480</td>
<td>1280x1024</td>
</tr>
<tr>
<td>720x512</td>
<td>1600x1200</td>
</tr>
<tr>
<td>800x600</td>
<td>1800x1440</td>
</tr>
<tr>
<td>1024x768</td>
<td>1920x1200</td>
</tr>
<tr>
<td>1152x882</td>
<td></td>
</tr>
</tbody>
</table>

use long file names

True if long file names are used when you save the document as a Web page. False if long file names are not used and the DOS file name format (8.3) is used. The default value is true. Read/write.

Class: window

Plural
windows

Elements
pane

Represents a window. Many document characteristics, such as scroll bars and rulers, are actually properties of the window. The windows list for the application object contains all the windows in the application, whereas the windows list for the document object contains only the windows that display the specified document.

Use window index, where index is the window name or the index number, to return a single window object. The following example maximizes the Document1 window. The index number is the number to the left of the window name on the Window menu. The following example displays the caption of the first window in the windows list.

display dialog (get caption of window 1)

To add a new window to the windows list, use the make command. The following statement creates a new window for the document in the active window.

make new window

Remarks

A colon (:) and a number appear in the window caption when more than one window is open for a document.

When you switch the view to print preview, a new window is created. This window is removed from the windows list when you close print preview.
**Properties**

**IME mode**

Returns or sets the default start-up mode for the Japanese Input Method Editor (IME). Read/write. Can be one of the following:

<table>
<thead>
<tr>
<th>Constant</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IME mode alpha</td>
<td>Activates the IME in half-width Latin mode.</td>
</tr>
<tr>
<td>IME mode alpha full</td>
<td>Activates the IME in full-width Latin mode.</td>
</tr>
<tr>
<td>IME mode hangul</td>
<td>Activates the IME in half-width hangul mode.</td>
</tr>
<tr>
<td>IME mode hangul full</td>
<td>Activates the IME in full-width hangul mode.</td>
</tr>
<tr>
<td>IME mode hiragana</td>
<td>Activates the IME in full-width hiragana mode.</td>
</tr>
<tr>
<td>IME mode katakana</td>
<td>Activates the IME in full-width katakana mode.</td>
</tr>
<tr>
<td>IME mode katakana half</td>
<td>Activates the IME in half-width katakana mode.</td>
</tr>
<tr>
<td>IME mode no control</td>
<td>Does not change the IME mode.</td>
</tr>
<tr>
<td>IME mode off</td>
<td>Disables the IME and activates Latin text entry.</td>
</tr>
<tr>
<td>IME mode on</td>
<td>Activates the IME.</td>
</tr>
</tbody>
</table>

**active**

True if the specified window is active. Read-only.

**active pane**

Returns a pane object that represents the active pane for the specified window. Read-only.

**caption**

Returns or sets the caption text for the specified document window. Read/write.

**display horizontal scroll bar**

True if a horizontal scroll bar is displayed for the specified window. Read/write.

**display rulers**

True if rulers are displayed for the specified window or pane. Equivalent to the Ruler command on the View menu. Read/write.

**Note** If the display rulers property is false, the horizontal and vertical rulers won't be displayed, regardless of the state of the display vertical ruler property.

**display screen tips**

True if comments, footnotes, endnotes, and hyperlinks are displayed as tips. Text marked as having comments is highlighted. Read/write.

**display vertical ruler**

True if a vertical ruler is displayed for the specified window or pane. Read/write.

**Note** A vertical ruler appears only in page layout view, and only if the display rulers property is set to true.
Microsoft Word Suite

display vertical scroll bar
   True if a vertical scroll bar is displayed for the specified window. Read/write.

document
   Returns a document object associated with the specified window. Read-only.

document map
   True if the document map is visible. Read/write.

document map percent width
   Returns or sets the width of the document map as a percentage of the width of the specified window. Read/write.

entry index
   Returns a number that indicates the position of the window in the windows list. Read-only.

height
   Returns or sets the height of the window. You cannot set this property if the window is maximized or minimized. To determine the maximum size for the window, use the usable height property of the application class. To determine the window state, use the window state property. Read/write.

horizontal percent scrolled
   Returns or sets the horizontal scroll position as a percentage of the document width. Read/write.

left position
   Returns or sets the horizontal position (in points) of the specified window. Read/write.

next window
   Returns the next window in the windows list. Read-only.

previous window
   Returns the previous window in the windows list. Read-only.

selection
   Returns the selection object that represents a selected range or the insertion point. Read-only.

split vertical
   Returns or sets the vertical split percentage for the specified window. Read/write.

   Note To remove the split, set this property to zero (0) or set the split window property to false.

split window
   True if the window is split into multiple panes. Read/write.

style area width
   Returns or sets the width (in points) of the style area. Read/write.

   Note When the style area width property is greater than 0 (zero), style names are displayed to the left of the text. The style area isn't visible in page layout view or online layout view.
Microsoft Word Suite

**top**
Returns or sets the vertical position (in points) of the specified window or task. Read/write.

**vertical percent scrolled**
Returns or sets the vertical scroll position as a percentage of the document length. Read/write.

**view**
Returns a view object that represents the view for the specified window. Read-only.

**width**
Returns or sets the width (in points) of the specified object. Read/write.

**window number**
Returns the window number of the document displayed in the specified window. For example, if the caption of the window is "Sales.doc:2," this property returns the number 2. Read-only.

**Note** Use the entry index property to return the number of the specified window in the windows list.

**window state**
Returns or sets the state of the specified document window or task window. Read/write.
Can be one the following:

- window state maximize
- window state minimize
- window state normal

The window state normal constant indicates a window that's not maximized or minimized. The state of an inactive window cannot be set. To activate a window prior to setting the window state, use the activate object command.

**window type**
Returns the window type. Read-only.
Can be either of the following:

- window document
- window template

**Class:** Word comment

**Plural**

**Word comments**
Represents a single comment. The Word comments list includes comments in a selection, range, or document.
Use **Word comment** index, where *index* is the index number, to return a single **Word comment** object. The index number represents the position of the comment in the specified selection, range, or document. The following example displays the author of the first comment in the active document.

display dialog (get author of Word comment 1 of active document)

To add a comment at the specified range, use the **make** command. The following example adds a comment immediately after the selection.

collapse range text object of selection direction collapse end

make new Word comment at selection with properties ¬

   {comment text:"review this"}

To return the reference mark associated with the specified comment, use the **note reference** property. To return the text associated with the specified comment, use the **comment text** property. The following example displays the text associated with the first comment in the active document.

display dialog (get content of comment text of Word comment 1 ¬

   of active document)

**Properties**

*author*

   Returns or sets the author name for a comment. Read/write.

*comment index*

   Returns a number that indicates the position of the Word comment in the Word comments list. Read-only.

*comment text*

   Returns a text range object that represents the portion of a document that's contained in the specified object. Read-only.

*initials*

   Returns or sets the initials of the user associated with a specific comment. Read/write.

*note reference*

   Returns a text range object that represents a comment reference mark. Read-only.

*scope*

   Returns a text range object that represents the range of text marked by the specified comment. Read-only.

*show tip*

   True if text associated with a comment is displayed in a screen tip. The screen tip remains displayed until you click the mouse or press a key. Read/write.
Class: Word list

Plural
Word lists

Elements
paragraph

Represents a single list format that's been applied to specified paragraphs in a document.
Use **Word list index**, where *index* is the index number, to return a single **Word list** object. The following example returns the number of items in list one in the active document.

```vba
set myCount to count numbered items Word list 1 of active document
```

Remarks

To apply a different list format to an existing list, use the **apply list template** command with the **Word list** object. To add a new list to a document, use the **apply list format template** command with the **list format** object for a specified range.

To determine whether you can continue the list formatting from a list that was previously applied to the document, use the **can continue previous list** command.

To return the number of items in a numbered or bulleted list, including LISTNUM fields, use the **count numbered items** command.

To determine whether a list contains more than one list template, use the **single list template** property.

You can manipulate the individual **Word list** objects within a document, but for more precise control you should work with the **list format** object.

Properties

**single list template**

*True* if the entire **Word list** object uses the same list template. Read-only.

**text object**

Returns a **text range** object that represents the portion of a document that's contained in the specified object. Read-only.

Class: Word options

Represents application and document options in Word. Many of the properties for the **Word options** object correspond to items in the **Preferences** dialog box (**Word** menu). Read/write.

Use the **settings** property to return the **Word options** object. The following example sets three application options for Word.

```vba
set allow drag and drop of settings to true
set confirm conversions of settings to false
set measurement unit of settings to points
```
Microsoft Word Suite

Properties

IME automatic control

Returns or sets whether Word is set to automatically open and close the Japanese Input Method Editor. Read/write.

RTF in clipboard

Returns or sets whether all text copied from Word to the Clipboard retains its character and paragraph formatting. Read/write.

allow accented uppercase

Returns or sets whether accents are retained when a French language character is changed to uppercase. This property affects only text that's been marked as standard French. For all other languages, accents are always retained even if the allow accented uppercase property has been set to false. If you change a character back to lowercase after an accent has been stripped from it, the accent won't reappear. Read/write.

allow click and type mouse

Returns or sets whether click and type functionality is enabled. Read/write.

allow drag and drop

Returns or sets whether dragging and dropping can be used to move or copy a selection. Read/write.

allow fast save

Returns or sets whether Word saves only changes to a document. When reopening the document, Word uses the saved changes to reconstruct the document. The allow fast save and create backup properties cannot be set to true concurrently. Read/write.

animate screen movements

Returns or sets whether Word animates mouse movements, uses animated cursors, and animates actions such as background saving and find-and-replace operations. Read/write.

apply east asian fonts to ascii

Returns or sets whether Word applies East Asian fonts to Latin text. This property only applies when you have selected an East Asian language for editing. If this property is false and you apply an East Asian font to a specified range, Word will not apply the font to any Latin text in the range. Read/write.

auto format apply bulleted lists

Returns or sets whether characters at the beginning of list paragraphs — such as asterisks, hyphens, and greater-than signs — are replaced with bullets from the Bullets and Numbering dialog box when Word formats a document or range automatically. Read/write.

auto format apply first indents

Returns or sets whether Word automatically replaces a space entered at the beginning of a paragraph with a first-line indent when Word formats a document or range automatically. Read/write.
Microsoft Word Suite

auto format apply headings
Returns or sets whether styles are applied to headings automatically when Word formats a document or range automatically. Read/write.

auto format apply lists
Returns or sets whether styles are applied to lists automatically when Word formats a document or range automatically. Read/write.

auto format apply other paragraphs
Returns or sets whether styles are applied to paragraphs automatically that aren't headings or list items when Word formats a document or range automatically. Read/write.

auto format as you type apply borders
Returns or sets whether a series of three or more hyphens (-), equal signs (=), or underscore characters (_) are automatically replaced by a specific border line when the ENTER key is pressed. Read/write.

auto format as you type apply bulleted lists
Returns or sets whether bullet characters — such as asterisks, hyphens, and greater-than signs — are replaced with bullets from the bullets and numbering dialog box as you type. Read/write.

auto format as you type apply closings
Returns or sets whether Word automatically applies the closing style to letter closings as you type. Read/write.

auto format as you type apply dates
Returns or sets whether Word automatically applies the date style to dates as you type. Read/write.

auto format as you type apply first indents
Returns or sets whether Word automatically replaces a space entered at the beginning of a paragraph with a first-line indent. Read/write.

auto format as you type apply headings
Returns or sets whether styles are automatically applied to headings as you type. Read/write.

auto format as you type apply numbered lists
Returns or sets whether paragraphs are automatically formatted as numbered lists with a numbering scheme from the Bullets and Numbering dialog box according to what's typed. Read/write.

auto format as you type apply tables
Returns or set whether Word automatically creates a table when you type a plus sign, a series of hyphens, another plus sign, and so on, and then press ENTER. The plus signs become the column borders, and the hyphens become the column widths. Read/write.
Microsoft Word Suite

**auto format as you type auto letter wizard**
Returns or sets whether Word automatically starts the Letter Wizard when the user enters a letter salutation or closing. Read/write.

**auto format as you type define styles**
Returns or sets whether Word automatically creates new styles based on manual formatting. Read/write.

**auto format as you type delete auto spaces**
Returns or sets whether Word automatically deletes spaces inserted between Japanese and Latin text as you type. Read/write.

**auto format as you type format list item beginning**
Returns or sets whether Word repeats character formatting applied to the beginning of a list item to the next list item. Read/write.

**auto format as you type insert closings**
Returns or sets whether Word automatically inserts the corresponding memo closing when the user enters a memo heading. Read/write.

**auto format as you type insert overs**
Returns or sets whether Microsoft Word is set to automatically insert “以上” when the user enters “または” or “もしくは”. Read/write.

**auto format as you type match parentheses**
Returns or sets whether Word automatically corrects improperly paired parentheses. Read/write.

**auto format as you type replace east asian dashes**
Returns or sets whether Microsoft Word automatically corrects long vowel sounds and dashes. Read/write.

**auto format as you type replace fractions**
Returns or sets whether typed fractions are replaced with fractions from the current character set as you type. Read/write.

**auto format as you type replace hyperlinks**
Returns or sets whether e-mail addresses, server and share names (also known as UNC paths), and Internet addresses (also known as URLs) are automatically changed to hyperlinks as you type. Word changes any text that looks like an e-mail address, UNC, or URL to a hyperlink. Word doesn't check the validity of the hyperlink. Read/write.

**auto format as you type replace ordinals**
Returns or sets whether the ordinal number suffixes st, nd, rd, and th are replaced with the same letters in superscript as you type. For example, 1st is replaced with 1 followed by st formatted as superscript. Read/write.
Microsoft Word Suite

auto format as you type replace plain text emphasis
Returns or sets whether manual emphasis characters are automatically replaced with character formatting as you type. For example, "**bold**" is changed to "**bold**" and "_italic_" is changed to "italic". Read/write.

auto format as you type replace quotes
Returns or sets whether straight quotation marks are automatically changed to smart (curly) quotation marks as you type. Read/write.

auto format as you type replace symbols
Returns or sets whether two consecutive hyphens (--) are replaced with an en dash (–) or an em dash (—) as you type. Read/write.

auto format delete auto spaces
Returns or sets whether Word automatically deletes spaces inserted between Japanese and Latin text when Word formats a document or range automatically. Read/write.

auto format match parentheses
Returns or sets whether Word automatically corrects improperly paired parentheses when Word formats a document or range automatically. Read/write.

auto format preserve styles
Returns or sets whether previously applied styles are preserved when Word formats a document or range automatically. Read/write.

auto format replace east asian dashes
Returns or sets whether Word automatically corrects long vowel sounds and dashes when Word formats a document or range automatically. Read/write.

auto format replace fractions
Returns or sets whether typed fractions are replaced with fractions from the current character set when Word formats a document or range automatically. Read/write.

auto format replace hyperlinks
Returns or sets whether e-mail addresses, server and share names (also known as UNC paths), and Internet addresses (also known as URLs) are automatically changed to hyperlinks when Word formats a document or range automatically. Word changes any text that looks like an e-mail address, UNC, or URL to a hyperlink. Word doesn't check the validity of the hyperlink. Read/write.

auto format replace ordinals
Returns or sets whether the ordinal number suffixes st, nd, rd, and th are replaced with the same letters in superscript when Word formats a document or range automatically. For example, 1st is replaced with 1 followed by st formatted as superscript. Read/write.

auto format replace plain text emphasis
Returns or sets whether manual emphasis characters are replaced with character formatting when Word formats a document or range automatically. Read/write.
auto format replace quotes

Returns or sets whether straight quotation marks are automatically changed to smart, curly, quotation marks when Word formats a document or range automatically. Read/write.

auto format replace symbols

Returns or set whether two consecutive hyphens (--) are replaced by an en dash (–) or an em dash (—) when Word formats a document or range automatically. Read/write.

auto word selection

Returns or sets whether dragging selects one word at a time instead of one character at a time. Read/write.

ay match fuzzy

Returns or sets whether Microsoft Word ignores the distinction between “ż” and “ź” following ż-row and ź-row characters during a search. Read/write.

blue screen

Returns or sets whether Word displays text as white characters on a blue background. Read/write.

button field clicks integer

Returns or sets the number of clicks, either one or two, required to run a GOTOBUTTON or MACROBUTTON field. Read/write.

bv match fuzzy

Returns whether Microsoft Word ignores the distinction between “ź” and “ź” and between “ż” and “ż” during a search. Read/write.

byte match fuzzy

Returns or sets whether Word ignores the distinction between full-width and half-width characters (Latin or Japanese) during a search. Read/write.

case match fuzzy

Returns or sets whether Word ignores the distinction between uppercase and lowercase letters during a search. Read/write.

check grammar as you type

Returns or sets whether Word checks grammar and marks errors automatically as you type. This property marks grammatical errors, but to see them on screen, you must set the show grammatical errors property of the document object to true. Read/write.

check grammar with spelling

Returns or sets whether Word checks grammar while checking spelling. This property controls whether Word checks grammar when you check spelling by using the Spelling command (Tools menu). To check spelling from a script, use the check spelling command and use the check grammar command to check both grammar and spelling. Read/write.
Microsoft Word Suite

check spelling as you type

Returns or sets whether Word checks spelling and marks errors automatically as you type. This property marks spelling errors, but to see them on screen you must set the show spelling errors property of the document object to true. Read/write.

confirm conversions

Returns or sets whether Word displays the Convert File dialog box before it opens or inserts a file that isn't a Word document or template. In the Convert File dialog box, the user chooses the format to convert the file from. Read/write.

convert high ansi to east asian

Returns or sets whether Word converts text that is associated with an East Asian font to the appropriate font when it opens a document. Read/write.

create backup

Returns or sets whether Word creates a backup copy each time a document is saved. The create backup and allow fast save properties cannot be set to true concurrently. Read/write.

dash match fuzzy

Returns or sets whether Word ignores the distinction between minus signs, long vowel sounds, and dashes during a search. Read/write.

default border color RGB color

Returns or sets the default RGB color to use for new border objects. Read/write.

default border color index

Returns or sets the default line color index for borders. Read/write.

Can be one of the following:

- auto
- black
- blue
- turquoise
- bright green
- pink
- red
- yellow
- white
- dark blue
- teal
- green
- violet
- dark red
- dark yellow
- gray50
- gray25
- by author
- no highlight
default border line style

Returns or sets the default border line style. Read/write.

Can be one of the following:

- line style none
- line style single
- line style dot
- line style dash small gap
- line style dash large gap
- line style dash dot
- line style dash dot dot
- line style double
- line style triple
- line style thin thick small gap
- line style thick thin small gap
- line style thin thick thin small gap
- line style thin thick med gap
- line style thick thin med gap
- line style thick thick small gap
- line style thick thick med gap
- line style thick thick large gap
- line style dash dot stroked
- line style emboss_3D
- line style engrave_3D
- line style outset
- line style inset
- line style single wavy
- line style double wavy
- line style dash dot stroked
- line style emboss_3D
- line style engrave_3D
- line style outset
- line style inset

default border line width

Returns or sets the default line width of borders. Read/write.

Can be one of the following:

- line width25 point
- line width50 point
- line width75 point
- line width100 point
- line width150 point
- line width225 point
- line width300 point
- line width450 point
- line width600 point

default highlight color index

Returns or sets the color index used to highlight text formatted with the highlight button. Read/write.

Can be one of the following:

- auto
- black
- blue
- turquoise
- bright green
- pink
- red
- yellow
- white
- dark blue
- teal
- green
- violet
- dark red
- dark yellow
- gray50
- gray25
- by author
- no highlight
default open format

Returns or sets the default file converter used to open documents. Read/write.
Can be one of the following:

- open format auto
- open format document
- open format template
- open format rtf
- open format text
- open format Unicode text
- open format web pages
- open format mac readable

deprecated text color

Returns or sets the color of text that is deleted while change tracking is enabled. Read/write.
Can be one of the following:

auto  
black  
blue  
turquoise  
bright green  
pink  
red  
yellow  
white  
dark blue  
teal  
green  
violet  
dark red  
dark yellow  
gray50  
gray25  
by author  
nobold  
dark blue

deprecated text mark

Returns or sets the format of text that is deleted while change tracking is enabled. Read/write.
Can be one of the following:

deprecated text mark hidden  
deprecated text mark strike through  
deprecated text mark caret  
deprecated text mark pound  
deprecated text mark none  
deprecated text mark bold  
deprecated text mark italic  
deprecated text mark underline  
deprecated text mark double underline  
deprecated text mark color only

display grid lines

Returns or sets whether Word displays the document grid. This property affects only the document grid. For table gridlines, use the table gridlines property of the view object. Read/write.


Microsoft Word Suite

*display paste options*

Returns or sets whether Word displays the **Paste Options** button, which displays directly under newly pasted text. Read/write.

*dz match fuzzy*

Returns or sets whether Word ignores the distinction between some Japanese characters. Read/write.

*enable misused words dictionary*

Returns or sets whether Word checks for misused words when checking the spelling and grammar in a document. Word looks for the following when checking for misused words: incorrect usage of adjectives and adverbs, comparatives and superlatives, "like" as a conjunction, "nor" versus "or", "what" versus "which", "who" versus "whom", units of measurement, conjunctions, prepositions, and pronouns. Read/write.

*enable sound*

Returns or sets whether Word makes the computer respond with a sound whenever an error occurs. Read/write.

*envelope feeder installed*

Returns **true** if the current printer has a special feeder for envelopes. Read/write.

*fancy font menu*

Returns or sets whether the **Fancy Font** menu is shown. Read/write.

*grid distance horizontal small real*

Returns or sets the amount of horizontal space between the invisible gridlines that Word uses when you draw, move, and resize AutoShapes or East Asian characters in new documents. Read/write.

*grid distance vertical small real*

Returns or sets the amount of vertical space between the invisible gridlines that Word uses when you draw, move, and resize AutoShapes or East Asian characters in new documents. Read/write.

*grid origin horizontal small real*

Returns or sets the point, relative to the left edge of the page, where you want the invisible grid for drawing, moving, and resizing AutoShapes or East Asian characters to begin in new documents. Read/write.

*grid origin vertical small real*

Returns or sets the point, relative to the top of the page, where you want the invisible grid for drawing, moving, and resizing AutoShapes or East Asian characters to begin in new documents. Read/write.

*hf match fuzzy*

Returns or sets whether Word ignores the distinction between some Japanese characters. Read/write.
Microsoft Word Suite

*hiragana match fuzzy*

Returns or sets whether Word ignores the distinction between hiragana and katakana during a search. Read/write.

*ignore internet and file addresses*

Returns or sets whether file name extensions, paths, e-mail addresses, server and share names (also known as UNC paths), and Internet addresses (also known as URLs) are ignored while checking spelling. Read/write.

*ignore mixed digits*

Returns or sets whether words that contain numbers are ignored while checking spelling. Read/write.

*ignore uppercase*

Returns or sets whether words in all uppercase letters are ignored while checking spelling. Read/write.

*inline conversion*

Returns or sets whether Word displays an unconfirmed character string in the Japanese Input Method Editor as an insertion between existing character strings. Read/write.

*insert key for paste*

Returns or sets whether the insert key can be used for pasting the Clipboard contents. Read/write.

*inserted text color*

Returns or sets the color of text that is inserted while change tracking is enabled. Read/write.

Can be one of the following:

<table>
<thead>
<tr>
<th>auto</th>
<th>teal</th>
</tr>
</thead>
<tbody>
<tr>
<td>black</td>
<td>green</td>
</tr>
<tr>
<td>blue</td>
<td>violet</td>
</tr>
<tr>
<td>turquoise</td>
<td>dark red</td>
</tr>
<tr>
<td>bright green</td>
<td>dark yellow</td>
</tr>
<tr>
<td>pink</td>
<td>gray50</td>
</tr>
<tr>
<td>red</td>
<td>gray25</td>
</tr>
<tr>
<td>yellow</td>
<td>by author</td>
</tr>
<tr>
<td>white</td>
<td>no highlight</td>
</tr>
<tr>
<td>dark blue</td>
<td></td>
</tr>
</tbody>
</table>
inserted text mark

Returns or sets how Word formats inserted text while change tracking is enabled. If change tracking is not enabled, this property is ignored. Use this property with the inserted text color property to control the look of inserted text. Read/write. Can be one of the following:

- inserted text mark none
- inserted text mark bold
- inserted text mark italic
- inserted text mark underline
- inserted text mark double underline
- inserted text mark color only

iteration mark match fuzzy

Returns or sets whether Word ignores the distinction between types of repetition marks during a search. Read/write.

kanji match fuzzy

Returns or sets whether Word ignores the distinction between standard and nonstandard kanji ideography during a search. Read/write.

ki ku match fuzzy

Returns or sets whether Word ignores the distinction between some Japanese characters. Read/write.

live word count

Returns or sets whether the instant word count is displayed in the status bar. Read/write.

map paper size

Returns or sets whether documents formatted for another country's or region's standard paper size — for example, A4 — are automatically adjusted so that they're printed correctly on your country's/region's standard paper size — for example, Letter. This property affects only the printout of your document; its formatting is left unchanged. Read/write.

measurement unit

Returns or sets the standard measurement unit for Word. Read/write. Can be one of the following:

- inches
- centimeters
- millimeters
- points
- picas
Microsoft Word Suite

old kana match fuzzy

Returns or sets whether Word ignores the distinction between new kana and old kana characters during a search. Read/write.

overtype

Returns or sets whether Overtype mode is active. In Overtype mode, the characters you type replace existing characters one by one. When Overtype isn’t active, the characters you type move existing text to the right. Read/write.

pagination

Returns or sets whether Word repaginates documents in the background. Read/write.

paste adjust paragraph spacing

Returns or sets whether Word automatically adjusts the spacing of paragraphs when you cut and paste selections. Read/write.

paste adjust table formatting

Returns or sets whether Word automatically adjusts the formatting of tables when you cut and paste selections. Read/write.

paste adjust word spacing

Returns or sets whether Word automatically adjusts the spacing of words when you cut and paste selections. Read/write.

paste merge from Excel

Returns or sets whether text formatting will be merged when you paste from Microsoft Excel. Read/write.

paste merge from PowerPoint

Returns or sets whether text formatting will be merged when you paste from Microsoft PowerPoint. Read/write.

paste merge lists

Returns or sets whether the formatting of pasted lists will be merged with surrounding lists. Read/write.

paste smart cut paste

Returns or sets whether Word intelligently pastes selections into a document. Read/write.

paste smart style behavior

Returns or sets whether Word intelligently merges styles when you paste a selection from a different document. Read/write.

picture editor Unicode text

Returns or sets the name of the application to use to edit pictures. Read/write.
Microsoft Word Suite

print comments
Returns or sets whether Word prints comments, starting on a new page at the end of the document. Setting the print comments property to true automatically sets the print hidden text property to true. However, setting the print comments property to false has no effect on the setting of the print hidden text property. Read/write.

print drawing objects
Returns or sets whether Word prints drawing objects. Read/write.

print field codes
Returns or sets whether Word prints field codes instead of field results. Read/write.

print hidden text
Returns or sets whether hidden text is printed. Setting the print hidden text property to false automatically sets the print comments property to false. However, setting the print hidden text property to true has no effect on the print comments property. Read/write.

print properties
Returns or sets whether Word prints document summary information on a separate page at the end of the document. Read/write.

print reverse
Returns or sets whether Word prints pages in reverse order. Read/write.

prolonged sound mark match fuzzy
Returns or sets whether Word ignores the distinction between short and long vowel sounds during a search. Read/write.

punctuation match fuzzy
Returns or sets whether Word ignores the distinction between types of punctuation marks during a search. Read/write.

replace selection
Returns or sets whether the result of typing or pasting replaces the selection. If false, the result of typing or pasting is added before the selection, leaving the selection intact. Read/write.

revised lines color
Returns or sets the color of changed lines in a document with tracked changes. Read/write. Can be one of the following:

- auto
- black
- blue
- turquoise
- bright green
- pink
- red
- yellow
- white
- dark blue
t
- teal
- green
- violet
- dark red
- dark yellow
- gray50
- gray25
- by author
- no highlight
Revised lines mark

Returns or sets the placement of changed lines in a document with tracked changes. Read/write.

Can be one of the following:

- revised lines mark none
- revised lines mark left border
- revised lines mark right border
- revised lines mark outside border

Revised properties color

Returns or sets the color index used to mark formatting changes while change tracking is enabled. Read/write.

Can be one of the following:

- auto
- black
- blue
- turquoise
- bright green
- pink
- red
- yellow
- white
- dark blue
- teal
- green
- violet
- dark red
- dark yellow
- gray50
- gray25
- by author
- no highlight

Revised properties mark

Returns or sets the mark used to show formatting changes while change tracking is enabled. Read/write.

Can be one of the following:

- revised properties mark none
- revised properties mark bold
- revised properties mark italic
- revised properties mark underline
- revised properties mark double underline
- revised properties mark color only

Save interval

Returns or sets the time interval in minutes for saving AutoRecover information. Set the save interval property to 0 (zero) to turn off AutoRecover information. Read/write.
save normal prompt

Returns or sets whether Word prompts the user for confirmation to save changes to the Normal template before it quits. If this is set to false, Word automatically saves changes to the Normal template before it quits. Read/write.

save properties prompt

Returns or sets whether Word prompts for document property information when you save a new document. Read/write.

send mail attach

True if the Send To command on the File menu inserts the active document as an attachment to a mail message. False if the Send To command inserts the contents of the active document as text in a mail message. Read/write.

show readability statistics

Returns or sets whether Word displays a list of summary statistics, including measures of readability, when it has finished checking grammar. Read/write.

show wizard welcome

Returns or sets whether the Welcome wizard should be shown. Read/write.

small kana match fuzzy

Returns or sets whether Word ignores the distinction between diphthongs and double consonants during a search. Read/write.

smart cut paste

Returns or sets whether Word automatically adjusts the spacing between words and punctuation when you cut and paste. Read/write.

smart paragraph selection

Returns or sets whether Word includes the paragraph mark in a selection when you select most or all of a paragraph. Read/write.

snap to grid

Returns or sets whether AutoShapes or East Asian characters are automatically aligned with an invisible grid when they are drawn, moved, or resized in new documents. Read/write.

snap to shapes

Returns or sets whether Word automatically aligns AutoShapes or East Asian characters with invisible gridlines that go through the vertical and horizontal edges of other AutoShapes or East Asian characters in new documents. The snap to shapes property works independently of the snap to grid property. Read/write.

space match fuzzy

Returns or sets whether Word ignores the distinction between space markers used during a search. Read/write.
suggest from main dictionary only

Returns or sets whether Word draws spelling suggestions from the main dictionary only. If false, it draws spelling suggestions from the main dictionary and any custom dictionaries that have been added. Read/write.

suggest spelling corrections

Returns or sets whether Word always suggests alternative spellings for each misspelled word when checking spelling. Read/write.

tab indent key

Returns or sets whether the TAB and BACKSPACE keys can be used to increase and decrease, respectively, the left indent of paragraphs and whether the BACKSPACE key can be used to change right-aligned paragraphs to centered and centered paragraphs to left-aligned. Read/write.

tc match fuzzy

Returns or sets whether Word ignores the distinction between some Japanese characters. Read/write.

update fields at print

Returns or sets whether Word updates fields automatically before printing a document. Read/write.

update links at open

Returns or sets whether Word automatically updates all embedded OLE links in a document when it's opened. Read/write.

update links at print

Returns or sets whether Word updates fields automatically before printing a document. Read/write.

use character unit

Returns or sets whether Word uses characters as the default measurement unit for the current document. Read/write.

use german spelling reform

Returns or sets whether Word uses the German post-reform spelling rules when checking spelling. This property might not be available to you, depending on the language support (U.S. English, for example) that you've selected or installed. Read/write.

warn before saving printing sending markup

Returns or sets whether Word displays a warning when you save, print, or send as e-mail a document containing comments or tracked changes. Read/write.

zj match fuzzy

Returns or sets whether Word ignores the distinction between some Japanese characters. Read/write.
Class: work menu item

Plural

work menu items

Represents a single item on the Work menu.

Use work menu item index, where index is the Work menu item name or index number, to return a work menu item object. The following example deletes the first item from the Work menu.

delete work menu item 1

The following example displays the name of the second item on the Work menu.

if (count of work menu items) ≥ 2 then display dialog (get name of ¬
     work menu item 2)

To add an item to the Work menu, use the make command. The following example adds the document named Hello to the Work menu.

make new work menu item with properties ¬
    {path:"Macintosh HD:Users:Shared:Hello"}

Properties

document to open

Returns the document object to open for the specified work menu item. This property will always return missing value. Read-only.

entry index

Returns a number that indicates the position of an item in a list. Read-only.

name

Returns the name of the work menu item. Read-only.

path

Returns the path to the specified work menu item object. Read-only.

Note  The path doesn't include a trailing character; for example, Macintosh HD:Users:Shared.

read only

True if changes to the document cannot be saved to the original document. Read/write.

Class: zoom

Plural

zooms

Contains magnification options (for example, the zoom percentage) for a window or pane.

To return a single zoom object for a window, use the zoom property of the view object. The following example sets the zoom percentage for the active window to 110 percent.

set percentage of zoom of view of active window to 110
To return a single **zoom** object for a pane, use the `get zoom` command. The following example sets the page layout magnification for the active window so that an entire page is visible.

```vba
set page fit of (get zoom active pane of active window zoom type page view) to page fit full page
```

The `make` command isn't available for the **zooms** list. The **zooms** list includes a single **zoom** object for each of the various view types (outline, normal, page layout, and so on).

**Properties**

*page columns*

Returns or sets the number of pages to be displayed side by side on-screen at the same time in page layout view or print preview. Read/write.

*page fit*

Returns or sets the view magnification of a window so that either the entire page is visible or the entire width of the page is visible. Read/write.

Can be one of the following:

- **page fit best fit**
- **page fit full page**
- **page fit none**

The **page fit full page** constant has no effect if the document isn't in page layout view.

When the **page fit** property is set to **page fit best fit**, the zoom percentage is automatically recalculated every time the document window size is changed. Setting this property to **page fit none** keeps the zoom percentage from being recalculated whenever this happens.

*page rows*

Returns or sets the number of pages to be displayed one above the other on-screen at the same time in page layout view or print preview. Read/write.

*percentage*

Returns or sets the magnification for a window as a percentage. Read/write.
Commands

accept .......................................................................................................................... 216
accept all revisions ................................................................................................... 216
activate object .......................................................................................................... 217
append to spike ....................................................................................................... 217
apply bullet default ................................................................................................. 218
apply list format template ....................................................................................... 218
apply list template .................................................................................................. 219
apply number default ............................................................................................. 220
apply page borders to all sections ......................................................................... 220
apply theme ............................................................................................................. 221
auto format .............................................................................................................. 221
auto summarize ...................................................................................................... 222
automatic change ................................................................................................... 222
break link ............................................................................................................... 223
build key code ....................................................................................................... 223
calculate selection ................................................................................................ 224
can continue previous list ...................................................................................... 224
centimeters to points ............................................................................................. 225
collapse outline ..................................................................................................... 225
check ......................................................................................................................... 226
check consistency .................................................................................................... 226
check grammar ....................................................................................................... 226
check spelling ......................................................................................................... 227
clean string ............................................................................................................ 227
clear .......................................................................................................................... 228
clear all fuzzy options ........................................................................................... 229
clear formatting ...................................................................................................... 230
click object .............................................................................................................. 231
close print preview ................................................................................................. 231
collapse outline ..................................................................................................... 232
compare .................................................................................................................. 232
compute statistics .................................................................................................. 233
convert ..................................................................................................................... 233
copy numbers to text .............................................................................................. 234
copy bookmark ...................................................................................................... 234
copy format ............................................................................................................ 235
copy object ............................................................................................................. 235
copy styles from template ...................................................................................... 235
count numbered items ............................................................................................ 236
create data source ................................................................................................. 237
create header source ............................................................................................... 238
create letter content ............................................................................................... 239
create new document for hyperlink ...................................................................... 240
create new mailing label document ...................................................................... 241
create range ............................................................................................................ 242
create text box ....................................................................................................... 243
cut object ................................................................................................................. 243
data form ................................................................................................................ 244
disable ...................................................................................................................... 244
display Word dialog ............................................................................................... 245
do Word repeat ...................................................................................................... 246
do away .................................................................................................................... 246
edit data source .................................................................................................... 246
edit header source ................................................................................................. 247
edit main document ............................................................................................... 247
edit type ................................................................................................................... 248
enable........................................................................................................................................249
end key.......................................................................................................................................249
endnote convert..................................................................................................................................250
escape key.......................................................................................................................................250
execute data merge.........................................................................................................................250
execute dialog..................................................................................................................................251
execute find.......................................................................................................................................251
execute key binding.........................................................................................................................253
expand ...............................................................................................................................................253
extend...............................................................................................................................................254
find key...............................................................................................................................................254
find record........................................................................................................................................254
fit to pages.......................................................................................................................................255
follow...............................................................................................................................................256
follow hyperlink...............................................................................................................................256
footnote convert...............................................................................................................................257
get active writing style.....................................................................................................................258
get border.........................................................................................................................................258
get cross reference items................................................................................................................259
get default file path .........................................................................................................................260
get default theme for.......................................................................................................................260
get dialog..........................................................................................................................................260
get document compatibility............................................................................................................261
get international information........................................................................................................264
get keys bound to.............................................................................................................................264
get list gallery..................................................................................................................................265
get next field.....................................................................................................................................265
get previous field............................................................................................................................266
get private profile string.................................................................................................................266
get profile string............................................................................................................................267
get selection information...............................................................................................................267
get spelling suggestions..................................................................................................................270
get story range...............................................................................................................................271
get synonym info object................................................................................................................272
get theme name..............................................................................................................................273
get webpage font...........................................................................................................................273
get zoom...........................................................................................................................................274
home key..........................................................................................................................................275
inches to points..............................................................................................................................275
insert...............................................................................................................................................275
insert auto text...............................................................................................................................276
insert auto text entry......................................................................................................................276
insert caption....................................................................................................................................277
insert caption....................................................................................................................................277
insert cells.........................................................................................................................................278
insert columns...............................................................................................................................278
insert cross reference.....................................................................................................................279
insert database.............................................................................................................................279
insert date time..............................................................................................................................280
insert envelope data......................................................................................................................282
insert file..........................................................................................................................................283
insert formula....................................................................................................................................284
insert paragraph.............................................................................................................................285
insert rows.......................................................................................................................................286
insert symbol....................................................................................................................................287
key string..........................................................................................................................288
large scroll ..........................................................................................................................288
lines to points ......................................................................................................................289
list commands .....................................................................................................................289
list indent ..............................................................................................................................289
list outset .............................................................................................................................290
make compatibility default .................................................................................................290
make new data merge ask field ..........................................................................................291
make new data merge fill in field .........................................................................................291
make new data merge if field ...............................................................................................292
make new data merge next field .........................................................................................292
make new data merge next if field ......................................................................................293
make new data merge rec field ..........................................................................................294
make new data merge sequence field ...............................................................................294
make new data merge set field ..........................................................................................295
make new data merge skip if field ......................................................................................296
manual hyphenation ............................................................................................................296
mark entry for table of contents ........................................................................................297
mark entry for table of figures ...........................................................................................297
mark for index .......................................................................................................................298
merge ...................................................................................................................................299
merge subdocuments ........................................................................................................299
millimeters to points ...........................................................................................................300
modified ..............................................................................................................................300
next for browser ................................................................................................................300
next for header footer .......................................................................................................301
next revision .......................................................................................................................301
on time ................................................................................................................................301
open as document .................................................................................................................302
open data source ................................................................................................................303
open document from workmenu ..........................................................................................304
open header source .............................................................................................................304
open recent file .....................................................................................................................305
open subdocument .............................................................................................................305
open version .........................................................................................................................306
organizer copy ....................................................................................................................306
organizer delete ..................................................................................................................307
organizer rename ...............................................................................................................307
page scroll .........................................................................................................................308
paste format .........................................................................................................................308
paste object ........................................................................................................................309
picas to points ......................................................................................................................310
points to centimeters ..........................................................................................................310
points to inches ..................................................................................................................310
points to millimeters ..........................................................................................................311
points to picas .....................................................................................................................311
present it ..............................................................................................................................311
previous for browser .........................................................................................................312
previous header footer ......................................................................................................312
previous revision ...............................................................................................................312
print out ..............................................................................................................................312
print out envelope .............................................................................................................314
print out mailing label ......................................................................................................315
print preview ......................................................................................................................316
rebind .................................................................................................................................316
...
Microsoft Word Suite

Command: accept

Accepts the specified tracked change. The revision marks are removed, and the change is incorporated into the document.

Syntax

accept revision  Required. An expression that returns a revision object.

Example

This example accepts all of the tracked changes found if the change type is inserted text.

repeat with aRev in (get revisions of active document)
    if (revision type of aRev) is revision insert then accept aRev
end repeat

This example accepts the first revision in the selection.

accept revision 1 of selection

Command: accept all revisions

Accepts all tracked changes in the specified document.

Syntax

accept all revisions  document  Required. An expression that returns a document object.

Example

This example checks the main story in the active document for tracked changes, and if there are any, the example incorporates all revisions in all stories in the document.

if (count of revisions of active document) ≥ 1 then
    accept all revisions active document
Command: activate object
Activates the specified object.

Syntax
activate object document/window/pane  Required. An expression that returns a document, window, or pane object.

Example
This example activates the document named "Sales".
activate object document "Sales"
This example activates the second window in the windows list.
activate object window 2
This example splits the active window and then activates the first pane.
set split vertical of active window to 50
activate object pane 1 of active window

Command: append to spike
Deletes the specified range and adds the contents of the range to the Spike (a built-in AutoText entry). This command returns the Spike as an auto text entry object.

Syntax
append to spike template  Required. An expression that returns a template object.

range  text range  Required text range object. The range that’s deleted and appended to the Spike.

Example
This example deletes the selection and adds its contents to the Spike in the Normal template.
if (count of characters of text object of selection) > 1 then
   append to spike normal template range text object of selection
end if
This example clears the Spike and adds the first and third words in the active document to the Spike in the attached template. The contents of the Spike are then inserted at the insertion point.
collapse range text object of selection direction collapse start
set myTemplate to attached template of active document
repeat with entry in (get auto text entries of myTemplate)
   if name of entry is "Spike" then delete entry
end repeat
append to spike myTemplate range word 1 of active document
append to spike myTemplate range word 2 of active document
insert auto text entry auto text entry "Spike" of myTemplate where ¬
text object of selection
Command: apply bullet default

Adds bullets and formatting to the paragraphs in the range for the specified list format object. If the paragraphs are already formatted with bullets, this method removes the bullets and formatting.

**Syntax**

`apply bullet default` list format  
Required. An expression that returns a list format object.

`[default list behavior  enumeration]` Optional. Sets a value that specifies whether Word uses Web-oriented formatting for better list display. Can be either of the following constants: `word8 list behavior` (use formatting compatible with Microsoft Word 98) or `word9 list behavior` (use Web-oriented formatting). For compatibility reasons, the default constant is `word8 list behavior`, but in new procedures you should use `word9 list behavior` to take advantage of improved Web-oriented formatting with respect to indenting and multilevel lists.

**Example**

This example adds bullets and formatting to the paragraphs in the selection. If there are already bullets in the selection, the example removes the bullets and formatting.

```plaintext
apply bullet default list format of text object of selection
```

This example adds a bullet and formatting to, or removes them from, the second paragraph in MyDoc.doc.

```plaintext
apply bullet default list format of text object of paragraph 2 of document "MyDoc.doc"
```

This example sets the variable myRange to a range that includes paragraphs three through six of the active document, and then it checks to see whether the range contains list formatting. If there's no list formatting, default bullets are added.

```plaintext
set myDoc to active document
set myRange to create range myDoc start (start of content of text object of paragraph 3 of myDoc) end (end of content of text object of paragraph 6 of myDoc)
if list type of list format of myRange is list no numbering then
   apply bullet default list format of myRange
end if
```

Command: apply list format template

Applies a set of list-formatting characteristics to the specified list format object.

**Syntax**

`apply list format template` list format  
Required. An expression that returns a list format object.

`list template` list template  
Required list template object. The list template to be applied.

`[continue previous list]` Boolean. Optional. True to continue the numbering from the previous list; false to start a new list.

`[apply to enumeration]` Optional. The portion of the list that the list template is to be applied to. Can be one of the following: `list apply to selection`, `list apply to whole list`, or `list apply to this point forward`.
This example sets the variable myRange to a range in the active document, and then it checks to see whether the range has list formatting. If no list formatting has been applied, the fourth outline-numbered list template is applied to the range.

set myDoc to active document
set myRange to create range myDoc start (start of content of text object of paragraph 3 of myDoc) end (end of content of text object of paragraph 6 of myDoc)
if list type of list format of myRange is list no numbering then
apply list format template list format of myRange list template of (list template 4 of (get list gallery outline number gallery))
end if

This example sets the variable myLstRange to the list formatting in the second paragraph of MyDocument.doc. The example then applies the third numbered list template from that point forward in the list.

set myLstRange to list format of text object of paragraph 2 of document "MyDocument.doc"
apply list format template myLstRange list template (list template 3 of (get list gallery number gallery)) ~
apply to list apply to this point forward

Command: apply list template

Applies a set of list-formatting characteristics to the specified list.

Syntax

apply list template  Word list  Required. An expression that returns a Word list object.
list template  list template  Required list template object. The list template to be applied.
[continue previous list  Boolean]  Optional. True to continue the numbering from the previous list; false to start a new list.
[default list behavior  enumeration]  Optional. Sets a value that specifies whether Word uses new Web-oriented formatting for better list display. Can be either of the following constants:
word8 list behavior (use formatting compatible with Microsoft Word 98) or word9 list behavior (use Web-oriented formatting). For compatibility reasons, the default constant is word8 list behavior, but in new procedures you should use word9 list behavior to take advantage of improved Web-oriented formatting with respect to indenting and multilevel lists.
Example
This example sets the variable myList to the fourth list in MyDocument.doc, and then it applies the third bulleted list template to the list.

```plaintext
set myList to Word list 4 of document "MyDocument.doc"
apply list template myList list template (list template 3 of ~
   (get list template bullet gallery))
```

Command: apply number default

Adds the default numbering scheme to the paragraphs in the range for the specified list format object. If the paragraphs are already formatted as a numbered list, this command removes the numbers and formatting.

Syntax

```plaintext
apply number default list format required. An expression that returns a list format object.
```

```plaintext
default list behavior enumeration] Optional. Sets a value that specifies whether Word uses new Web-oriented formatting for better list display. Can be either of the following constants: word8 list behavior (use formatting compatible with Microsoft Word 98) or word9 list behavior (use Web-oriented formatting). For compatibility reasons, the default constant is word8 list behavior, but in new procedures you should use word9 list behavior to take advantage of improved Web-oriented formatting with respect to indenting and multilevel lists.
```

Example
This example numbers the paragraphs in the selection. If the selection is already a numbered list, the example removes the numbers and formatting.

```plaintext
apply number default list format of text object of selection
```

This example sets the variable myRange to include paragraphs three through six of the active document, and then it checks to see whether the range contains list formatting. If there's no list formatting, default numbers are applied to the range.

```plaintext
set myDoc to active document
set myRange to create range myDoc start (start of content of text object ~
   of paragraph 3 of myDoc) end (end of content of text object of ~
   paragraph 6 of myDoc)
if list type of list format of myRange is list no numbering then
   apply number default list format of myRange
end if
```

Command: apply page borders to all sections

Applies the specified page-border formatting to all sections in a document.

Syntax

```plaintext
apply page borders to all sections border options required. An expression that returns a border options object.
```

220
Example

This example adds a single-line page border to all sections in the active document.

```
set s1 to section 1 of active document
set outside line style of border options of s1 to line style single
set outside line width of border options of s1 to line width 50 point
apply page borders to all sections border options of s1
```

Command: apply theme

Applies a theme to a document.

Syntax

```
apply theme   document  Required. An expression that returns a document object.
name          Unicode text  Required. The name of the theme plus any theme formatting options you
              want to apply. The format of this string is "theme nnn \" where theme and nnn are defined as
              follows:

<table>
<thead>
<tr>
<th>String</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>theme</td>
<td>The name of the folder that contains the data for the requested theme. (The default location for theme data folders is Microsoft Office 2004/Office/Themes.) You must use the folder name for the theme rather than the display name that appears in the Theme dialog box (Theme command, Format menu).</td>
</tr>
<tr>
<td>nnn</td>
<td>A three-digit string that indicates which theme formatting options to activate (1 to activate, 0 to deactivate). The digits correspond to the Vivid Colors, Active Graphics, and Background Image check boxes in the Theme dialog box (Theme command, Format menu). If this string is omitted, the default value for nnn is &quot;011&quot; (Active Graphics and Background Image are activated).</td>
</tr>
</tbody>
</table>
```

Example

This example applies the Artsy theme to the active document and activates the Vivid Colors option.

```
apply theme active document name "artsy 100"
```

Command: auto format

Automatically formats a document.

Syntax

```
auto format   document  Required. An expression that returns a document object.
```

Example

This example automatically formats the active document.

```
auto format active document
```
Command: auto summarize

Creates an automatic summary of the specified document, and returns a text range object. Corresponds to the AutoSummarize options (Tools menu).

Syntax

auto summarize  document  Required. An expression that returns a document object.

[auto summarize length  integer]  Optional. The length of the summary as a percentage of the total document length (the larger the number, the more detail that's included in the summary).

[summary mode  enumeration]  Optional. Specifies the way the summary is displayed. Can be one of the following.

<table>
<thead>
<tr>
<th>Constant</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>summary mode highlight</td>
<td>Highlights the key points in the specified document and displays the AutoSummarize toolbar.</td>
</tr>
<tr>
<td>summary mode insert</td>
<td>Inserts a summary at the beginning of the specified document.</td>
</tr>
<tr>
<td>summary mode create new</td>
<td>Creates a new document and inserts the summary.</td>
</tr>
<tr>
<td>summary mode hide all but summary</td>
<td>Hides everything except the summary and displays the AutoSummarize toolbar.</td>
</tr>
</tbody>
</table>

[update properties  Boolean]  Optional. True to update the keyword and comment text in the Properties dialog box (File menu) to reflect the content of the summary for the specified document.

Example

This example creates an automatic summary of the active document by highlighting its key points.

auto summarize active document auto summarize length 30 summary mode summary mode highlight

Command: automatic change

Performs an AutoFormat action when there's a change suggested by the Office Assistant. If no AutoFormat action is active, this command generates an error.

Syntax

automatic change

Example

This example completes an Office Assistant AutoFormat action if one is active.

automatic change
Command: break link

Breaks the link between the source file and the specified picture.

**Note**  After you use this command, the link result won't be automatically updated if the source file is changed.

**Syntax**

```
break link  link format   Required. An expression that returns a link format object.
```

**Example**

This example updates and then breaks the links to any shapes that are linked OLE objects in the active document (Windows only).

```
repeat with s in (get shapes of active document)
    if shape type of s is shape type linked OLE object then
        update link format of s
        break link link format of s
    end if
end repeat
```

Command: build key code

Returns a unique number for the specified key combination.

**Syntax**

```
build key code
```

- `key1` enumeration Required. A key you specify by using one of the specified constants. See the Microsoft Word AppleScript Dictionary for the list of constants.

- `[key2–key4 enumeration]` Optional. A key you specify by using one of the specified constants. See the Microsoft Word AppleScript Dictionary for the list of constants.

**Example**

This example assigns the CMD + F1 key combination to the Organizer command.

```
set customization context to normal template
make new key binding with properties ¬
    {key code:(build key code key1 command_key key2 f1_key), ¬
     key category:command, command:"Organizer"}
```

This example removes the CMD+F1 key assignment from the Normal template.

```
set customization context to normal template
clear (find key key code (build key code key1 command_key key2 f1_key))
```

This example displays the command assigned to the F1 key.

```
set customization context to normal template
display dialog (get command of (find key key code (build key code ¬
    key1 f1_key)))
```
Command: calculate selection

Calculates a mathematical expression within a range or selection. Returns the result as a small real number.

Syntax

```
calculate selection  Required. An expression that returns a selection object.
```

Example

This example calculates the selected mathematical expression and displays the result.

```
display dialog "And the answer is... " & (calculate selection selection)
```

Command: can continue previous list

Returns a constant (continue disabled, reset list, or continue list) that indicates whether the formatting from the previous list can be continued.

Syntax

```
can continue previous list  list format/Word list  Required. An expression that returns a Word list or list format object.
```

list template  list template  Required list template object. A list template that's been applied to previous paragraphs in the document.

Remarks

This method returns the state of the Continue previous list and Restart numbering options in the Bullets and Numbering dialog box for a specified list format. To change the settings of these options, set the continue previous list argument of the apply list template command.

Example

This example checks to see whether numbering from a previous list is disabled. If it isn't disabled, the current list template is applied with numbering set to continue from the previous list. The selection must be within the second list, or this example creates an error.

```
set myLF to list format of text object of selection
set temp to can continue previous list myLF list template (list template ¬ of myLF)
if temp is not continue disabled then
    apply list template Word list of selection list template (list template ¬ of myLF) with continue previous list
end if
```
Command: centimeters to points

Converts a measurement from centimeters to points (1 cm = 28.35 points). Returns the converted measurement as a small real number.

Syntax

centimeters to points

centimeters  small real  Required. The centimeter value to be converted to points.

Example

This example adds a centered tab stop to the first paragraph in the selection. The tab stop is positioned at 1.5 centimeters from the left margin.

make new tab stop at paragraph 1 of selection with properties ¬
   {tab stop position: (centimeters to points centimeters 1.5), ¬
   alignment:align tab center}

This example sets a first-line indent of 2.5 centimeters for the first paragraph in the active document.

set first line indent of paragraph 1 of active document to ¬
   centimeters to points centimeters 2.5

Command: change file open directory

Sets the folder in which Word searches for documents. The specified folder's contents are listed the next time the Open dialog box (File menu) is displayed.

Note  Word searches the specified folder for documents until the user changes the folder in the Open dialog box or the current Word session ends. Use the set default file path command to change the default folder for documents in every Word session

Syntax

change file open directory

   path  Unicode text  Required. The path to the folder in which Word searches for documents.

Example

This example changes the folder in which Word searches for documents, and then opens a file named "Test.doc."

change file open directory path "Macintosh HD:Users:Shared:"
open document file name "Test.doc"
Command: check
Simulates the mail merge operation, pausing to report each error as it occurs.

Syntax
check  data merge  Required. An expression that returns a data merge object.

Example
This example checks the active document for mail merge errors.
set theState to state of data merge of active document
if (theState is main and data source) or ¬
    (theState is main and source and header) then
    check data merge of active document
end if

Command: check consistency
Checks Japanese text for consistency of phonetic spelling and ideographic character usage.

Syntax
check consistency  document  Required. An expression that returns a document object.

Remarks
Japanese language features are available only in the Japanese version of Word or in an English version of Word that you have registered as Japanese by using the Microsoft Language Register application.

Example
This example checks the consistency of Japanese characters in the active document.
check consistency active document

Command: check grammar
Checks a string for grammatical errors. Returns True if the string contains no errors.

Syntax
check grammar  application/document  Required. An expression that returns an application or document object.

    text to check  Unicode text  Required. The string you want to check for grammatical errors.
Example
This example displays the result of a grammar check on the selection.
set pass to check grammar text to check (get content of text object of selection)
display dialog "The selection is grammatically correct: " & pass

Command: check spelling
Checks a string for spelling errors. Returns true if the string has no spelling errors.

Syntax
check spelling application/document An expression that returns an application or document object.
text to check Unicode text Required. The text whose spelling is to be checked.
[custom dictionary dictionary] Optional. Either an expression that returns a dictionary object or the file name of the custom dictionary.
[ignore uppercase Boolean] Optional. True if capitalization is ignored. If this argument is omitted, the current value of the ignore uppercase property of the Word options class is used.
[main dictionary dictionary] Optional. Either an expression that returns a dictionary object or the file name of the main dictionary.
[custom dictionary2–custom dictionary10 dictionary] Optional. Either an expression that returns a dictionary object or the file name of an additional custom dictionary. You can specify as many as nine additional dictionaries.

Example
This example displays the result of a spelling check on the selection.
set pass to check spelling text to check (get content of text object of selection)
display dialog "The selection has no spelling errors: " & pass

Command: clean string
Removes nonprinting characters (character codes 1-29) and special Word characters from the specified string or changes them to spaces (character code 32), as described in the "Remarks" section. Returns the result as a string.

Syntax
clean string

item to check Unicode text Required. The source string that you want to clean.
Remarks
The following characters are converted as described in this table.

<table>
<thead>
<tr>
<th>Character code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>7 (beep)</td>
<td>Removed unless preceded by character 13 (paragraph), then converted to character 9 (tab).</td>
</tr>
<tr>
<td>10 (line feed)</td>
<td>Converted to character 13 (paragraph) unless preceded by character 13, then removed.</td>
</tr>
<tr>
<td>13 (paragraph)</td>
<td>Unchanged.</td>
</tr>
<tr>
<td>31 (optional hyphen)</td>
<td>Removed.</td>
</tr>
<tr>
<td>194 (optional hyphen)</td>
<td>Removed.</td>
</tr>
<tr>
<td>202 (nonbreaking space)</td>
<td>Converted to character 32 (space).</td>
</tr>
</tbody>
</table>

Example
This example removes nonprinting characters from the selected text and inserts the result into a new document.

`set clean to clean string item to check (content of text object of selection)`
`set myDoc to make new document`
`set content of text object of myDoc to clean`

This example removes nonprinting characters from the selected field code and then displays the result.

`set show field codes of view of active window to true`
`select field 1 of active document`
`display dialog (clean string item to check content of text object of selection)`

Command: clear
- **drop cap** object: Removes the dropped capital letter formatting.
- **tab stop** object: Removes the specified custom tab stop.
- **text input** object: Deletes the text from the specified text form field.
- **key binding** object: Removes the key binding from the key bindings list and resets a built-in command to its default key assignment.

Syntax
- **clear**  drop cap/tab stop/text input/key binding  Required. An expression that returns a **drop cap**, **tab stop**, **text input**, or **key binding** object.
Example
This example clears the first custom tab in the first paragraph of the active document.
clear tab stop 1 of paragraph 1 of active document
This example protects the document for forms and deletes the text from the first form field if the field is a text form field.
protect active document protection type allow only form fields with no reset
if form field type of form field 1 of active document is ¬
  field form text input then clear text input of ¬
  form field 1 of active document
This example removes dropped capital letter formatting from the first letter in the active document.
set drop1 to drop cap of paragraph 1 of active document
if drop position of drop1 is not drop none then clear drop1
This example removes the CMD+F1 key assignment from the Normal template.
set customization context to normal template
clear (find key key code (build key code key1 command_key key2 f1_key))

Command: clear all fuzzy options
Clears all nonspecific search options associated with Japanese text.

Syntax
clear all fuzzy options  find  Required. An expression that returns a find object.

Remarks
This command sets the following properties to false:

ay match fuzzy  kanji match fuzzy
bv match fuzzy  ki ku match fuzzy
byte match fuzzy  old kana match fuzzy
case match fuzzy  prolonged sound mark match fuzzy
dash match fuzzy  punctuation match fuzzy
dz match fuzzy  small kana match fuzzy
hf match fuzzy  space match fuzzy
hiragana match fuzzy  tc match fuzzy
iteration mark match fuzzy  zj match fuzzy
Example

This example clears all nonspecific options before executing a search in the selected range. If the
word "4½" is formatted as bold, the entire paragraph will be selected and copied to the
Clipboard.

clear formatting find object of selection
clear all fuzzy options find object of selection
set bold of font object of find object of selection to true
set utxtViolin to «data utxt30D030A430AA30EA30F3»
execute find find object of selection find text utxtViolin with ¬
        find format and match forward
if found of find object of selection is true then
        expand selection by a paragraph item
        copy object selection
end if

Command: clear formatting

Removes any formatting specified as part of a find or replace operation. Corresponds to the No
Formatting button in the Find and Replace dialog box (Edit menu).

Note  To ensure that unwanted formats aren't included as criteria in a find or replace operation, use
this command before carrying out the operation.

Syntax

clear formatting  find/replacement/selection object  Required. An expression that returns a find,
replacement, or selection object.

Example

This example clears formatting from the find or replace criteria before replacing the word "Inc." with
"incorporated" throughout the active document.

set myRange to text object of active document
clear formatting find object of myRange
clear formatting replacement of find object of myRange
set match whole word of find object of myRange to true
execute find find object of myRange find text "Inc." ¬
        replace with "incorporated" replace replace all
This example removes formatting from the find criteria before searching through the selection. If the word "Hello" with bold formatting is found, the entire paragraph is selected and copied to the Clipboard.

```
clear formatting find object of selection
set bold of font object of find object of selection to true
execute find find object of selection find text "Hello" with find format and ¬
        match forward
if found of find object of selection is true then
    expand selection by a paragraph item
    copy object selection
end if
```

Command: `click object`

Clicks the specified field. If the field is a GOTOBUTTON field, this command moves the insertion point to the specified location or selects the specified bookmark. If the field is a MACROBUTTON field, this command runs the specified macro. If the field is a HYPERLINK field, this command jumps to the target location.

**Syntax**

```
click object  field  Required. An expression that returns a field object.
```

**Example**

If the first field in the selection is a GOTOBUTTON field, this example clicks it (the insertion point is moved to the specified location, or the specified bookmark is selected).

```
set myField to field 1 of selection
if field type of myField is field go to button then click object myField
```

Command: `close print preview`

Switches the specified document from print preview to the previous view. If the specified document isn't in print preview, an error occurs.

**Syntax**

```
close print preview  document  Required. An expression that returns a document object.
```

**Example**

This example switches the active window from print preview to normal view.

```
if do print preview is true then close print preview active document
set view type of view of active window to normal view
```
**Command: collapse outline**

Collapses the text under the specified text range by one heading level.

**Note** If the document isn't in outline or master document view, an error occurs.

**Syntax**

`collapse outline`  
Required. An expression that returns a `view` object.

`[text range]`  
Optional. The text range of paragraphs to be collapsed. If this argument is omitted, the entire selection is collapsed.

**Example**

This example applies the Heading 2 style to the second paragraph in the active document, switches the active window to outline view, and collapses the text under the second paragraph in the document.

```
set style of paragraph 2 of active document to style heading2
set view type of view of active window to outline view
collapse outline view of active window text range text object of paragraph 2 of active document
```

This example collapses every heading in the document by one level.

```
set view type of view of active window to outline view
collapse outline view of active window text range text object of active document
```

**Command: compare**

Displays revision marks that indicate where the specified document differs from another document.

**Syntax**

`compare`  
Required. An expression that returns a `document` object.

`path`  
Required. The name and path of the document that the specified document is compared with.

**Note** The document you want to use as the base document to compare the specified document to must be the active document for revision marks to appear.

**Example**

This example compares the active document with the document named "First Rev.doc" in the Shared folder.

```
compare active document path "Macintosh HD:Users:Shared:First Rev.doc"
```
Command: compute statistics

Returns a statistic based on the contents of the specified document. You must use the compute statistics command before you can access the readability statistics for the document.

Syntax

compute statistics document Required. An expression that returns a document object.

statistic enumeration Required. The statistic. Can be one of the following: statistic characters, statistic characters with spaces, statistic east asian characters, statistic lines, statistic pages, statistic paragraphs, or statistic words.

[include footnotes and endnotes Boolean] Optional. True to include footnotes and endnotes when computing statistics. If this argument is omitted, the default value is false.

Example

This example displays the number of words and characters in the first paragraph of Report.doc.

set wordCount to compute statistics document "Report.doc" statistic statistic words
set charCount to compute statistics document "Report.doc" statistic statistic characters
display dialog "The document contains " & wordCount & " words and a total of " & charCount & " characters."

This example displays the number of words in the active document, including footnotes.
display dialog "" & (compute statistics active document statistic statistic words with include footnotes and endnotes true) & " words."

Command: convert

Converts a multiple-level list to a single-level list, or vice versa.

Syntax

convert list template Required. An expression that returns a list template object.

[level integer] Optional. The level to use for formatting the new list. When this argument is used to convert a multiple-level list to a single-level list, it can be a number from 1 through 9. When this argument is used to convert a single-level list to a multiple-level list, 1 is the only valid value. If this argument is omitted, 1 is used.

Remarks

You cannot use the convert command on a list template that is derived from the list galleries object.

Example

This example converts the first list template in the active document. If the list template is multiple-level, it becomes single-level, or vice versa.

convert list template 1 of active document
Command: convert numbers to text

Changes the list numbers and LISTNUM fields in the specified document, Word list, or list format object to text.

Syntax

convert numbers to text  document/list format/Word list  Required. An expression that returns a document, Word list, or list format object.

[number type  enumeration]  Optional. The type of number to be converted. Can be one of the following: number paragraph, number listnum, or number all numbers. The default value is number all numbers.

Remarks

There are two types of numbers: preset numbers (number paragraph), which you can add to paragraphs by selecting a template in the Bullets and Numbering dialog box; and LISTNUM fields (number listnum), which allow you to add more than one number per paragraph.

The convert numbers to text command is useful if you want to work with a document in another application and that application doesn't recognize list formatting or LISTNUM fields.

Note  After you convert list numbers to text, you can no longer manipulate them in a list.

Example

This example converts the list numbers and LISTNUM fields in the active document to text.

convert numbers to text active document

This example converts the numbers in the first list to text.

convert numbers to text Word list 1 of active document

This example converts the preset numbers in myRange to text without affecting any LISTNUM fields.

set myDoc to active document
set myRange to create range myDoc start (start of content of text object of paragraph 12 of myDoc) end (end of content of text object of paragraph 20 of myDoc)
convert numbers to text list format of myRange number type number paragraph

Command: copy bookmark

Sets the bookmark specified by the name argument to the location marked by another bookmark, and returns a bookmark object.

Syntax

copy bookmark  bookmark  Required. An expression that returns a bookmark object.

name  Unicode text  Required. The name of the new bookmark.

Example

This example sets the Book2 bookmark to the location marked by the Book1 bookmark.

copy bookmark bookmark "book1" of active document name "book2"
**Command: copy format**

Copies the character formatting of the first character in the selected text. If a paragraph mark is selected, Word copies paragraph formatting in addition to character formatting.

**Note** You can apply the copied formatting to another selection by using the **paste format** command.

**Syntax**

```
copy format  selection  Required. An expression that returns a selection object.
```

**Example**

This example copies the formatting of the first paragraph to the second paragraph in the active document.

```
select text object of paragraph 1 of active document
copy format selection
select text object of paragraph 2 of active document
paste format selection
```

**Command: copy object**

Copies the content of the specified object to the Clipboard.

**Syntax**

```
copy object  field/frame/form field/data merge field/selection object/page number  Required. An expression that returns a field, form field, frame, data merge field, page number, or selection object.
```

**Example**

This example copies the contents of the selection into a new document.

```
if selection type of selection is selection normal then
  copy object selection
  make new document
  paste object selection
end if
```

**Command: copy styles from template**

Copies styles from the specified template to a document.

**Syntax**

```
copy styles from template  document  Required. An expression that returns a document object.
```

**Remarks**

When styles are copied from a template to a document, like-named styles in the document are redefined to match the style descriptions in the template. Unique styles from the template are copied to the document. Unique styles in the document remain intact.
Example
This example copies the styles from the active document's template to the document.
copy styles from template active document template full name of ¬
attached template of active document
This example copies the styles from the Sales96.dot template to Sales.doc.
copy styles from template active document template ¬
"Macintosh HD:Users:Shared:Sales96.dot"

Command: count numbered items
Returns the number of bulleted or numbered items and LISTNUM fields in the specified document, Word list, or list format object.

Syntax
count numbered items  document/list format/Word list  Required. An expression that returns a
document, Word list, or list format object.

  [number type  enumeration]  Optional. The type of numbers to be counted. Can be one of the
      following: number paragraph, number listnum, or number all numbers. The default value is
      number all numbers.

  [level  integer]  Optional. A number that corresponds to the numbering level you want to count.
      If this argument is omitted, all levels are counted.

Remarks
Bulleted items are counted when either number paragraph or number all numbers (the default) is
specified for number type.

There are two types of numbers: preset numbers (number paragraph), which you can add to
paragraphs by selecting a template in the Bullets and Numbering dialog box; and LISTNUM fields
(number listnum), which allow you to add more than one number per paragraph.

Example
This example formats the current selection as a list, using the second numbered list template. The
example then counts the numbered and bulleted items and LISTNUM fields in the active document
and displays the result in a message box.
apply list format template list format of text object of selection ¬
        list template (list template 2 of (get list gallery number gallery))
display dialog (count numbered items active document)
This example counts the number of first-level numbered or bulleted items in the active document.
count numbered items list format of text object of active document level 1
This example counts the number of LISTNUM fields in the variable myRange. The result is displayed in a message box.

set myDoc to active document
set myRange to create range myDoc start (start of content of text object of paragraph 12 of myDoc) end (end of content of text object of paragraph 20 of myDoc)
set numfields to count numbered items list format of myRange number type number listnum
display dialog numfields

Command: create data source

Creates a Word document that uses a table to store data for a data merge. The new data source is attached to the specified document, which becomes a main document if it's not one already.

Syntax

create data source  data merge  Required. An expression that returns a data merge object.

[ name  Unicode text]  Optional. The path and file name for the new data source.

[password document  Unicode text]  Optional. The password required to open the new data source.

[write password  Unicode text]  Optional. The password required to save changes to the data source.

[header record  Unicode text]  Optional. Field names for the header record. If this argument is omitted, the standard header record is used: “Title, FirstName, LastName, JobTitle, Company, Address1, Address2, City, State, PostalCode, Country, HomePhone, WorkPhone.”

[MS Query  Boolean]  Optional. True to launch Microsoft Query, if it's installed. The name, password document, and header record arguments are ignored.


[SQL statement1  Unicode text]  Optional. If the query string is longer than 255 characters, SQL statement specifies the first portion of the string, and SQL statement1 specifies the second portion.

[connection  Unicode text]  Optional. When retrieving data through ODBC, the connection string that specifies the database within which the query specified by SQL statement will be performed.

[link to source  Boolean]  Optional. True to perform the query specified by connection and SQL statement each time the main document is opened.
Example

This example creates a new data source document named "Data.doc" and attaches the data source to the active document. The new data source includes a five-column table that has the field names specified by the header record argument.

```
create data source data merge of active document ¬
    name "Macintosh HD:Users:Shared:Data.doc" ¬
    header record "Name, Address, City, State, Zip"
```

Command: create header source

Creates a Word document that stores a header record that's used in place of the data source header record in a data merge. This method attaches the new header source to the specified document, which becomes a main document if it's not one already.

**Note** The new header source uses a table to arrange data merge field names.

**Syntax**

```
create header source  data merge  Required. An expression that returns a data merge object.
    name  Unicode text  Required. The path and file name for the new header source.
    [password document  Unicode text] Optional. The password required to open the new header source.
    [write password  Unicode text] Optional. The password required to save changes to the header source.
    [header record  Unicode text] Optional. A string that specifies the field names for the header record. If this argument is omitted, the standard header record is used: "Title, FirstName, LastName, JobTitle, Company, Address1, Address2, City, State, PostalCode, Country, HomePhone, WorkPhone."
```

Example

This example creates a header source with five field names and attaches the new header source named "Header.doc" to the active document.

```
create header source data merge of active document name "Header.doc" ¬
    header record "Name, Address, City, State, Zip"
```

This example creates a header source for the document named "Main.doc" (with the standard header record) and opens the data source named "Data.doc."

```
create header source data merge of document "Main.doc" name "Fields.doc"
open data source data merge of document "Main.doc" ¬
    name "Macintosh HD:Users:Shared:Data.doc"
```
Command: create letter content

Creates and returns a letter content object based on the specified letter elements.

**Syntax**

```plaintext
create letter content  
  document  Required. An expression that returns a document object.
  date format  Unicode text  Required. The date for the letter.
  include header footer  Boolean  Required. True to include the header and footer from the page design template.
  page design  Unicode text  Required. The name of the template attached to the document.
  letter style  enumeration  Required. The document layout. Can be one of the following: full block, modified block, or semi block.
  letterhead  Boolean  Required. True to reserve space for a preprinted letterhead.
  letterhead location  enumeration  Required. The location of the preprinted letterhead. Can be one of the following: letter bottom, letter left, letter right, or letter top.
  letterhead size  small real  Required. The amount of space (in points) to be reserved for a preprinted letterhead.
  recipient name  Unicode text  Required. The name of the person who will be receiving the letter.
  recipient address  Unicode text  Required. The mailing address of the person who will be receiving the letter.
  salutation  Unicode text  Required. The salutation text for the letter.
  salutation type  enumeration  Required. The salutation type for the letter. Can be one of the following: salutation business, salutation formal, salutation informal, or salutation other.
  recipient reference  Unicode text  Required. The reference line text for the letter (for example, "In reply to:").
  mailing instructions  Unicode text  Required. The mailing instruction text for the letter (for example, "Certified Mail").
  attention line  Unicode text  Required. The attention line text for the letter (for example, "Attention:").
  subject  Unicode text  Required. The subject text for the specified letter.
  cc list  Unicode text  Required. The names of the carbon copy (CC) recipients for the letter.
  return address  Unicode text  Required. The text of the return mailing address for the letter.
  sender name  Unicode text  Required. The name of the person sending the letter.
  closing  Unicode text  Required. The closing text for the letter.
  sender company  Unicode text  Required. The company name of the person creating the letter.
  sender job title  Unicode text  Required. The job title of the person creating the letter.
  sender initials  Unicode text  Required. The initials of the person creating the letter.
  enclosure count  integer  Required. The number of enclosures for the letter.
```
Example

The following example uses the CreateLetterContent command to create a new letter content object in the active document and then uses this object with the RunLetterWizard command.

```plaintext
set myLetter to create letter content active document ¬
    date format "April 14, 1912" ¬
    include header footer false ¬
    page design "" ¬
    letter style full block ¬
    letterhead true ¬
    letterhead location letter top ¬
    letterhead size 108.0 ¬
    recipient name "Terry Earls"
    recipient address "432 SE Main St" & return & "Bellevue, WA 98004" ¬
    salutation "Dear Terry,," ¬
    salutation type salutation informal ¬
    recipient reference "" ¬
    mailing instructions "" ¬
    attention line "" ¬
    subject "End of Year Report" ¬
    cc list "" ¬
    return address "" ¬
    sender name "" ¬
    closing "Sincerely yours,"
    sender company ""
    sender job title "" ¬
    sender initials "" ¬
    enclosure count 0
run letter wizard active document letter content myLetter
```

Command: create new document for hyperlink

Creates a new document linked to the specified hyperlink.

Syntax

`create new document for hyperlink` hyperlink  Required. An expression that returns a hyperlink object.

- `file name`  Unicode text  Required. The file name of the specified document.
- `edit now`  Boolean  Required. `true` to have the specified document open immediately in its associated editing environment. The default value is `true`.  
overwrite  Boolean  Required. True to overwrite any existing file of the same name in the same folder. False if any existing file of the same name is preserved and the file name argument specifies a new file name. The default value is false.

Example
This example creates a new document based on the new hyperlink in the first document and then loads the new document into Word for editing. The document is named "Link.doc" and it overwrites any file of the same name in the "Macintosh HD:Users:Shared" folder.

set objHyper to make new hyperlink object at active document with properties -
    {text object: text object of selection, ~
        hyperlink address:"Macintosh HD:Users:Shared:Link.doc"}
create new document for hyperlink objHyper file name ~
    "Macintosh HD:Users:Shared:Link.doc" with edit now and overwrite

Command: create new mailing label document

Creates a new label document using either the default label options or ones that you specify.
Returns a document object.

Syntax
create new mailing label document  mailing label  Required. An expression that returns a mailing label object.

[name  Unicode text] Optional. The mailing label name.
[address  Unicode text] Optional. The text for the mailing label.
[auto text  Unicode text] Optional. The name of the AutoText entry that includes the mailing label text.
[extract address  Boolean] Optional. True to use the address text marked by the user-defined bookmark named EnvelopeAddress instead of using the address argument.
[laser tray  enumeration] Optional. The laser printer tray. Can be one of the following: printer automatic sheet feed, printer default bin, printer envelope feed, printer form source, printer large capacity bin, printer large format bin, printer lower bin, printer manual envelope feed, printer manual feed, printer middle bin, printer only bin, printer paper cassette, printer small format bin, printer tractor feed, or printer upper bin.
[single label  Boolean] Optional. True if the text is placed within a single label on a sheet that contains multiple labels. This argument is used in conjunction with row and column. The default value is false.
[row  integer] Optional. Specifies the row in which to place the text when single label is set to true.
[column  integer] Optional. Specifies the column in which to place the text when single label is set to true.
Example
This example creates a new Avery 2160 minilabel document using a predefined address.
set addr to "Terry Earls" & return & "123 Skye St." & return & "Our Town, WA 98004"
create new mailing label document mailing label object name "2160 mini" ¬
    address addr without extract address
This example creates a new Avery 5664 shipping-label document using the selected text as the address.
set addr to content of text object of selection
create new mailing label document mailing label object name "5664" ¬
    address addr laser tray printer upper bin

Command: create range
Returns a text range object by using the specified starting and ending character positions.

Syntax
create range  document  Required. An expression that returns a document object.
    [start  integer]  Optional. The starting character position.
    [end  integer]  Optional. The ending character position.

Remarks
Character position values begin with 0 (zero) at the beginning of the document. All characters are counted, including nonprinting characters. Hidden characters are counted even if they're not displayed. If you don't specify starting and ending character positions for the create range command, the entire document is returned as a text range object.

Example
This example applies bold formatting to the first 10 characters in the active document.
set bold of (create range active document start 0 end 10) to true
This example creates a range that starts at the beginning of the active document and ends at the end of the selection.
set myRange to create range active document start 0 end (selection end ¬
    of selection)
This example sets the variable myRange to paragraphs three through six in the active document, and then it right aligns the paragraphs in the range.
set aDoc to active document
if (count of paragraphs of aDoc) ≥ 6 then
    set myRange to create range aDoc start (start of content of text object ¬
        of paragraph 3 of aDoc) end (end of content of text object of ¬
        paragraph 6 of aDoc)
    set alignment of paragraph format of myRange to align paragraph right
end if
Command: create textbox

Adds a default-size text box around the selection. If the selection is an insertion point, this method changes the pointer to a cross-hair pointer so that the user can draw a text box.

Syntax

create text box selection Required. An expression that returns a selection object.

Remarks

Using this method is equivalent to clicking the Text Box button on the Drawing toolbar. A text box is a rectangle with an associated text frame.

Example

This example adds a text box around the selection and then changes the line style of the text box.

if selection type of selection is selection normal then
    create textbox selection
    set dash style of line format of text box 1 of active document to ~
        line dash style dash dot
end if

Command: cut object

Removes the specified object from the document and places it on the Clipboard.

Syntax

cut object field/frame/form field/data merge field/selection object/page number Required. An expression that returns a field, frame, form field, data merge field, page number, or selection object.

Example

This example cuts the first field in the active document and pastes the field at the insertion point.

if (count of fields of active document) ≥ 1 then
    cut object field 1 of active document
    collapse range text object of selection direction collapse end
    paste object selection
end if

This example cuts the first word in the first paragraph and pastes the word at the end of the paragraph.

cut object word 1 of paragraph 1 of active document
home key selection move unit a story
set selection start of selection to ~
    (end of content of text object of paragraph 1 of active document) -1
paste object selection
This example cuts the contents of the selection and pastes them into a new document.

if selection type of selection is selection normal then
    cut object selection
    make new document
    paste object selection
end if

Command: data form
Displays the Data Form dialog box, in which you can add, delete, or modify data records.

Note You can use this command with a data merge main document, a data merge data source, or any document that contains data delimited by table cells or separator characters.

Syntax

data form  document  Required. An expression that returns a document object.

Example
This example displays the Data Form dialog box if the active document is a mail merge document.

if state of data merge of active document is not normal document then
    data form active document
end if

This example creates a table in a new document and then displays the Data Form dialog box.

set aDoc to make new document
make new table at aDoc with properties {number of rows:2, number of columns:2}
set content of text object of cell 1 of row 1 of table 1 of aDoc to "Name"
set content of text object of cell 2 of row 1 of table 1 of aDoc to "Age"
data form aDoc

Command: disable
Removes the specified key combination if it's currently assigned to a command. After you use this command, the key combination has no effect. Using this command is equivalent to clicking the Remove button in the Customize Keyboard dialog box (Tools menu).

Note To reset a built-in command to its default key assignment, use the clear command with a key binding object. You don't need to remove or rebind a key binding object before adding it elsewhere.

Syntax

disable  key binding  Required. An expression that returns a key binding object.
Example

This example removes the CTRL+SHIFT+B key assignment. This key combination is assigned to the **Bold** command by default.

```plaintext
set customization context to normal template
disable (find key key code (build key code key1 control_key key2 shift_key ¬
key3 b_key))
```

This example assigns the CTRL+SHIFT+O key combination to the **Organizer** command. The example then uses the `disable` command to remove the CTRL+SHIFT+O key combination and then displays a message.

```plaintext
set customization context to normal template
make new key binding with properties {key code:(build key code key1 ¬
control_key key2 shift_key key3 o_key), key category:key ¬
category command, command:"Organizer"}
display dialog (get command of (find key key code (build key code key1 ¬
control_key key2 shift_key key3 o_key)) & " is assigned to ¬
CTRL+SHIFT+O"
disable (find key key code (build key code key1 control_key key2 shift_key ¬
key3 o_key))
if command of (find key key code (build key code key1 control_key ¬
key2 shift_key key3 o_key)) is "" then display dialog "Nothing is ¬
assigned to CTRL+SHIFT+O"
```

This example removes all key assignments for the global macro named "Macro1."

```plaintext
set customization context to normal template
repeat with aKey in (get keys bound to key category key category macro ¬
command "Macro1")
    disable aKey
end repeat
```

**Command: display Word dialog**

Displays the specified built-in Word dialog box until either the user closes it or the specified amount of time has passed. Returns an integer that indicates which button was clicked to close the dialog box.

<table>
<thead>
<tr>
<th>Return value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-2</td>
<td>The <strong>Close</strong> button.</td>
</tr>
<tr>
<td>-1</td>
<td>The <strong>OK</strong> button.</td>
</tr>
<tr>
<td>0 (zero)</td>
<td>The <strong>Cancel</strong> button.</td>
</tr>
<tr>
<td>&gt; 0 (zero)</td>
<td>A command button: 1 is the first button, 2 is the second button, and so on.</td>
</tr>
</tbody>
</table>
Note Any actions initiated or settings specified while a dialog box is displayed using this command aren't carried out. To display a dialog box and carry out actions or apply settings, use the show command.

Syntax

display Word dialog  dialog  Required. An expression that returns a dialog object.

[time out  integer] Optional. The amount of time that Word will wait before closing the dialog box automatically. One unit is approximately 0.001 second. Concurrent system activity may increase the effective time value. If this argument is omitted, the dialog box is closed when the user closes it.

Example
This example displays the About dialog box.
set aDialog to get dialog dialog help about
display Word dialog aDialog
This example displays the Zoom dialog box for approximately nine seconds.
display Word dialog (get dialog dialog view zoom) time out 9000

Command: do Word repeat
Repeats the most recent editing action one or more times. Returns true if the commands were repeated successfully.

Note Using this command is equivalent to using the Repeat command on the Edit menu.

Syntax
do Word repeat

[times  integer] Optional. The number of times you want to repeat the last command.

Example
This example inserts the text 'Hello' followed by two paragraphs (the second typing action is repeated once).
type text selection text "Hello"
type paragraph selection
do Word repeat

Command: edit data source
Opens or switches to the mail merge data source.

Syntax
edit data source  data merge  Required. An expression that returns a data merge object.

Remarks
If the data source is a Word document, this command opens the data source (or activates the data source if it's already open).
**Example**
This example opens or activates the data source attached to the document named "Sales.doc."

`edit data source data merge of document "Sales.doc"`

**Command: edit header source**
Opens the header source attached to a data merge main document, or activates the header source if it's already open.

**Note**  If the data merge main document doesn't have a header source, this method causes an error.

**Syntax**
`edit header source`  `data merge`  Required. An expression that returns a `data merge` object.

**Example**
This example attaches a header source to the active document and then opens the header source.

`set main document type of data merge of active document to document type ¬ form letters`
`open header source data merge of active document ¬ name "Macintosh HD:Users:Shared:Header.doc"`
`edit header source data merge of active document`

This example opens the header source if the active document has an associated header file attached to it.

`set DM to data merge of active document`
`if (state of DM is main and source and header) or (state of DM is ¬ main and header) then`
`edit header source DM`
`end if`

**Command: edit main document**
Activates the data merge main document associated with the specified header source or data source document.

**Note**  If the main document isn't open, an error occurs. If the main document isn't currently open, use the `open` command.

**Syntax**
`edit main document`  `data merge`  Required. An expression that returns a `data merge` object.
Example

This example attempts to activate the main document associated with the active data source document. If the main document isn't open, the Open dialog box is displayed, with a message in the status bar.

try
    edit main document data merge of document "data.doc"
on error
    set status bar to "Main document is not open"
    show (get dialog dialog file open)
end try

Command: edit type

Sets options for the specified text form field.

Syntax

edit type  text input  Required. An expression that returns a text input object.

form field type  enumeration  Required. The text box type. Can be one of the following:
  calculation text, current date text, current time text, date text, number text, or regular text.

[default type  Unicode text]  Optional. The default text that appears in the text box.

[type format  Unicode text]  Optional. The formatting string used to format the text, number, or date (for example, "0.00," "Title Case," or "M/d/yy"). For more examples of formats, see the list of formats for the specified text form field type in the Text Form Field Options dialog box.

Example

This example adds a text form field named "Date" at the beginning of the active document. The edit type command is used to set the type to date text and set the date format to "M/d/yy."

set ffield to make new form field at active document with properties ¬
    {text object:(create range active document start 0 end 0), form ¬
     field type:field form text input}
set name of ffield to "Date"
edit type (text input of ffield) form field type date text ¬
    type format "M/d/yy"
Command: enable

Formats the first character in the specified paragraph as a dropped capital letter.

Syntax

enable drop cap  Required. An expression that returns a drop cap object.

Example

This example formats the first paragraph in the selection to begin with a dropped capital letter.

enable drop cap of paragraph 1 of selection
set lines to drop of drop cap of paragraph 1 of selection to 2
set font name of drop cap of paragraph 1 of selection to "Arial"

Command: end key

Moves or extends the selection to the end of the specified unit. This command returns a text range that represents the new text range of the selection, or returns a missing value if the move was unsuccessful.

Note  This command corresponds to the functionality of the END key.

Syntax

end key  selection object  Required. An expression that returns a selection object.

[move  enumeration]  Optional. The unit by which the selection is to be moved or extended. Can be one of the following: unit a story, unit a column, unit a line, unit a screen, unit a section, or unit a row. The default value is unit a line.

[extend  enumeration]  Optional. Specifies the way the selection is moved. Can be one of the following: by moving or by selecting. If the value of this argument is by moving, the selection is collapsed to an insertion point and moved to the end of the specified unit. If the value of this argument is by selecting, the end of the selection is extended to the end of the specified unit. The default value is by moving.

Example

This example moves the selection to the end of the current line and assigns the new text range to the pos variable.

set pos to end key selection move unit a line extend by moving

This example moves the selection to the beginning of the current table column and then extends the selection to the end of the column.

if (get selection information selection information type with in table) ¬
    is "True" then
    home key selection move unit a column extend by moving
end if

This example moves the selection to the end of the current story. If the selection is in the main text story, the example moves the selection to the end of the document.

end key selection move unit a story extend by moving
**Command: endnote convert**

Converts endnotes to footnotes, or vice versa.

**Syntax**

```plaintext
endnote convert  endnote options  Required. An expression that returns an endnote options object.
```

**Example**

This example converts all endnotes in the active document to footnotes.

```plaintext
set myEndnotes to (get endnotes of active document)
if (count of myEndnotes) > 0 then endnote convert endnote options ¬
   of active document
```

**Command: escape key**

Cancels a mode such as Extend or Column Select (equivalent to pressing the ESC key).

**Syntax**

```plaintext
escape key  selection object  Required. An expression that returns a selection object.
```

**Example**

This example turns on and then cancels Extend mode.

```plaintext
set extend mode of selection to true
escape key selection
```

**Command: execute data merge**

Performs the specified data merge operation.

**Syntax**

```plaintext
execute data merge  data merge  An expression that returns a data merge object.

   [pause  Boolean]  Optional. True to have Word pause and display a troubleshooting dialog box if
   a data merge error is found. False to report errors in a new document.
```

**Example**

This example executes a data merge if the active document is a main document with an attached
data source.

```plaintext
set myMerge to data merge of active document
if state of myMerge is main and data source then execute data merge myMerge
```
Command: execute dialog

Applies the current settings of a Word dialog box.

Syntax

execute dialog dialog Required. An expression that returns a dialog object.

Example

The following example executes the current settings on the Line and Page Breaks tab in the Paragraph dialog box.

set myDialog to get dialog dialog format paragraph
set default dialog tab of myDialog to dialog format paragraph tab text flow
execute dialog myDialog

Command: execute find

Runs the specified find operation. Returns true if the find operation is successful.

Syntax

execute find find Required. An expression that returns a find object.

[find text Unicode text] Optional. The text to be searched for. Use an empty string (""") to search for formatting only. You can search for special characters by specifying appropriate character codes. For example, "^p" corresponds to a paragraph mark and "^t" corresponds to a tab character.

If match wildcards is true, you can specify wildcard characters and other advanced search criteria. For example, "(ing)" finds any word that ends in "ing."

To search for a symbol character, type a caret (^), a zero (0), and then the character code for the symbol.

[match case Boolean] Optional. True to specify that the find text be case sensitive. Corresponds to the Match case check box in the Find and Replace dialog box (Edit menu).

[match whole word Boolean] Optional. True to have the find operation locate only entire words, not text that's part of a larger word. Corresponds to the Find whole words only check box in the Find and Replace dialog box.

[match wildcards Boolean] Optional. True to have the find text be a special search operator. Corresponds to the Use wildcards check box in the Find and Replace dialog box.

[match sounds like Boolean] Optional. True to have the find operation locate words that sound similar to the find text argument. Corresponds to the Sounds like check box in the Find and Replace dialog box.

[match all word forms Boolean] Optional. True to have the find operation locate all forms of the find text argument (for example, "sit" locates "sitting" and "sat"). Corresponds to the Find all word forms check box in the Find and Replace dialog box.
[match forward  Boolean]  Optional. True to search forward (toward the end of the document).

[wrap find enumeration]  Optional. Controls what happens if the search begins at a point other than the beginning of the document and the end of the document is reached (or vice versa if match forward is set to false). This argument also controls what happens if there’s a selection or text range and the search text isn’t found in the selection or range. Can be one of the following:

<table>
<thead>
<tr>
<th>Constant</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>find ask</td>
<td>After searching the selection or range, Word displays a message asking whether to search the remainder of the document.</td>
</tr>
<tr>
<td>find continue</td>
<td>The find operation continues if the beginning or end of the search range is reached.</td>
</tr>
<tr>
<td>find stop</td>
<td>The find operation ends if the beginning or end of the search range is reached.</td>
</tr>
</tbody>
</table>

[find format Boolean]  Optional. True to have the find operation locate formatting in addition to or instead of the find text.

[replace with Unicode text]  Optional. The replacement text. To delete the text specified by the find text argument, use an empty string ("""). You specify special characters and advanced search criteria just as you do for the find text argument. To specify a graphic object or other non-text item as the replacement, put the item on the Clipboard and specify "^c" for replace with.

[replace enumeration]  Optional. Specifies how many replacements are to be made: one, all, or none. Can be one of the following: replace all, replace none, or replace one.

Example
This example finds and selects the next occurrence of the word "library."
set myFind to find object of selection
clear formatting myFind
set match whole word of myFind to true
set match case of myFind to false
execute find myFind find text "library"

This example finds all occurrences of the word "hi" in the active document and replaces each occurrence with "hello."
set myRange to text object of active document
execute find object of myRange find text "hi" replace with "Hello" ~
    replace replace all
Command: execute key binding

Runs the command associated with the specified key combination.

Syntax

```
execute key binding key binding
```

Required. An expression that returns a `key binding` object.

Example

This example assigns the CTRL+SHIFT+C key combination to the `FileClose` command and then executes the key combination (the document is closed).

```
set customization context to attached template of active document
make new key binding with properties ¬
  { key code:(build key code key1 control_key key2 shift_key ¬
    key3 c_key), key category:key category command, command:"FileClose"}
execute key binding (find key key code(build key code key1 control_key ¬
  key2 shift_key key3 c_key))
```

Command: expand

Expands the specified range or selection.

Syntax

```
expand selection object
```

Required. An expression that returns a `selection` object.

[by enumeration] Optional. The unit by which to expand the range. Can be one of the following: a character item, a word item, a sentence item, a paragraph item, a line item, a section item, a story item, a screen item, a cell item, a column item, a row item, a window item, a character formatting item, a paragraph formatting item, a item unit, or a table item.

Example

This example capitalizes the first character in the selection and then expands the selection to include the entire sentence.

```
set case of character 1 of selection to title sentence
expand selection by a sentence item
```

Command: expand outline

Expands the text under the selection or the specified range by one heading level.

Note If the document isn’t in outline or master document view, an error occurs.

Syntax

```
expand outline view
```

Required. An expression that returns a `view` object.

[text range] Optional `text range` object. The range of paragraphs to be expanded. If this argument is omitted, the entire selection is expanded.
Example
This example expands every heading in the document by one level.
set view type of view of active window to outline view
expand outline view of active window text range text object of active document
This example expands the active paragraph in the Document2 window.
activate window "Document2"
set view type of view of window "Document2" to outline view
expand outline view of window "Document2"

Command: extend
Turns Extend mode on and extends the selection to the next unit of text. The progression is as follows: word, sentence, paragraph, section, entire document. If character is specified, extends the selection through the next instance of the specified character.

Syntax
extend  selection object  Required. An expression that returns a selection object.
  [character  Unicode text]  Optional. The character that the selection is to be extended through.
    This argument is case sensitive.

Example
This example extends the selection through the next instance of a capital "R".
extend selection character "R"

Command: find key
Returns a key binding object that represents the specified custom key combination.

Syntax
find key
  key code  integer  Required. A key code returned by the build key code command.
  [key_code_2  enumeration]  Optional. A secondary key code returned from the build key code command.

Remarks
Use the build key code command to create the key code or key_code_2 argument.
Example

This example assigns the OPTION+F4 key combination to the Arial font and then displays the number of items in the key bindings list. The example then clears the key combinations (returns it to its default setting) and redispays the number of items in the key bindings list.

set customization context to normal template
make new key binding with properties ¬
   { key code:(build key code key1 key alt key2 f4_key), ¬
     key category:key category font, command:"Arial" }
display dialog (get (count of key bindings) as string) & " keys in the key ¬
    binding list"
clear (find key key code (build key code key1 key alt key2 f4_key))
display dialog (get (count of key bindings) as string) & " keys in the key ¬
    binding list"

This example assigns the CTRL+SHIFT+V key combination to the macro named "Macro1" in the active document. The example uses the find key command to return a key binding object so that Word can retrieve and display the command name.

set customization context to active document
make new key binding with properties ¬
   { key code:(build key code key1 control_key key2 shift_key ¬
       key3 v_key), key category:key category macro, command:"Macro1" }
display dialog (get command of (find key key code (build key code key1 ¬
       control_key key2 shift_key key3 v_key)))

This example determines whether the CTRL+SHIFT+A key combination is part of the key bindings list.

set customization context to normal template
set myKey to find key key code (build key code key1 control_key key2 ¬
   shift_key key3 a_key)
if command of myKey is "" then display dialog "Key is not in the key ¬
   bindings list"

Command: find record

Searches the contents of the specified mail merge data source for text in a particular field. Returns True if the search text is found.

Note  Corresponds to the Find Record button on the Data Merge Manager.

Syntax

find record  data merge data source  Required. An expression that returns a data merge data source object.
find text    Required. The text to be looked for.
field name  Required. The name of the field to be searched.
Example
This example displays a merge document for the first data record in which the FirstName field contains "Joe." If the data record is found, the number of the record is stored in the numRecord variable.

set view data merge field codes of data merge of active document to false
set myDMData to data source of data merge of active document
if (find record myDMData find text "Joe" field name "FirstName") is true then set numRecord to active record of myDMData

Command: fit to pages
Decreases the font size of text just enough so that the document will fit on one fewer pages. An error occurs if Word is unable to reduce the page count by one.

Syntax
fit to pages document Required. An expression that returns a document object.

Example
This example attempts to reduce the page count of the active document by one page.
try
  fit to pages active document
on error
  display dialog "Fit to pages failed"
end try
This example attempts to reduce the page count of each open document by one page.
repeat with doc in (get documents)
   fit to pages doc
end repeat

Command: follow
Displays a cached document associated with the specified hyperlink object, if it's already been downloaded. Otherwise, this method resolves the hyperlink, downloads the target document, and displays the document in the appropriate application.

Note If the hyperlink uses the file protocol, this method opens the document instead of downloading it.

Syntax
follow hyperlink object Required. An expression that returns a hyperlink object.

new window Boolean Optional. True to display the target document in a new window. The default value is false.

extra info Unicode text Optional. A string or byte array that specifies additional information for HTTP to use to resolve the hyperlink. For example, you can use extra info to specify the coordinates of an image map. Use the extra info required property of the hyperlink object to determine whether extra information is required.
Example
This example follows the first hyperlink in Home.doc.
follow (hyperlink object 1 of document "Home.doc")
This example inserts a hyperlink to www.msn.com and then follows the hyperlink.
collapse range text object of selection direction collapse end
set myLink to make new hyperlink object at active document with properties ¬
    {text to display:"MSN", hyperlink address:"http://www.msn.com", ¬
      text object:text object of selection}
follow myLink

Command: follow hyperlink
Displays a cached document if it's already been downloaded. Otherwise, this method resolves the hyperlink, downloads the target document, and displays the document in the appropriate application.

Note If the hyperlink uses the file protocol, this method opens the document instead of downloading it.

Syntax
follow hyperlink  document  Required. An expression that returns a document object.
address  Unicode text  Required. The address of the target document.
[sub address  Unicode text]  Optional. The location within the target document. The default value is an empty string.
[new window  Boolean]  Optional. True to display the target location in a new window. The default value is false.
[add history  Boolean]  Optional. This argument is reserved for future use.
[extra info  Unicode text]  Optional. A string or a byte array that specifies additional information for HTTP to use to resolve the hyperlink. For example, you can use extra info to specify the coordinates of an image map, the contents of a form, or a file name. To determine whether extra information is required, use the extra info required property.

Example
This example follows the specified URL address and displays the Microsoft home page in a new window.
follow hyperlink active document address "http://www.microsoft.com/" ~
    with new window and add history
This example displays the HTML document named "Default.htm."
follow hyperlink active document ¬
    address "Macintosh HD:Users:Shared:Default.htm"
Command: **footnote convert**

Converts endnotes to footnotes, or vice versa.

**Syntax**

`footnote convert` **footnote options**  Required. An expression that returns a **footnote options** object.

**Example**

This example converts the footnotes in the selection to endnotes.

```plaintext
if (get count of footnotes of selection) > 0 then footnote convert ~
   footnote options of selection
```

Command: **get active writing style**

Returns the writing style for a specified language in the specified document.

**Syntax**

`get active writing style` **document**  Required. An expression that returns a **document** object.

  **language id**  enumeration  Required. The language you want to set the writing style for in the specified document. Can be one of the following:

<table>
<thead>
<tr>
<th>Language</th>
<th>Language</th>
<th>Language</th>
<th>Language</th>
</tr>
</thead>
<tbody>
<tr>
<td>afrikaans</td>
<td>french</td>
<td>russian</td>
<td></td>
</tr>
<tr>
<td>arabic</td>
<td>french canadian</td>
<td>serbian cyrillic</td>
<td></td>
</tr>
<tr>
<td>basque</td>
<td>german</td>
<td>serbian latin</td>
<td></td>
</tr>
<tr>
<td>belgian dutch</td>
<td>greek</td>
<td>sesotho</td>
<td></td>
</tr>
<tr>
<td>belgian french</td>
<td>hebrew</td>
<td>simplified chinese</td>
<td></td>
</tr>
<tr>
<td>brazilian portuguese</td>
<td>hungarian</td>
<td>slovak</td>
<td></td>
</tr>
<tr>
<td>bulgarian</td>
<td>italian</td>
<td>slovenian</td>
<td></td>
</tr>
<tr>
<td>byelorussian</td>
<td>icelandic</td>
<td>spanish</td>
<td></td>
</tr>
<tr>
<td>catalan</td>
<td>japanese</td>
<td>spanish modern sort</td>
<td></td>
</tr>
<tr>
<td>croatian</td>
<td>korean</td>
<td>swedish</td>
<td></td>
</tr>
<tr>
<td>czech</td>
<td>language none</td>
<td>swiss french</td>
<td></td>
</tr>
<tr>
<td>danish</td>
<td>latvian</td>
<td>swiss german</td>
<td></td>
</tr>
<tr>
<td>dutch</td>
<td>macedonian</td>
<td>swiss italian</td>
<td></td>
</tr>
<tr>
<td>english aus</td>
<td>malaysian</td>
<td>traditional chinese</td>
<td></td>
</tr>
<tr>
<td>english canadian</td>
<td>mexicanSpanish</td>
<td>tsonga</td>
<td></td>
</tr>
<tr>
<td>english new zealand</td>
<td>language no proofing</td>
<td>tswana</td>
<td></td>
</tr>
<tr>
<td>english south africa</td>
<td>norwegian bokmøl</td>
<td>turkish</td>
<td></td>
</tr>
<tr>
<td>english uk</td>
<td>norwegian nynorsk</td>
<td>ukrainian</td>
<td></td>
</tr>
<tr>
<td>english us</td>
<td>polish</td>
<td>venda</td>
<td></td>
</tr>
<tr>
<td>estonian</td>
<td>portuguese</td>
<td>xhosa</td>
<td></td>
</tr>
<tr>
<td>farsi</td>
<td>romanian</td>
<td>zulu</td>
<td></td>
</tr>
</tbody>
</table>

**Example**

This example returns the writing style for the language of the selection.

```plaintext
set myLang to language id of selection
display dialog (get active writing style active document language id myLang)
```
Command: get border

Returns the specified border object.

Syntax

get border  font/frame/selection object  Required. The object to get the border from.

which border  enumeration  Required. The border to be retrieved. Can be one of the following:
border top, border left, border bottom, border right, border horizontal, border vertical,
border diagonal down, or border diagonal up.

Example

This example adds a border around all of the pages in the current section.

set art style of (get border section 1 of selection which border border top) ¬
to art basic black dots
set art style of (get border section 1 of selection which border border bottom) ¬
to art basic black dots
set art style of (get border section 1 of selection which border border left) ¬
to art basic black dots
set art style of (get border section 1 of selection which border border right) ¬
to art basic black dots
set art width of (get border section 1 of selection) to 6

Command: get cross reference items

Returns a list of items that can be cross-referenced based on the specified cross-reference type. The
array corresponds to the items listed in the For which box in the Cross-reference dialog box (Insert
menu).

Note  An item returned by this method can be used as the reference item argument for the insert
cross reference command.

Syntax

get cross reference items  document  Required. An expression that returns a document object.

reference type  enumeration  Required. The type of item you want to insert a cross-reference to.
Can be one of the following: reference type bookmark, reference type footnote, reference
type endnote, reference type heading, or reference type numbered item.

Example

This example displays the name of the first bookmark in the active document that can be cross-
referenced.

if (get count of bookmarks of active document) > 1 then
  set myBookmarks to get cross reference items active document ¬
    reference type reference type bookmark
  display dialog item 1 of myBookmarks
end if
Command: get default file path
Returns the default folders for items such as documents, templates, and graphics.

Note  The path doesn't include a trailing character; for example, Macintosh HD:Applications:Microsoft Office 2004. To add the character that separates folders and volumes, use the path separator property.

Syntax
get default file path

file path type  enumeration  Required. The default folder to return. Can be one of the following: auto recover path, border art path, current folder path, documents path, graphics filters path, office path, pictures path, program path, proofing tools path, startup path, style gallery path, temp file path, text converters path, tools path, trash path, tutorial path, type libraries path, user options path, user templates path, or workgroup templates path.

Example
This example returns the current default path for user templates (corresponds to the default path setting in the File Locations pane of the Preferences dialog box).
get default file path file path type user templates path

Command: get default theme for
Returns the name of the default theme plus the theme formatting options Word uses for new documents, e-mail messages, or Web pages.

Syntax
get default theme for

theme type  enumeration  Required. The type of new document for which you want to retrieve the default theme name. Can be one of the following: for a document, for a email message, or for a web page.

Example
This example displays the name of the theme Word uses for new Web pages.
display dialog (get default theme for theme type for a web page)

Command: get dialog
Returns a dialog object that represents the specified built-in dialog box in Word.

Syntax
get dialog  enumeration  Required. The dialog type constant for the dialog box you want to return.

The dialog type constants are formed from the prefix "dialog" followed by the name of the menu and the dialog box. For example, the constant for the Page Setup dialog box is dialog file page setup, and the constant for the Project Gallery dialog box is dialog file new.

Example
The following example displays and carries out the actions taken in the built-in Open dialog box (File menu).
show (get dialog dialog file open)
Command: get document compatibility

Returns the current state of the specified compatibility item for the specified document. True if the compatibility option specified by the compatibility item argument is enabled. Compatibility options affect how a document is displayed in Word. These options are listed in the Compatibility pane of the Preferences dialog box (Word menu).

Syntax

get document compatibility  document  Required. An expression that returns a document object.
compatibility item  enumeration  Required. Can be one of the following:

<table>
<thead>
<tr>
<th>Constant</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>align tables row by row</td>
<td>True if Word aligns table rows independently.</td>
</tr>
<tr>
<td>Word95 auto space</td>
<td>True if Word auto spaces like Word 95.</td>
</tr>
<tr>
<td>convert data merge escapes</td>
<td>True if Word correctly interprets characters preceded by backslashes () in Word version 2.x mail-merge data sources. For example, &quot; is interpreted as &quot;.</td>
</tr>
<tr>
<td>do not adjust line height in table</td>
<td>True if Word doesn't adjust line height to grid height in tables.</td>
</tr>
<tr>
<td>do not balance SBCS and DBCS characters</td>
<td>True if Word doesn't balance the width of single-byte and double-byte characters.</td>
</tr>
<tr>
<td>do not underline trailing spaces</td>
<td>True if Word doesn't draw underlines on trailing spaces.</td>
</tr>
<tr>
<td>do not use html paragraph autospacing</td>
<td>True if Word doesn't use HTML paragraph auto spacing.</td>
</tr>
<tr>
<td>exact on top</td>
<td>True if Word places text at the top of the space allocated for the line when using exact line spacing.</td>
</tr>
<tr>
<td>expand shift return</td>
<td>True if Word expands character spaces on required line breaks (SHIFT+RETURN).</td>
</tr>
<tr>
<td>Word98 footnote layout</td>
<td>True if Word lays out footnotes like Word 6.x/95/97.</td>
</tr>
<tr>
<td>forget last tab alignment</td>
<td>True if Word forgets the last tab alignment when adding a new tab.</td>
</tr>
<tr>
<td>layout raw table width</td>
<td>True if Word lays out tables with their raw width.</td>
</tr>
<tr>
<td>layout table rows apart</td>
<td>True if Word allows table rows to lay out apart.</td>
</tr>
</tbody>
</table>

261
<table>
<thead>
<tr>
<th>Constant</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>leave backslash alone</td>
<td>True if Word doesn't convert backslash characters into yen signs.</td>
</tr>
<tr>
<td>Word6 line wrap</td>
<td>True if Word wraps lines the same way Word 6.0 wraps lines.</td>
</tr>
<tr>
<td>use larger small caps</td>
<td>True if Word applies small capital letter (“small caps”) formatting as in Word version 5.x for the Macintosh, which produces slightly larger small caps.</td>
</tr>
<tr>
<td>no column balance</td>
<td>True if Word doesn’t balance text columns above continuous section breaks.</td>
</tr>
<tr>
<td>no extra line spacing</td>
<td>True if Word handles line spacing as in WordPerfect version 5.x. This argument is available only in Word version 7.0.</td>
</tr>
<tr>
<td>no extra spacing between rows of text</td>
<td>True if Word displays lines of text without leading as in Word version 5.x for the Macintosh.</td>
</tr>
<tr>
<td>no space for underlines</td>
<td>True if Word doesn't add extra space for underlines. This argument is available only in Word version 7.0.</td>
</tr>
<tr>
<td>no space for raised or lowered characters</td>
<td>True if Word doesn't add extra line spacing for raised and lowered characters.</td>
</tr>
<tr>
<td>no tab hanging indent</td>
<td>True if Word doesn’t automatically add a tab stop to a paragraph formatted with a hanging indent.</td>
</tr>
<tr>
<td>original word table rules</td>
<td>True if Word combines table borders as in Word version 5.x for the Macintosh.</td>
</tr>
<tr>
<td>print body text before header</td>
<td>True if Word prints the main text layer before the header and footer layer (the reverse of the default order). This allows Word to process PostScript codes in the main text layer the same as in Word version 5.x for the Macintosh.</td>
</tr>
<tr>
<td>print colors black</td>
<td>True if Word prints colors as black on printers that don't support color.</td>
</tr>
<tr>
<td>Word96 shape layout</td>
<td>True if Word lays out AutoShapes like Word 97.</td>
</tr>
<tr>
<td>show breaks in frames</td>
<td>True if Word displays manual (“hard”) page or column breaks in any frames that contain them.</td>
</tr>
<tr>
<td>spacing in whole points</td>
<td>True if Word rounds character-spacing measurements up or down to the nearest whole number.</td>
</tr>
<tr>
<td>substitute font by size</td>
<td>True if Word substitutes fonts based on the font size in WordPerfect version 6.0 documents. This argument is available only in Word version 7.0.</td>
</tr>
<tr>
<td>Constant</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>suppress bottom spacing</td>
<td>True if Word removes extra line spacing at the bottom of the page.</td>
</tr>
<tr>
<td>suppress space before after page break</td>
<td>True if Word removes space before or after hard page breaks and column breaks.</td>
</tr>
<tr>
<td>suppress top spacing</td>
<td>True if Word removes extra line spacing at the top of the page.</td>
</tr>
<tr>
<td>suppress top spacing Mac Word5</td>
<td>True if Word handles extra line spacing at the top of the page the same as in Word version 5.x for the Macintosh.</td>
</tr>
<tr>
<td>swap borders facing pages</td>
<td>True if Word prints a left paragraph border (not a box) on the right side on odd-numbered pages if either the Different odd and even check box (Layout tab) or the Mirror margins check box (Margins tab) is selected in the Page Setup dialog box (File menu).</td>
</tr>
<tr>
<td>transparent metafiles</td>
<td>True if Word doesn't &quot;blank&quot; the area behind metafile pictures.</td>
</tr>
<tr>
<td>truncate font height</td>
<td>True if Word rounds the font size up or down as in WordPerfect version 6.x for Windows. This argument is available only in Word version 7.0.</td>
</tr>
<tr>
<td>use printer metrics</td>
<td>True if Word uses printer metrics to lay out the document.</td>
</tr>
<tr>
<td>use Word97 line breaking rules</td>
<td>True if Word uses Word 97 line-breaking rules for Asian text.</td>
</tr>
<tr>
<td>WordPerfect justification</td>
<td>True if Word lays out fully justified text the same as in WordPerfect version 6.x.</td>
</tr>
<tr>
<td>WordPerfect space width</td>
<td>True if Word sets the default width of spaces between words to be the same as in WordPerfect version 5.x for Windows and WordPerfect version 6.0 for DOS.</td>
</tr>
<tr>
<td>wrap trail spaces</td>
<td>True if Word wraps spaces at the end of lines to the next line.</td>
</tr>
<tr>
<td>Word6 border rules</td>
<td>True if Word handles left and right borders of text lines affected by frames the same as in Word version 6.x.</td>
</tr>
</tbody>
</table>

**Example**
This example returns whether the Suppress Space Before after a hard page or column break option in the Compatibility pane of the Preferences dialog box (Word menu) for the active document is set.

```plaintext
get document compatibility active document compatibility item suppress space before after page break
```
Command: get international information

Returns information about the current international settings.

Syntax

get international information  enumeration  Required. Specifies a single item to be returned. Can be one of the following:

<table>
<thead>
<tr>
<th>Constant</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>twenty four hour clock</td>
<td>Returns true if you're using 24-hour time; returns false if you're using 12-hour time.</td>
</tr>
<tr>
<td>currency code</td>
<td>Returns the currency symbol ($ in U.S. English).</td>
</tr>
<tr>
<td>date separator</td>
<td>Returns the date separator (/ in U.S. English).</td>
</tr>
<tr>
<td>decimal separator</td>
<td>Returns the decimal separator (. in U.S. English).</td>
</tr>
<tr>
<td>international am</td>
<td>Returns the string used to indicate morning hours (for example, 10 AM).</td>
</tr>
<tr>
<td>international pm</td>
<td>Returns the string used to indicate afternoon and evening hours (for example, 2 PM).</td>
</tr>
<tr>
<td>list separator</td>
<td>Returns the list separator (, in U.S. English).</td>
</tr>
<tr>
<td>product language ID</td>
<td>Returns the language version of Word.</td>
</tr>
<tr>
<td>thousands separator</td>
<td>Returns the thousands separator (, in U.S. English).</td>
</tr>
<tr>
<td>time separator</td>
<td>Returns the time separator (: in U.S. English).</td>
</tr>
</tbody>
</table>

Example

This example displays the currency format in the status bar.

set status bar to "Currency Format: " & get international information - currency code

Command: get keys bound to

Returns a list of key binding objects that represents all the key combinations assigned to the specified item.

Syntax

get keys bound to

key category  enumeration  Required. The category of the key combination. Can be one of the following: key category nil, key category disable, key category auto text, key category command, key category font, key category macro, key category style, key category prefix, or key category symbol.

command  Unicode text  Required. The name of the command.
Example

This example displays all the key combinations assigned to the **FileOpen** command in the template attached to the active document.

```plaintext
set customization context to attached template of active document
set myKey to (get keys bound to key category key category command ~
   command "FileClose")
repeat with kbs in myKey
   display dialog (get binding key string of kbs)
end repeat
```

This example removes all key assignments from Macro1 in the Normal template.

```plaintext
set customization context to normal template
repeat with aKey in get keys bound to key category key category macro ~
   command "Macro1"
   disable aKey
end repeat
```

**Command: get list gallery**

Returns a **list gallery** object that represents the three list template galleries (**Bulleted**, **Numbered**, and **Outline Numbered**). Each gallery corresponds to a tab in the **Bullets and Numbering** dialog box.

**Syntax**

get list gallery  enumeration   Required. The list gallery to return. Can be one of the following: **bullet gallery**, **number gallery**, or **outline number gallery**.

**Example**

This example sets the variable `mylsttmp` to the second list template on the **Outline Numbered** tab in the **Bullets and Numbering** dialog box. The example then applies that template to the first list in the active document.

```plaintext
set mylsttmp to list template 2 of (get list gallery outline number gallery)
apply list template list 1 of active document list template mylsttmp
```

**Command: get next field**

Selects the next field. If a field is found, this command returns a **field** object; if not, it returns **undefined**.

**Syntax**

get next field  selection object   Required. An expression that returns a **selection** object.
Example
This example updates the next field in the selection.
try
  update field (get next field selection)
on error
  display dialog "There is no next field"
end try

Command: get previous field
Selects the previous field. If a field is found, this command returns a field object; if not, it returns undefined.

Syntax
get previous field  selection object  Required. An expression that returns a selection object.

Example
This example updates the previous field (the field immediately preceding the selection).
try
  update field (get previous field selection)
on error
  display dialog "There is no previous field"
end try

Command: get private profile string
Returns a string in a settings file.

Syntax
get private profile string  system object  Required. An expression that returns a system object.

  file name  Unicode text  Required. The file name for the settings file. If no path is specified, the Users/<user name>/Library/Preferences/Microsoft folder is assumed.

  section  Unicode text  Required. The name of the section in the settings file that contains key.

  key  Unicode text  Required. The key whose setting you want to retrieve.

Remarks
You can write scripts that use a settings file to store and retrieve settings. For example, you can store the name of the active document when you quit Word so that it can be reopened automatically the next time you start Word.
Command: get profile string

Returns or sets a setting in the Word Settings (10) file.

Syntax

get profile string system object   Required. An expression that returns a system object.
section   Unicode text  Required. The name of the section in the settings file that contains the key.
key   Unicode text  Required. The value in the key specified by section (for example, "BackgroundPrint" in the Options key).

Command: get selection information

Returns information about the specified selection or range. This command returns a Unicode text string.

Syntax

get selection information selection object   Required. An expression that returns a selection object.
information type   enumeration   Required. The information to return. Can be one of the following:

- active end adjusted page number  Returns the number of the page that contains the active end of the specified selection or range. If you set a starting page number or make other manual adjustments, returns the adjusted page number (unlike active end page number).

- active end page number  Returns the number of the page that contains the active end of the specified selection or range, counting from the beginning of the document. Any manual adjustments to page numbering are disregarded (unlike active end adjusted page number).

- active end section number  Returns the number of the section that contains the active end of the specified selection or range.

- at end of row marker  Returns "True" if the specified selection or range is at the end-of-row mark in a table.

- info caps lock  Returns "True" if Caps Lock is in effect.

- end_of range column number  Returns the table column number that contains the end of the specified selection or range.

- end_of range row number  Returns the table row number that contains the end of the specified selection or range.

- first character column number  Returns the character position of the first character in the specified selection or range. If the selection or range is collapsed, the character number immediately to the right of the range or selection is returned (this is the same as the character column number displayed in the status bar after "Col").

- first character line number  Returns the line number of the first character in the selection. If the pagination property is false or the draft property is "True", returns -1.
frame is selected Returns "True" if the selection or range is an entire frame or text box.

header footer type Returns a value that indicates the type of header or footer that contains the specified selection or range, as shown in the following table.

<table>
<thead>
<tr>
<th>Value</th>
<th>Type of header or footer</th>
</tr>
</thead>
<tbody>
<tr>
<td>-1</td>
<td>None (the selection or range isn't in a header or footer)</td>
</tr>
<tr>
<td>0</td>
<td>Even page header</td>
</tr>
<tr>
<td>1</td>
<td>Odd page header (or the only header, if there aren't odd and even headers)</td>
</tr>
<tr>
<td>2</td>
<td>Even page footer</td>
</tr>
<tr>
<td>3</td>
<td>Odd page footer (or the only footer, if there aren't odd and even footers)</td>
</tr>
<tr>
<td>4</td>
<td>First page header</td>
</tr>
<tr>
<td>5</td>
<td>First page footer</td>
</tr>
</tbody>
</table>

horizontal position relative to page Returns the horizontal position of the specified selection or range; this is the distance from the left edge of the selection or range to the left edge of the page, in twips (20 twips = 1 point, 72 points = 1 inch). If the selection or range isn't within the screen area, returns -1.

horizontal position relative to text boundary Returns the horizontal position of the specified selection or range, relative to the left edge of the nearest text boundary enclosing it, in twips (20 twips = 1 point, 72 points = 1 inch). If the selection or range isn't within the screen area, returns -1.

in clipboard Returns "True" if the specified selection or range is on the Clipboard.
in comment pane Returns "True" if the specified selection or range is in a comment pane.
in endnote Returns "True" if the specified selection or range is in an endnote area in page layout view or in the endnote pane in normal view.
in footnote Returns "True" if the specified selection or range is in a footnote area in page layout view or in the footnote pane in normal view.
in footnote endnote pane Returns "True" if the specified selection or range is in the footnote or endnote pane in normal view or in a footnote or endnote area in page layout view. For more information, see the descriptions of in footnote and in endnote in the preceding paragraphs.
in header footer Returns "True" if the selection or range is in the header or footer pane or in a header or footer in page layout view.
in master document Returns "True" if the selection or range is in a master document (that is, a document that contains at least one subdocument).
- **in word mail**  Returns a value that indicates the WordMail location of the selection or range, as shown in the following table.

<table>
<thead>
<tr>
<th>Value</th>
<th>WordMail location</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 (zero)</td>
<td>The selection or range isn't in a WordMail message.</td>
</tr>
<tr>
<td>1</td>
<td>The selection or range is in a WordMail send note.</td>
</tr>
<tr>
<td>2</td>
<td>The selection or range is in a WordMail read note.</td>
</tr>
</tbody>
</table>

- **maximum number of columns**  Returns the greatest number of table columns within any row in the selection or range.

- **maximum number of rows**  Returns the greatest number of table rows within the table in the specified selection or range.

- **number of pages in document**  Returns the number of pages in the document associated with the selection or range.

- **info num lock**  Returns "True" if Num Lock is in effect.

- **over type**  Returns "True" if Overtype mode is in effect. The overtype property can be used to change the state of overtype mode.

- **reference of type**  Returns a value that indicates where the selection is in relation to a footnote, endnote, or comment reference, as shown in the following table.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-1</td>
<td>The selection or range includes but isn't limited to a footnote, endnote, or comment reference.</td>
</tr>
<tr>
<td>0 (zero)</td>
<td>The selection or range isn't before a footnote, endnote, or comment reference.</td>
</tr>
<tr>
<td>1</td>
<td>The selection or range is before a footnote reference.</td>
</tr>
<tr>
<td>2</td>
<td>The selection or range is before an endnote reference.</td>
</tr>
<tr>
<td>3</td>
<td>The selection or range is before a comment reference.</td>
</tr>
</tbody>
</table>

- **revision marking**  Returns "True" if change tracking is in effect.

- **selection mode**  Returns a value that indicates the current selection mode, as shown in the following table.

<table>
<thead>
<tr>
<th>Value</th>
<th>Selection mode</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 (zero)</td>
<td>Normal selection</td>
</tr>
<tr>
<td>1</td>
<td>Extended selection (&quot;EXT&quot; appears on the status bar)</td>
</tr>
<tr>
<td>2</td>
<td>Column selection. (&quot;COL&quot; appears on the status bar)</td>
</tr>
</tbody>
</table>
- **start of range column number**  Returns the table column number that contains the beginning of the selection or range.

- **start of range row number**  Returns the table row number that contains the beginning of the selection or range.

- **vertical position relative to page**  Returns the vertical position of the selection or range; this is the distance from the top edge of the selection to the top edge of the page, in twips (20 twips = 1 point, 72 points = 1 inch). If the selection isn't visible in the document window, returns -1.

- **vertical position relative to page boundary**  Returns the vertical position of the selection or range, relative to the top edge of the nearest text boundary enclosing it, in twips (20 twips = 1 point, 72 points = 1 inch). This is useful for determining the position of the insertion point within a frame or table cell. If the selection isn't visible, returns -1.

- **with in table**  Returns "True" if the selection is in a table.

- **zoom percentage**  Returns the current percentage of magnification as set by the percentage property.

**Example**

This example displays the current page number and the total number of pages in the active document.

```vba
display dialog "The selection is on page " & (get selection information ¬
selection information type active end page number) & " of page " & ¬
(get selection information information type ¬
number of pages in document)
```

**Command: get spelling suggestions**

Returns a record that represents the type of error and the list of words suggested as spelling replacements for a given word.

**Syntax**

```vba
get spelling suggestions

item to check    Unicode text    Required. The word whose spelling is to be checked.


[ignore uppercase    Boolean]    Optional. True to ignore words in all uppercase letters. If this argument is omitted, the current value of the ignore uppercase property of the Word options class is used.

[main dictionary    dictionary]    Optional. An expression that returns a dictionary object.
```
[suggestion mode  enumeration]  Optional. Specifies the way Word makes spelling suggestions. Can be one of the following. The default value is spelling word spell word.

<table>
<thead>
<tr>
<th>Constant</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>spelling word spell word</td>
<td>Word suggests correct spellings for the word or the first word in the specified range.</td>
</tr>
<tr>
<td>spelling word type wildcard</td>
<td>Word suggests replacements that match the search criteria for a word that contains the question mark (?) or asterisk (*) wildcard character. This constant is not supported in the current spell checker.</td>
</tr>
<tr>
<td>spelling word type anagram</td>
<td>Word suggests anagrams for the word. Word doesn't suggest anagrams from a custom dictionary. This constant is not supported in the current spell checker.</td>
</tr>
</tbody>
</table>

Note  This parameter may be ignored, depending on the dictionary file currently in use.

[custom dictionary2-custom dictionary10  dictionary]  Optional. An expression that returns a dictionary object. You can specify as many as nine additional dictionaries.

Remarks
If the word is spelled correctly, the type class property of the returned record returns spelling correct and the list is empty.

Example
This example looks for alternate spelling suggestions for the first word in the selection. If there are suggestions, the example runs a spelling check on the selection.

```plaintext
set l1 to list of (get spelling suggestions item to check (content of selection))
if (count of l1) is 0 then
display dialog "No suggestions"
else
    check spelling text object of selection
end if
```

Command: get story range

Returns a text range that represents the story specified by the story type argument.

Syntax
get story range  document  Required. An expression that returns a document object.

story type  enumeration  Required. Specifies the story range to return. Can be one of the following: main text story, footnotes story, endnotes story, comments story, text frame story, even pages header story, primary header story, even pages header footer story, primary footer story, first page header story, or first page footer story.
Remarks
If you attempt to return a story that isn't available in the specified document, an error occurs. The following example determines whether or not a footnote story is available in the active document.

```vba
try
    set myRange to get story range active document story type footnotes story
on error
    display dialog "The footnotes story is not available."
end try
```

Example
This example adds text to the primary header story and then displays the text.

```vba
set content of text object of (get header section 1 of active document index ¬
    header footer primary) to "Header text"
display dialog content of (get story range active document story type primary ¬
    header story) as string
```

Command: `get synonym info object`

Returns a `synonym info` object that contains information from the thesaurus on synonyms, antonyms, or related words and expressions for the specified word or phrase.

Syntax
`get synonym info`

- `item to check` *Unicode text* Required. The word or phrase to look up in the thesaurus.
- `language id` *enumeration* Optional. The language used for the thesaurus. Can be one of the following:
  - `afrikaans`
  - `arabic`
  - `basque`
  - `belgian dutch`
  - `belgian french`
  - `brazilian portuguese`
  - `bulgarian`
  - `byelorussian`
  - `catalan`
  - `croatian`
  - `czech`
  - `danish`
  - `dutch`
  - `english aus`
  - `english canadian`
  - `english new zealand`
  - `english south africa`
  - `english uk`
  - `english us`
  - `estonian`
  - `farsi`
  - `finnish`
  - `french`
  - `french canadian`
  - `german`
  - `greek`
  - `hebrew`
  - `hungarian`
  - `italian`
  - `icelandic`
  - `japanese`
  - `korean`
  - `language none`
  - `latvian`
  - `macedonian`
  - `malaysian`
  - `mexicanSpanish`
  - `language no proofing`
  - `norwegian bokmål`
  - `norwegian nynorsk`
  - `polish`
  - `portuguese`
  - `romanian`
  - `russian`
  - `serbian cyrillic`
  - `serbian latin`
  - `sesotho`
  - `simplified chinese`
  - `slovak`
  - `slovenian`
  - `spanish`
  - `spanish modern sort`
  - `swedish`
  - `swiss french`
  - `swiss german`
  - `swiss italian`
  - `traditional chinese`
  - `tsonga`
  - `tsswana`
  - `turkish`
  - `ukrainian`
  - `venda`
  - `xhosa`
  - `zulu`
Example
This example returns a list of antonyms for the word "big" in U.S. English.

```plaintext
set Alist to antonyms of (get synonym info object item to check "big" language id english us)
repeat with i in Alist
    display dialog i
end repeat
```

Command: `get theme name`

Retrieves the string name of a theme used in an existing Word application.

**Syntax**

`get theme name`

- **theme ID** integer Required. An index into the array of themes as listed in the Themes dialog box (Format menu).
- **use friendly name** Boolean Required. Determines whether or not the scheme name returned is one suitable for display in the user interface. If `false`, the theme name returned is suitable for use with the `apply theme` command.

**Remarks**

This command is used primarily to support wizards.

Command: `get webpage font`

Returns a web page font object for a particular character set.

**Syntax**

`get webpage font enumeration` Required. The character set from which you want to return the web page font object. Can be one of the following:

- Arabic character set
- Cyrillic character set
- English character set
- Greek character set
- Hebrew character set
- Japanese character set
- Korean character set
- Multilingual Unicode character set
- Simplified Chinese character set
- Thai character set
- Traditional Chinese character set
- Vietnamese character set

**Example**

The following example uses the `get webpage font` command to set `myFont` to the web page font object for the English character set in the active application.

```plaintext
set myFont to (get webpage font English character set)
```
**Command: get zoom**

Returns a `zoom` object that represents the magnification for the specified pane.

**Syntax**

```
get zoom   pane   Required   An expression that returns a `pane` object.
```

**Example**

This example sets the magnification in normal view to 100 percent for each open window.

```
repeat with myWindow in (get windows)
    set percentage of (get zoom active pane of myWindow zoom type normal view) to 100
end repeat
```

This example sets the magnification in page layout view so that an entire page is visible.

```
set page fit of (get zoom pane 1 of active window zoom type page view) to page fit full page
```

**Command: grow font**

Increases the font size to the next available size. If the selection or range contains more than one font size, each size is increased to the next available setting.

**Syntax**

```
grow font   font   Required. An expression that returns a `font` object.
```

**Example**

This example increases the font size of the fourth word in a new document.

```
set myDoc to make new document
set content of text object of myDoc to "This is a test of the grow font command."
display dialog "Click OK to increase the font size of the fourth word."
grow font font object of word 4 of myDoc
```

This example increases the font size of the selected text.

```
if selection type of selection is selection normal then
    grow font font object of text object of selection
else
    display dialog "You need to select some text."
end if
Command: home key

Moves or extends the selection to the beginning of the specified unit. This command returns a text range that represents the new text range of the selection, or returns a missing value if the move was unsuccessful.

Note This command corresponds to the functionality of the HOME key.

Syntax

home key selection object  Required. An expression that returns a selection object.

[move enumeration] Optional. The unit by which the selection is to be moved or extended. Can be one of the following: unit a story, unit a column, unit a screen, unit a section, unit a line, or unit a row. The default value is unit a line.

[extend enumeration] Optional. Specifies the way the selection is moved. Can be one of the following: by moving or by extending. If the value of this argument is by moving, the selection is collapsed to an insertion point and moved to the beginning of the specified unit. If the value of this argument is by extending, the beginning of the selection is extended to the beginning of the specified unit. The default value is by moving.

Example

This example moves the selection to the beginning of the current story. If the selection is in the main text story, the selection is moved to the beginning of the document.

home key selection move unit a story extend by moving

Command: inches to points

Converts a measurement from inches to points (1 inch = 72 points). Returns the converted measurement as a small real number.

Syntax

inches to points

inches  Required. The inch value to be converted to points.

Example

This example sets the space before for the selected paragraphs to 0.25 inch.

set space before of paragraph format of selection to ¬ inches to points inches 0.25

This example prints each open document after setting the left and right margins to 0.65 inch.

repeat with openDoc in (every document)
    set left margin of page setup of openDoc to inches to points inches 0.65
    set right margin of page setup of openDoc to inches to points inches 0.65
    print out openDoc
end repeat
Command: insert

Inserts the specified text at the end of a range or selection.

Syntax

insert

  text  Unicode text  Required. The text to be inserted.

  at    location reference  Required. The location to insert the new text. This argument must specify a text range within the document.

Example

This example inserts text at the end of the active document.

insert text "end of document" at end of text object of active document

Command: insert auto text

Attempts to match the text in the specified range or the text surrounding the range with an existing AutoText entry name. If any such match is found, insert auto text inserts the AutoText entry to replace that text. If a match cannot be found, an error occurs.

Syntax

insert auto text

  at    text range  Required. An expression that returns a text range object.

Example

This example inserts an AutoText entry that matches the text around a selection.

make new document
type text selection text "Best w"
insert auto text at text object of selection

This example inserts an AutoText entry with a name that matches the first word in the active document.

make new document
type text selection text "In "
set myRange to word 1 of active document
insert auto text at myRange
Command: insert auto text entry

Inserts the AutoText entry in place of the specified range. If you don't want to replace the range, use the collapse range command before using this command. The inserted AutoText entry is returned as a text range object.

Syntax

insert auto text entry  auto text entry  Required. An expression that returns an auto text entry object.

where  text range  Required text range object. The location for the AutoText entry.

[rich text  Boolean] Optional. True to insert the AutoText entry with its original formatting.

Example

This example inserts the formatted AutoText entry named 'one' after the selection.

collapse range text object of selection direction collapse end
insert auto text entry auto text entry "one" of normal template ¬
where text object of selection with rich text

Command: insert break

Inserts a page, column, or section break.

Syntax

insert break

at  text range  Required. An expression that returns a text range.

[break type  enumeration] Optional. The type of break to be inserted. Can be one of the following: page break, column break, section break next page, section break continuous, section break even page, section break odd page, or line break. The default value is page break.

Remarks

When you insert a page or column break, the text range is replaced by the break. If you don't want to replace the text range, use the collapse range command before using the insert break command. When you insert a section break, the break is inserted immediately preceding the text range object.

Example

This example inserts a page break immediately following the second paragraph in the active document.

set myRange to text object of paragraph 2 of active document
set myRange to collapse range myRange direction collapse end
insert break at myRange break type page break
Command: `insert caption`

Inserts a caption immediately preceding or following the specified range or selection.

**Syntax**

```plaintext
insert caption
   at Required. An expression that returns a text range object.

   caption label  enumeration  Required. The caption label to be inserted. Can be a string or one of
                  the following: caption equation, caption figure, or caption table. If the label hasn't yet been
                  defined, an error occurs. To define new caption labels, use the make command with the
                  caption label class.

   [title  Unicode text]  Optional. The string to be inserted immediately following the label in the
                          caption.

   [caption position  enumeration]  Optional. Specifies whether the caption will be inserted above
                                    or below the text range object. Can be either of the following: caption position above or
                                    caption position below.
```

**Example**

This example inserts a caption below the first table in the active document.

```
insert caption at text object of table 1 of active document caption label caption table caption position caption position below
```

This example inserts a Figure caption at the insertion point.

```
collapse range text object of selection direction collapse start
insert caption at text object of selection caption label caption figure title ": Sales Results" caption position caption position below
```

Command: `insert cells`

Adds cells to an existing table. The number of cells inserted is equal to the number of cells in the selection.

**Syntax**

```plaintext
insert cells  selection  Required. An expression that returns a selection object.

   [shift cells  enumeration]  Optional. Can be one of the following:
```

<table>
<thead>
<tr>
<th>Constant</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>insert cells entire column</td>
<td>Inserts an entire column to the left of the column that contains the selection.</td>
</tr>
<tr>
<td>insert cells entire row</td>
<td>Inserts an entire row above the row that contains the selection.</td>
</tr>
<tr>
<td>insert cells shift down</td>
<td>Inserts new cells above the selected cells.</td>
</tr>
<tr>
<td>insert cells shift right</td>
<td>Inserts new cells to the left of the selected cells.</td>
</tr>
</tbody>
</table>
Example
This example inserts new cells to the left of the selected cells, and then it surrounds the selected cells with a red, single-line border.

```plaintext
if (count of cells of selection) ≥ 1 then
    insert cells selection shift cells insert cells shift right
    set outside line style of border options of selection to line style single
    set outside color index of border options of selection to red
end if
```

Command: *insert columns*

Inserts columns to the left of the column that contains the selection. If the selection isn't in a table, an error occurs.

**Syntax**

```plaintext
insert columns selection
```

- **selection**: Required. An expression that returns a `selection` object.
- **position** (enumeration) Optional. The location to insert the columns relative to the selection. Can be either of the following: *insert on the right* or *insert on the left*.

**Example**

This example inserts new columns to the left of the column that contains the selection. The number of columns inserted is equal to the number of columns selected.

```plaintext
if (get selection information selection information type with in table) ¬
    is "True" then
    insert columns selection
    set texture of shading of selection to texture10 percent
end if
```

Command: *insert cross reference*

Inserts a cross-reference to a heading, bookmark, footnote, or endnote, or to an item for which a caption label is defined (for example, an equation, figure, or table).

**Syntax**

```plaintext
insert cross reference
```

- **at** (text range) Required. An expression that returns a `text range` object.
- **reference type** (enumeration) Required. The type of item for which a cross-reference is to be inserted. Can be one of the following: *reference type bookmark*, *reference type endnote*, *reference type footnote*, *reference type heading*, or *reference type numbered item*. 

279
**reference kind**  enumeration  Required. The information to be included in the cross-reference. Can be one of the following: **reference content text**, **reference endnote number**, **reference endnote number formatted**, **reference entire caption**, **reference footnote number**, **reference footnote number formatted**, **reference number full context**, **reference number no context**, **reference number relative text**, **reference only caption text**, **reference only label and number**, **reference page number**, or **reference position**.

**reference item**  Unicode text  Required. If **reference type** is **reference type bookmark**, this argument specifies a bookmark name. For all other **reference type** values, this argument specifies the item number or name in the **Reference type** box in the **Cross-reference** dialog box. To return a list of item names that can be used with this argument, use the **get cross reference items** command.

**[insert as hyperlink]**  Boolean  Optional. **True** to insert the cross-reference as a hyperlink to the referenced item.

**[include position]**  Boolean  Optional. **True** to insert "above" or "below," depending on the location of the reference item in relation to the cross-reference.

**Example**

This example inserts at the beginning of the active document a cross-reference to the page that includes the first bookmark in the document.

```plaintext
set myRange to create range active document start 0 end 0
set myBookmarks to (get cross reference items active document reference type ¬
    reference type bookmark)
set content of myRange to "Page "
set myRange to move end of range myRange by a word item count 1
collapse range myRange direction collapse end
insert cross reference at myRange reference type reference type bookmark ¬
    reference kind reference page number reference item ¬
    item 1 of myBookmarks
```

**Command: insert database**

Retrieves data from a data source (for example, a separate Word document) and inserts the data as a table in place of the specified range.

**Syntax**

```plaintext
insert database

at  text range  Required. An expression that returns a **text range** object.

[format  enumeration]  Optional. A format listed in the **Formats** box in the **Table AutoFormat** dialog box (**Table menu**).
```
[style  integer] Optional. The attributes of the AutoFormat specified by \texttt{Format} that are applied to the table. Use the sum of any combination of the following values:

<table>
<thead>
<tr>
<th>Value</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 (zero)</td>
<td>None</td>
</tr>
<tr>
<td>1</td>
<td>Borders</td>
</tr>
<tr>
<td>2</td>
<td>Shading</td>
</tr>
<tr>
<td>4</td>
<td>Font</td>
</tr>
<tr>
<td>8</td>
<td>Color</td>
</tr>
<tr>
<td>16</td>
<td>Auto Fit</td>
</tr>
<tr>
<td>32</td>
<td>Heading Rows</td>
</tr>
<tr>
<td>64</td>
<td>Last Row</td>
</tr>
<tr>
<td>128</td>
<td>First Column</td>
</tr>
<tr>
<td>256</td>
<td>Last Column</td>
</tr>
</tbody>
</table>

[link to source  Boolean] Optional. \texttt{True} to establish a link between the new table and the data source.

[connection  Unicode text] Optional. When retrieving data through ODBC, specify a connection string.

[SQL statement  Unicode text] Optional. An optional query string that retrieves a subset of the data in a primary data source to be inserted into the document.

[SQL statement1  Unicode text] Optional. If the query string is longer than 255 characters, \texttt{SQL statement} denotes the first portion of the string and \texttt{SQL statement1} denotes the second portion.

[password document  Unicode text] Optional. The password (if any) required to open the data source.

[password template  Unicode text] Optional. If the data source is a Word document, this argument is the password (if any) required to open the attached template.

[write password  Unicode text] Optional. The password required to save changes to the document.

[write password template  Unicode text] Optional. The password required to save changes to the template.

[data source  Unicode text] Optional. The path and file name of the data source.

[from  integer] Optional. The number of the first data record in the range of records to be inserted.

[to  integer] Optional. The number of the last data record in the range of records to be inserted.

[include fields  Boolean] Optional. \texttt{True} to include field names from the data source in the first row of the new table.
**Example**

This example inserts a Microsoft Excel spreadsheet named "Data.xls" after the selection. The style value (191) is a combination of the numbers 1, 2, 4, 8, 16, 32, and 128.

```plaintext
Example:
This example inserts a Microsoft Excel spreadsheet named "Data.xls" after the selection. The style value (191) is a combination of the numbers 1, 2, 4, 8, 16, 32, and 128.
```

**Command: insert date time**

Inserts the current date or time, or both, either as text or as a TIME field.

**Syntax**

```plaintext
insert date time

at text range Required. An expression that returns a text range.

[date time format] Unicode text Optional. The format to be used for displaying the date or time, or both.

[insert as field] Boolean Optional. True to insert the specified information as a TIME field. The default value is true.

Example

This example inserts a TIME field for the current date. A possible result might be "November 18, 1996."

insert date time at text object of selection date time format "MMMM dd, yyyy" with insert as field

This example inserts the current date at the end of the active document. A possible result might be "01/12/97."

collapse range text object of active document direction collapse end
insert date time at text object of active document date time format "MM/dd/yy" without insert as field

This example inserts a TIME field for the current date in the footer for the active document.

insert date time at text object of (get footer section 1 of active document index header footer primary) date time format "MMMM dd, yyyy" with insert as field
```
**Command: insert envelope data**

Inserts an envelope as a separate section at the beginning of the specified document. The arguments for this method correspond to the options in the Envelope dialog box (Tools menu).

**Syntax**

insert envelope data  envelope  Required. An expression that returns an envelope object.

- extract address  Boolean  Optional. True to use the text marked by the EnvelopeAddress bookmark (a user-defined bookmark) as the recipient's address.
- address  Unicode text  Optional. A string that specifies the recipient's address (ignored if extract address is True).
- auto text  Unicode text  Optional. A string that specifies an AutoText entry to use for the address. If specified, address is ignored.
- omit return address  Boolean  Optional. True to omit the return address.
- return address  Unicode text  Optional. A string that specifies the return address.
- return autotext  Unicode text  Optional. A string that specifies an AutoText entry to use for the return address. If specified, return address is ignored.
- print bar code  Boolean  Optional. True to add a POSTNET bar code. For U.S. mail only.
- print FIMA  Boolean  Optional. True to add a Facing Identification Mark (FIMA) for use in presorting courtesy reply mail. For U.S. mail only.
- envelope size  Unicode text  Optional. A string that specifies the envelope size. The string should match one of the sizes listed on the left in the Envelope size box in the Envelope Options dialog box (for example, "Size 10" or "C4").
- envelope height  integer  Optional. The height of the envelope (in points) when the envelope size argument is set to "Custom size."
- envelope width  integer  Optional. The width of the envelope (in points) when the envelope size argument is set to "Custom size."
- feed source  Boolean  Optional. True to use the feed source property of the specified envelope object to specify the paper tray to use when the envelope is printed.
- address from left  integer  Optional. The distance (in points) between the left edge of the envelope and the recipient's address.
- address from top  integer  Optional. The distance (in points) between the top edge of the envelope and the recipient's address.
- return address from left  integer  Optional. The distance (in points) between the left edge of the envelope and the return address.
- return address from top  integer  Optional. The distance (in points) between the top edge of the envelope and the return address.
- default face up  Boolean  Optional. True to print the envelope face up; false to print the envelope face down.
**[default orientation]** enumeration] Optional. The orientation for the envelope. Can be one of the following: left portrait, center portrait, right portrait, left landscape, center landscape, right landscape, left clockwise, center clockwise, or right clockwise.

**[size from page setup]** Boolean] Optional. True if the envelope’s address areas are sized according to settings in the Envelopes dialog box in the Page Setup dialog box (Tools menu). False if they are sized according to custom settings. The default value is true.

**[show page setup dialog]** Boolean] Optional. True if the Page Setup dialog box is displayed to allow adjustment of settings. Size from page setup must be set to true for the box to be displayed. The default value is false.

**[create new document]** Boolean] Optional. True if the envelope is not inserted into the active document but created separately. The default value is true.

**Example**
This example adds a Size 10 envelope to the active document by using the addresses stored in the tAddr and fAddr variables.

```vba
set tAddr to "Tony Wang" & return & "123 Skye St." & return & ¬
    "OurTown, WA 98107"
set fAddr to "Amy Rusko" & return & "456 Erde Lane" & return & ¬
    "OurTown, WA 98107"
insert envelope data envelope object of active document address tAddr ¬
    return address fAddr envelope size "Size 10"
```

**Command: insert file**

Inserts all or part of the specified file.

**Syntax**

```vba
insert file

at text range  Required. An expression that returns a text range object.

file name Unicode text  Required. The path and file name of the file to be inserted. If you don’t
    specify a path, Word assumes the file is in the current folder.

[file range] Unicode text  Optional. If the specified file is a Word document, this parameter
    refers to a bookmark. If the file is another type (for example, a Microsoft Excel worksheet), this
    parameter refers to a named range or a cell range (for example, R1C1:R3C4).

[confirm conversions] Boolean] Optional. True to have Word prompt you to confirm conversion
    when inserting files in formats other than the Word Document format.

[link] Boolean] Optional. True to insert the file by using an INCLUDETEXT field.
```

**Example**
This example uses an INCLUDETEXT field to insert the TEST.DOC file at the insertion point.

```vba
collapse range text object of selection direction collapse end
insert file at text object of selection file name ¬
    "Macintosh HD:Users:Shared:TEST.DOC" with link
```
Command: insert formula

Inserts an = (Formula) field that contains a formula at the selection.

Note The formula replaces the selection, if the selection isn't collapsed.

Syntax

insert formula selection object Required. An expression that returns a selection object.

[formula Unicode text] Optional. The mathematical formula you want the = (Formula) field to evaluate. Spreadsheet-type references to table cells are valid. For example, "=SUM(A4:C4)" specifies the first three values in the fourth row. For more information about the = (Formula) field, see Field codes:= (Formula) field.

[number format Unicode text] Optional. A format for the result of the = (Formula) field. For information about the types of formats you can apply, see Numeric Picture (\#) field switch.

Remarks

If you're using a spreadsheet application, such as Microsoft Excel, embedding all or part of a worksheet in a document is often easier than using the = (Formula) field in a table.

The formula argument is optional only if the selection is in a cell and there's at least one cell that contains a value above or to the left of the cell that contains the insertion point. If the cells above the insertion point contain values, the inserted field is [=SUM(ABOVE)]; if the cells to the left of the insertion point contain values, the inserted field is [=SUM(LEFT)]. If both the cells above the insertion point and the cells to the left of it contain values, Word uses the following rules to determine which SUM function to insert:

- If the cell immediately above the insertion point contains a value, Word inserts [=SUM(ABOVE)].
- If the cell immediately above the insertion point doesn't contain a value but the cell immediately to the left of the insertion point does, Word inserts [=SUM(LEFT)].
- If neither the cell immediately above the insertion point nor the cell immediately below it contains a value, Word inserts [=SUM(ABOVE)].
- If you don't specify formula and all the cells above and to the left of the insertion point are empty, using the = (Formula) field causes an error.

Example

This example creates a table with three rows and three columns at the beginning of the active document and then calculates the average of all the numbers in the first column.

```
set myRange to create range active document start 0 end 0
set myTable to make new table at myRange with properties {number of rows:3, number of columns:3}
set content of text object of cell 1 of row 1 of myTable to "100"
set content of text object of cell 1 of row 2 of myTable to "50"
select (cell 1 of row 3 of myTable)
insert formula selection formula "=Average(Above)"
```
This example inserts a formula field that's subtracted from a value represented by the bookmark named "GrossSales." The result is formatted with a dollar sign.

collapse range text object of selection direction collapse start
insert formula selection formula "=GrossSales-45,000.00" 
number format "$#,##0.00"

Command: insert paragraph

Replaces the specified range or selection with a new paragraph.

Note  After this method has been used, the text range is the new paragraph.

Syntax

insert paragraph
  at  text range  Required. An expression that returns a text range object.

Remarks

If you don't want to replace the text range, use the collapse range command before using this command.

Example

This example inserts a new paragraph at the beginning of the active document.

set myRange to create range active document start 0 end 0
insert paragraph at myRange
insert text "Dear Sirs, " at myRange

This example collapses the selection and then inserts a paragraph mark at the insertion point.

collapse range text object of selection direction collapse start
insert paragraph at text object of selection
collapse range text object of selection direction collapse end

Command: insert rows

Inserts the specified number of new rows above or below the row that contains the selection. If the selection isn't in a table, an error occurs.

Note  You can also insert rows by using the make command with the row class.

Syntax

insert rows  selection  Required. An expression that returns a selection object.

[position  enumeration]  Optional. Specifies whether to insert the new rows above or below the selection. Can be one of the following: above or below.

[number of rows  integer]  Optional. The number of rows to be added.
Example
This example inserts two new rows above the row that contains the selection, and then it removes the borders from the new rows.

```plaintext
if (get selection information selection information type with in table) -
  is "True" then
    insert rows selection position above number of rows 2
    set enable borders of border options of selection to false
end if
```

Command: insert symbol

Inserts a symbol in place of the specified range or selection.

**Note** If you don't want to replace the text range or selection, use the **collapse range** command before you use this command.

**Syntax**

```plaintext
insert symbol
    at text range  Required. An expression that returns a text range object.
    character number  integer  Required. The character number for the specified symbol. This value will always be the sum of 31 and the number that corresponds to the position of the symbol in the table of symbols (counting from left to right). For example, to specify a delta character at position 37 in the table of symbols in the Symbol font, set character number to 68.
    [font  Unicode text]  Optional. The name of the font that contains the symbol.
    [unicode  Boolean]  Optional. True to insert the Unicode character specified by character number; false to insert the ANSI character specified by character number. The default value is false.
    [bias  enumeration]  Optional. Sets the font bias for symbols. This argument is useful for setting the correct font bias for characters in East Asian versions of Word. Can be one of the following: font bias default, font bias do not care, or font bias east asian.
```

Example
This example inserts a double-headed arrow at the insertion point.

```plaintext
collapse range text object of selection direction collapse start
insert symbol at text object of selection character number 171 font "Symbol" -
  without unicode
```

This example inserts a bullet and a tab stop at the beginning of the first paragraph in the selection.

```plaintext
set myRange to text object of paragraph 1 of selection
set myRange to collapse range myRange direction collapse start
insert symbol at myRange character number 183 font "Symbol" without unicode
set myRange to move start of range myRange by a character item count 1
insert text tab at myRange
```
Command: key string

Returns the key combination string for the specified keys (for example, CTRL+SHIFT+A).

Syntax

key string

  key code  integer  Required. A key code you specify by using the build key code command to
          create the key code object.

  [key_code_2  enumeration]  Optional. A second key you specify by using one of the constants.

Example

This example displays the key combination string (CTRL+SHIFT+A) for the following constants:
control_key, shift_key, and a_key.

set customization context to attached template of active document

display dialog (key string key code (build key code key1 control_key
  ~
key2 shift_key key3 a_key))

Command: large scroll

Scrolls a window or pane by the specified number of screens. This command is equivalent to
clicking just before or just after the scroll boxes on the horizontal and vertical scroll bars.

Syntax

large scroll  window/pane  Required. An expression that returns a pane or window object.

  [down  integer]  Optional. The number of screens to scroll the window down.

  [up  integer]  Optional. The number of screens to scroll the window up.

  [to right  integer]  Optional. The number of screens to scroll the window to the right.

  [to left  integer]  Optional. The number of screens to scroll the window to the left.

Remarks

If down and up are both specified, the window is scrolled by the difference of the arguments. For
example, if down is 2 and up is 4, the window is scrolled up two screens. Similarly, if to left and to
right are both specified, the window is scrolled by the difference of the arguments.

Any of these arguments can be a negative number. If no arguments are specified, the window is
scrolled down one screen.

Example

This example scrolls the active window down one screen.

large scroll active window down 1

This example splits the active window and then scrolls up two screens and to the right one screen.

set split window of active window to true

large scroll active window up 2 to right 1
**Command: lines to points**

Converts a measurement from lines to points (1 line = 12 points). Returns the converted measurement as a small real number.

**Syntax**

`lines to points`

- `lines` (small real) Required. The line value to be converted to points.

**Example**

This example sets the paragraph line spacing in the selection to three lines.

```plaintext
set line spacing rule of paragraph format of selection to line space multiple
set line spacing of paragraph format of selection to (lines to points lines 3)
```

**Command: list commands**

Creates a new document and then inserts a table of Word commands along with their associated shortcut keys and menu assignments.

**Syntax**

`list commands`

- `list all commands` (Boolean) Required. True to include all Word commands and their assignments (whether customized or built-in). False to include only commands with customized assignments.

**Example**

This example creates a new document that lists all Word commands along with their associated shortcut keys and menu assignments. The example then prints and closes the new document without saving changes.

```plaintext
list commands with list all commands
print out active document
close active document saving no
```

**Command: list indent**

Increases the list level of the paragraphs in the range for the specified list format object, in increments of one level.

**Syntax**

`list indent list format`

- `list format` Required. An expression that returns a list format object.

**Example**

This example indents each paragraph in the first list in document one by one level.

```plaintext
list indent list format of text object of Word list 1 of document 1
```
This example formats paragraphs four through eight in the active document as an outline-numbered list, and then it indents the paragraphs one level.

```
set myDoc to active document
set myRange to create range myDoc start (start of content of text object of paragraph 4 of myDoc) end (end of content of text object of paragraph 8 of myDoc)
apply list format template list format of myRange list template (list template 1 of (get list gallery outline number gallery))
list indent list format of myRange
```

Command: list outdent

Decreases the list level of the paragraphs in the range for the specified list format object, in increments of one level.

**Syntax**

```
list outdent  list format  Required. An expression that returns a list format object.
```

**Example**

This example reduces the indent of each paragraph in the first list in the active document by one level.

```
list outdent list format of text object of Word list 1 of active document
```

This example formats paragraphs four through eight in MyDoc.doc as an outline-numbered list, indents the paragraphs one level, and then removes the indent from the list.

```
set myDoc to document "MyDoc.doc"
set myRange to create range myDoc start (start of content of text object of paragraph 4 of myDoc) end (end of content of text object of paragraph 8 of myDoc)
apply list format template list format of myRange list template (list template 1 of (get list gallery outline number gallery))
list indent list format of myRange
list outdent list format of text object of paragraph 4 of myDoc
```

Command: make compatibility default

Sets the compatibility options in the Compatibility pane of the Preferences dialog box (Word menu) as the default settings for new documents.

**Syntax**

```
make compatibility default  document  Required. An expression that returns a document object.
```
Example
This example sets a few compatibility options for the active document and then makes the current compatibility options the default settings.

```
set myDoc to active document
set document compatibility myDoc compatibility item suppress space before after page break with is compatible
set document compatibility myDoc compatibility item expand shift return with is compatible
set document compatibility myDoc compatibility item use printer metrics with is compatible
set document compatibility myDoc compatibility item no extra line spacing without is compatible
make compatibility default myDoc
```

Command: make new data merge ask field

Adds an ASK field to a mail merge main document. Returns a data merge field object. When updated, an ASK field displays a dialog box that prompts you for text to assign to the specified bookmark.

Syntax

```
make new data merge ask field data merge of active document ¬
   text range myRange name "company" prompt "Type your company name" ¬
   with ask once
```

Example
This example adds an ASK field at the end of the active mail merge main document.

```
set myRange to text object of active document
set myRange to collapse range myRange direction collapse end
make new data merge ask field data merge of active document ¬
   text range myRange name "company" prompt "Type your company name" ¬
   with ask once
```
Command: make new data merge fill in field

Adds a FILLIN field to a mail merge main document. Returns a data merge field object. When updated, a FILLIN field displays a dialog box that prompts you for text to insert into the document at the location of the FILLIN field.

Note To add a FILLIN field to a document other than a mail merge main document, use the make command with a field object.

Syntax
make new data merge fill in field data merge Required. An expression that returns a data merge object.

text range text range Required text range object. The location for the FILLIN field.
[prompt Unicode text] Optional. The text that's displayed in the dialog box.
[default fill in text Unicode text] Optional. The default response, which appears in the text box when the dialog box is displayed. Corresponds to the \d switch for an FILLIN field.
[ask once Boolean] Optional. True to display the prompt only once instead of each time a new data record is merged. Corresponds to the \o switch for a FILLIN field. The default value is false.

Example
This example adds a FILLIN field that prompts you for a name to insert after "Name: ".
collapse range text object of selection direction collapse start
insert text "Name: " at text object of selection
move range text object of selection by a word item count 2
make new data merge fill in field data merge of active document ¬
text range text object of selection prompt "Your name?" ¬
default fill in text "Joe" with ask once

Command: make new data merge if field

Adds an IF field to a mail merge main document. Returns a data merge field object. When updated, an IF field compares a field in a data record with a specified value, and then it inserts the appropriate text according to the result of the comparison.

Syntax
make new data merge if field data merge Required. An expression that returns a data merge object.

text range text range Required text range object. The location for the IF field.
merge field Unicode text Required. The merge field name.
comparison enumeration Required. The operator used in the comparison. Can be one of the following: merge if equal, merge if greater than, merge if greater than or equal, merge if is blank, merge if is not blank, merge if less than, merge if less than or equal, or merge if not equal.
[compare to  Unicode text]  Optional. The text to compare with the contents of merge field.
[true text  Unicode text]  Optional. The text that's inserted if the comparison is true.
[false text  Unicode text]  Optional. The text that's inserted if the comparison is false.

Example
This example inserts "for your personal use" if the Company merge field is blank and "for your business" if the Company merge field is not blank.

make new data merge if field data merge of active document text range ¬
    text object of selection merge field "Company" comparison ¬
    merge if is blank true text "for your personal use" false text ¬
    "for your business"

This example inserts an IF field that compares the contents of the merge field named "Title" with the text "Mr." When the merge is performed, "Hello" is inserted if the comparison is true.

set show field codes of view of active window to false
set myRange to create range active document start 0 end 0
make new data merge if field data merge of active document text range ¬
    myRange merge field "Title" comparison merge if equal compare to ¬
    "Mr." true text "Hello "

Command: make new data merge next field

Adds a NEXT field to a mail merge main document. Returns a data merge field object. A NEXT field advances to the next data record so that data from more than one record can be merged into the same merge document (for example, a sheet of mailing labels).

Syntax
make new data merge next field  data merge  Required. An expression that returns a data merge object.
    text range  text range  Required text range object. The location for the NEXT field.

Example
This example adds a NEXT field after the third MERGEFIELD field in Main.doc.

select data merge field 3 of data merge of document "Main.doc"
collapse range text object of selection direction collapse end
make new data merge next field data merge of document "Mail.doc" text range ¬
    text object of selection
Command: make new data merge next if field

Adds a NEXTIF field to a mail merge main document. Returns a data merge object. A NEXTIF field compares two expressions, and if the comparison is true, the next data record is merged into the current merge document.

Syntax

make new data merge next if field  data merge  Required. An expression that returns a data merge object.

text range  text range  Required text range object. The location for the NEXTIF field.

merge field  Unicode text  Required. The merge field name.

comparison  enumeration  Required. The operator used in the comparison. Can be one of the following: merge if equal, merge if greater than, merge if greater than or equal, merge if is blank, merge if is not blank, merge if less than, merge if less than or equal, or merge if not equal.

[compare to  Unicode text]  Optional. The text to compare with the contents of merge field.

Example

This example adds a NEXTIF field before the first MERGEFIELD field in Main.doc. If the next postal code equals 98004, the next data record is merged into the current merge document.

```plaintext
select data merge field 1 of data merge of document "Main.doc"
collapse range text object of selection direction collapse start
make new data merge next if field data merge of document "Main.doc" ¬
  text range text object of selection merge field "PostalCode" ¬
  comparison merge if equal compare to "98004"
```

Command: make new data merge rec field

Adds a MERGEREC field to a mail merge main document. Returns a data merge object. A MERGEREC field inserts the number of the current data record (the position of the data record in the current query result) during a mail merge.

Syntax

make new data merge rec field  data merge  Required. An expression that returns a data merge object.

text range  text range  Required text range object. The location for the MERGEREC field.

Example

This example inserts text and a MERGEREC field at the beginning of the active document.

```plaintext
set myRange to create range active document start 0 end 0
make new data merge rec field data merge of active document text range myRange
insert text "Record Number: " at myRange
```
Command: make new data merge sequence field

Adds a MERGESEQ field to a mail merge main document. Returns a data merge field object. A MERGESEQ field inserts a number based on the sequence in which data records are merged (for example, when record 50 of records 50 to 100 is merged, MERGESEQ inserts the number 1).

Syntax

make new data merge sequence field  Required. An expression that returns a data merge object.

text range  text range  Required text range object. The location for the MERGESEQ field.

Example

This example inserts text and a MERGESEQ field at the end of the active document.

set myRange to text object of active document
set myRange to collapse range myRange direction collapse end
make new data merge sequence field data merge of active document ¬
text range myRange
insert text "Sequence Number: " at myRange

Command: make new data merge set field

Adds a SET field to a mail merge main document. Returns a data merge field object. A SET field defines the text of the specified bookmark.

Syntax

make new data merge set field  data merge  Required. An expression that returns a data merge object.

text range  text range  Required text range object. The location for the SET field.

name  Unicode text  Required. The bookmark name that value text is assigned to.

[value text  Unicode text]  Optional. The text associated with the bookmark specified by the name argument.

Example

This example adds a SET field at the beginning of the active document and then adds a REF field to display the text after the selection.

set myRange to create range active document start 0 end 0
make new data merge set field data merge of active document text range ¬
myRange name "Name" value text "Joe Smith"
collapse range text object of selection direction collapse end
make new field at active document with properties {text range:text object ¬
of selection, field type:field ref, field text:"Name"}
Command: make new data merge skip if field

Adds a SKIPIF field to a mail merge main document. Returns a data merge field object. A SKIPIF field compares two expressions, and if the comparison is true, SKIPIF moves to the next data record in the data source and starts a new merge document.

Syntax

make new data merge skip if field  
data merge  Required. An expression that returns a data merge object.

text range  text range  Required text range object. The location for the SKIPIF field.

merge field  Unicode text  Required. The merge field name.

comparison  enumeration  Required. The operator used in the comparison. Can be one of the following: merge if equal, merge if greater than, merge if greater than or equal, merge if is blank, merge if is not blank, merge if less than, merge if less than or equal, or merge if not equal.

[compare to  Unicode text]  Optional. The text to compare with the contents of merge field.

Example

This example adds a SKIPIF field before the first MERGEFIELD field in Main.doc. If the next postal code equals 98040, the next data record is skipped and a new merge document is created.

select data merge field 1 of data merge of document "Main.doc"
collapse range text object of selection direction collapse start
make new data merge skip if field data merge of document "Main.doc" ¬
text range text object of selection merge field "PostalCode" ¬
comparison merge if equal compare to "98040"

Command: manual hyphenation

Initiates manual hyphenation of a document, one line at a time. The user is prompted to accept or decline suggested hyphenations.

Syntax

manual hyphenation  document  Required. An expression that returns a document object.

Example

This example starts manual hyphenation of the active document.

manual hyphenation active document

This example sets hyphenation options and then starts manual hyphenation of MyDoc.doc.

set hyphenation zone of document "MyDoc.doc" to inches to points inches 0.25
set hyphenate caps of document "MyDoc.doc" to false

manual hyphenation document "MyDoc.doc"
Command: mark entry for table of contents

Inserts a TC (Table of Contents Entry) field after the specified range. The TC field is returned as a field object.

Syntax

mark entry for table of contents  document  Required. An expression that returns a document object.

range  text range  Required text range object. The location of the entry. The TC field is inserted after range.

entry  Unicode text  Required. The text that appears in the table of contents. To indicate a subentry, include the main entry text and the subentry text, separated by a colon (;) (for example, "Introduction: The Product").

[table id]  Unicode text  Optional. A one-letter identifier for the table of figures or table of contents item (for example, "i" for an "illustration").

[level]  integer  Optional. A level for the entry in the table of contents or table of figures.

Example

This example inserts a table of contents entry that references the selected text. The text typed in the input box appears in the table of contents. A table of contents that uses fields is then added at the beginning of the active document.

set entryText to text returned of (display dialog "Type entry text" ~
  default answer "")
mark entry for table of contents active document range text object ~
  of selection entryText
set myRange to create range active document start 0 end 0
make new table of contents at active document with properties ~
  {text object:myRange, use fields: true, use heading styles: false}

Command: mark entry for table of figures

Inserts a TC (Table of Contents Entry) field after the specified range. The TC field is returned as a field object.

Syntax

mark entry for table of figures  document  Required. An expression that returns a document object.

range  text range  Required text range object. The location of the entry. The TC field is inserted after range.

entry  Unicode text  Required. The text that appears in the table of figures. To indicate a subentry, include the main entry text and the subentry text, separated by a colon (;) (for example, "Introduction: The Product").
[table id] Optional. A one-letter identifier for the table of figures item (for example, "i" for an "illustration").

[level] Optional. A level for the entry in the table of contents or table of figures.

Example
This example inserts a table of figures entry that references the selected text. The text typed in the input box appears in the table of figures. A table of figures that uses fields is then added at the beginning of the active document.

set entryText to text returned of (display dialog "Type entry text" default answer "")
mark entry for table of figures active document range text object of selection entryText
set myRange to create range active document start 0 end 0
make new table of figures at active document with properties {text object:myRange, use fields:true, use heading styles:false}

Command: mark for index
Inserts an XE (Index Entry) field after the specified range. The XE field is returned as a field object.

Syntax
mark for index document Required. An expression that returns a document object.
range text range Required text range object. The location of the entry. The XE field is inserted after range.
entry Unicode text Required. The text that appears in the index. To indicate a subentry, include the main entry text and the subentry text, separated by a colon (;) (for example, "Introduction: The Product").
[cross reference] Unicode text Optional. A cross-reference that will appear in the index (for example, "See Apples").
[bookmark name] Unicode text Optional. The name of the bookmark that marks the range of pages you want to appear in the index. If this argument is omitted, the number of the page containing the XE field appears in the index.

Example
This example inserts an index entry after the selection in the active document. The subentry text is the text from the selection.

if selection type of selection is selection normal then
mark for index active document range text object of selection entry "Introduction: " & content of text object of selection
end if
**Command: merge**

Merges the changes marked with revision marks from one document to another.

**Syntax**

- `merge` document Required. An expression that returns a **document** object.
- `file name` Unicode text Required. The path and file name of the original document you want to merge revisions with.

**Example**

This example merges changes from the active document into Sales1.doc.

```
merge active document file name "Macintosh HD:Users:Shared:Sales1.doc"
```

**Command: merge subdocuments**

Merges the specified subdocuments of a master document into a single subdocument.

**Syntax**

- `merge subdocuments` document Required. An expression that returns a master document that contains the subdocuments to be merged.
- `[first subdocument subdocument]` Optional. The first subdocument in a range of subdocuments to be merged.
- `[last subdocument subdocument]` Optional. The last subdocument in a range of subdocuments to be merged.

**Example**

This example merges the first and second subdocuments in the active document into one subdocument.

```
if (count of subdocuments of active document) ≥ 2 then
    set aDoc to active document
    merge subdocuments aDoc first subdocument subdocument 1 of aDoc last ¬
    subdocument subdocument 2 of aDoc
end if
```

**Command: millimeters to points**

Converts a measurement from millimeters to points (1 mm = 2.85 points). Returns the converted measurement as a small real number.

**Syntax**

- `millimeters to points`
  - `millimeters` small real Required. The millimeter value to be converted to points.
Example
This example sets the hyphenation zone in the active document to 8.8 millimeters.

```
set hyphenation zone of active document to millimeters to points ¬
millimeters 8.8
```

This example expands the spacing of the selected characters to 2.8 points.

```
set spacing of font object of selection to millimeters to points millimeters 1
```

**Command:** modified

Returns `true` if the specified list template is not the built-in list template for that position in the list gallery.

**Syntax**

```
modified  list gallery  Required. An expression that returns a list gallery object.

index  Required Long. A number from 1 to 7 that corresponds to the position of the template in
       the Bullets and Numbering dialog box. Skipping the None option, the templates are
       numbered from left to right, starting with the top row.
```

**Remarks**

To set a list template in a list gallery back to the built-in list template, use the `reset list gallery` command.

**Example**

This example checks to see whether the first template on the Bulleted tab in the Bullets and Numbering dialog box has been changed. If it has, the list template is reset.

```
set temp to modified (get list gallery bullet gallery) index 1
if temp is true then
    reset list gallery (get list gallery bullet gallery) index 1
else
    display dialog "This is the built-in list template"
end if
```

**Command:** next for browser

Moves the selection to the next item indicated by the browser target. Use the browser target property to change the browser target.

**Syntax**

```
next for browser  browser  Required. An expression that returns a browser object.
```

**Example**

This example moves the insertion point just before the next comment reference marker in the active document.

```
set browser target of browser object to browse comment
next for browser browser object
```
**Command: next header footer**

If the selection is in a header, this method moves to the next header within the current section (for example, from an odd header to an even header) or to the first header in the following section. If the selection is in a footer, this method moves to the next footer.

**Note** If the selection is in the last header or footer in the last section of the document, or if it’s not in a header or footer at all, an error occurs.

**Syntax**

`next header footer view`  Required. An expression that returns a `view` object.

**Example**

This example displays the first page header in the active document and then switches to the next header. The document needs to be at least two pages long.

```vba
set different first page header footer of page setup of active document to true
set view type of view of active window to page view
set seek view of view of active window to seek first page header
next header footer view of active window
```

**Command: next revision**

Locates and returns the next tracked change as a `revision` object.

**Syntax**

`next revision`  `selection object`  Required. An expression that returns a `selection` object.

`[wrap Boolean]`  Optional. True to continue searching for a revision at the beginning of the document when the end of the document is reached. The default value is `false`.

**Command: on time**

 Starts a background timer that runs a VB macro on the specified date and at the specified time.

**Syntax**

`on time`

`when`  Unicode text  Required. The time at which the macro is to be run. Can be a string that specifies a time (for example, "4:30 pm" or "16:30"), or it can be a serial number returned by a function. You can also include the date (for example, "6/30 4:15 pm").

`name`  Unicode text  Required. The name of the macro to be run. Use the complete macro path to ensure that the correct macro is run (for example, "Project.Module1.Macro1"). For the macro to run, the document or template must be available both when the on time instruction is run and when the time specified by when arrives. For this reason, it’s best to store the macro in Normal.dot or another global template that’s loaded automatically.
[tolerance  integer] Optional. The maximum time (in seconds) that can elapse before a macro that wasn’t run at the time specified by when is canceled. Macros may not always run at the specified time. For example, if a sort operation is under way or a dialog box is being displayed, the macro will be delayed until Word has completed the task. If this argument is 0 (zero) or omitted, the macro is run regardless of how much time has elapsed since the time specified by when.

Remarks
Word can maintain only one background timer set by on time. If you start another timer before an existing timer runs, the existing timer is canceled.

Example
This example runs the macro named "MyMacro" in the current module at 3:55 P.M.

on time when "15:55:00" name "MyMacro"

Command: open as document
Opens the specified template as a document and returns a document object.

Note Opening a template as a document allows the user to edit the contents of the template. This may be necessary if a property or command (the styles property, for example) isn’t available from the Template object.

Syntax
open as document  template  Required. An expression that returns a template object.

Example
This example opens the template attached to the active document, displays a message box if the template contains anything more than a single paragraph mark, and then closes the template.

set aDoc to open as document attached template of active document
if content of text object of aDoc is not return then
   display dialog "The template is not empty"
else
   display dialog "The template is empty"
end if
close aDoc saving no
This example saves a copy of the Normal template as "Backup.dot."

set aDoc to open as document normal template
save as aDoc file name "Backup.dot"
close aDoc saving no
This example changes the formatting of the Heading 1 style in the template attached to the active document. The **update styles** command updates the styles in the active document.

```vba
set aDoc to open as document attached template of active document
set name of font object of Word style style heading1 of aDoc to "Arial"
set font size of font object of Word style style heading1 of aDoc to 16
set bold of font object of Word style style heading1 of aDoc to false
close aDoc saving yes
update styles active document
```

**Command: open data source**

Attaches a data source to the specified document, which becomes a main document if it's not one already.

**Syntax**

```vba
open data source  data merge  Required. An expression that returns a data merge object.
    name  Unicode text  Required. The data source file name. You can specify a Microsoft Query (.qry) file instead of specifying a data source, a connection string, and a query string.
    [format  enumeration]  Optional. The file converter used to open the document. Can be one of the following: open format auto, open format document, open format mac readable, open format rtf, open format template, open format text, open format unicode text, or open format web pages. The default value is open format auto.
        To specify an external file format, use the open format property with a file converter object to determine the value to use with this argument.
    [confirm conversions  Boolean]  Optional. True to display the Convert File dialog box if the file isn't in Word format.
    [read only  Boolean]  Optional. True to open the data source on a read-only basis.
    [link to source  Boolean]  Optional. True to perform the query specified by connection and SQL statement each time the main document is opened.
    [add to recent files  Boolean]  Optional. True to add the file name to the list of recently used files at the bottom of the File menu.
    [password document  Unicode text]  Optional. The password used to open the data source.
    [password template  Unicode text]  Optional. The password used to open the template.
    [Revert  Boolean]  Optional. Controls what happens if name is the file name of an open document. True to discard any unsaved changes to the open document and reopen the file; false to activate the open document.
    [write password  Unicode text]  Optional. The password used to save changes to the document.
    [WritePasswordTemplate  Unicode text]  Optional. The password used to save changes to the template.
    [connection  Unicode text]  Optional. A range within which the query specified by SQL statement is to be performed. When retrieving data through ODBC, you specify a connection string.
```
Microsoft Word Suite


[SQL statement1]  Unicode text]  Optional. If the query string is longer than 255 characters, SQL statement specifies the first portion of the string, and SQL statement1 specifies the second portion.

Example
This example creates a new main document and attaches the document "Data.doc" as the data source.

set myDoc to make new document
set main document type of data merge of myDoc to document type form letters
open data source data merge of myDoc name "data.doc"

Command: open document from workmenu
Opens the document associated with the specified work menu item and returns it as a document object.

Syntax
open document from workmenu  work menu item  Required. An expression that returns a work menu item object.

Example
This example opens each document on the Work menu.

repeat with i in (get work menu items)
    open document from workmenu i
end repeat

Command: open header source
Attaches a data merge header source to the specified document.

Note  When a header source is attached, the first record in the header source is used in place of the header record in the data source.

Syntax
open header source  data merge  Required. An expression that returns a data merge object.

name  Unicode text  Required. The header source file name.

[format  enumeration]  Optional. The file converter used to open the document. Can be one of the following: open format auto, open format document, open format mac readable, open format rtf, open format template, open format text, open format unicode text, or open format web pages. The default value is open format auto.

To specify an external file format, use the open format property with a file converter object to determine the value to use with this argument.

[confirm conversions  Boolean]  Optional. True to display the Convert File dialog box if the file isn't in Word format.
[read only] Boolean] Optional. True to open the header source on a read-only basis.

[add to recent files] Boolean] Optional. True to add the file name to the list of recently used files at the bottom of the File menu.

[password document] Unicode text] Optional. The password used to open the header source.

[password template] Unicode text] Optional. The password used to open the template.

[Revert] Boolean] Optional. Controls what happens if name is the file name of an open document. True to discard any unsaved changes to the open document and reopen the file; false to activate the open document.

[write password] Unicode text] Optional. The password used to save changes to the document.

[write password template] Unicode text] Optional. The password used to save changes to the template.

Example
This example sets the active document as a main document for form letters, and then it attaches the header source named "Header.doc" and the data document named "Names.doc."

set main document type of data merge of active document to ¬
    document type form letters
open header source data merge of active document ¬
    name "Macintosh HD:Users:Shared:Header.doc" ¬
    without add to recent files and Revert
open data source data merge of active document ¬
    name "Macintosh HD:Users:Shared:Names.doc"

Command: open recent file
Opens the recent file and returns a document object.

Syntax
open recent file  recent file  Required. An expression that returns a recent file object.

Example
This example opens each document in the recent files list.
repeat with r in (get recent files)
    open recent file r
end repeat

Command: open subdocument
Opens the specified subdocument and returns a document object.

Syntax
open subdocument  subdocument  Required. An expression that returns a subdocument object.
Example
This example opens each subdocument in the master document "Master.doc".
repeat with s in (get subdocuments of document "Master.doc")
  open subdocument s
end repeat

Command: open version
Opens the specified document and adds it to the documents list. Returns a document object.
Syntax
open version  document version  Required. An expression that returns a document version object.
Example
This example opens the most recent version of Report.doc.
set myDoc to document "Report.doc"
open version last document version of myDoc

Command: organizer copy
Copies the specified AutoText entry, toolbar, style, or macro project item from the source document or template to the destination document or template.
Syntax
organizer copy
  source  Unicode text  Required. The document or template file name that contains the item you want to copy.
  destination  Unicode text  Required. The document or template file name to which you want to copy an item.
  name  Unicode text  Required. The name of the AutoText entry, toolbar, style, or macro you want to copy.
  organizer object type  enumeration  Required. The kind of item you want to copy. Can be one of the following: organizer object auto text, organizer object command bars, organizer object project items, or organizer object styles.
Example
This example copies all the AutoText entries in the template attached to the active document to the Normal template.
repeat with aEntry in (get auto text entries of attached template of ~
  active document)
  organizer copy source (full name of attached template of ~
    active document) destination (full name of normal template) ~
    name (name of aEntry) organizer object type organizer object ~
    auto text
end repeat
If the style named "SubText" exists in the active document, this example copies the style to Macintosh HD:Users:Shared:MyTemplate.dot.

```plaintext
if exists (Word style "SubText" of active document) then
    organizer copy source (name of active document) destination
    "Macintosh HD:Users:Shared:MyTemplate.dot" name
    "SubText" organizer object type organizer object styles
end if
```

---

**Command: organizer delete**

Deletes the specified style, AutoText entry, toolbar, or macro project item from a document or template.

**Syntax**

```
organizer delete

source  Unicode text  Required. The document or template file name that contains the item you want to delete.
name    Unicode text  Required. The name of the AutoText entry, toolbar, style, or macro you want to delete.
organizer object type  enumeration  Required. The kind of item you want to delete. Can be one of the following: organizer object auto text, organizer object command bars, organizer object project items, or organizer object styles.
```

**Example**

This example deletes the toolbar named "Custom 1" from the Normal template.

```plaintext
if exists (command bar "Custom 1") then
    organizer delete source (full name of normal template) name
    "Custom 1" organizer object type organizer object command bars
end if
```

---

**Command: organizer rename**

Renames the specified style, AutoText entry, toolbar, or macro project item in a document or template.

**Syntax**

```
organizer rename

source  Unicode text  Required. The document or template file name that contains the item you want to rename.
name    Unicode text  Required. The name of the AutoText entry, toolbar, style, or macro you want to rename.
new name  Unicode text  Required. The new name for the item.
organizer object type  enumeration  Required. The kind of item you want to rename. Can be one of the following: organizer object auto text, organizer object command bars, organizer object project items, or organizer object styles.
```
Example
This example changes the name of the style named "SubText" in the active document to "SubText2."
if exists (Word style "SubText" of active document) then
    organizer rename source (full name of active document) name ¬
    "SubText" new name "SubText2" organizer object type ¬
    organizer object styles
end if

Command: page scroll
Scrolls through the specified pane or window page by page.

Syntax
page scroll   window/pane   Required. An expression that returns a window or pane object.
  [down   integer]     Optional. The number of pages to be scrolled down. If this argument is omitted,
                       this value is assumed to be 1.
  [up     integer]     Optional. The number of pages to be scrolled up.

Remarks
The page scroll command is available only if you're in page layout view or online layout view. This
command doesn't affect the position of the insertion point.
If down and up are both specified, the window is scrolled by the difference of the arguments. For
example, if down is 2 and up is 4, the window is scrolled up two pages.

Example
This example scrolls down three pages in the active window.
set view type of view of active window to page view
page scroll active window down 3
This example scrolls up one page in the active pane.
set view type of view of active window to page view
page scroll active pane of active window up 1
This example scrolls down one page in the active window.
set view type of view of active window to page view
page scroll active window

Command: paste format
Applies formatting copied with the copy format command to the selection. If a paragraph mark was
selected when the copy format command was used, Word applies paragraph formatting in addition to character formatting.

Syntax
paste format   selection object   Required. An expression that returns a selection object.
Example

This example copies the paragraph and character formatting from the first paragraph in the active document to the next paragraph in the active document.

select text object of paragraph 1 of active document
copy format selection
select text object of (next paragraph paragraph 1 of active document)
paste format selection

This example collapses the selection and copies the character formatting to the next word.
collapse range text object of selection direction collapse start
copy format selection
select (next range text object of selection by a word item count 1)
paste format selection

Command: paste object

Inserts the contents of the Clipboard at the specified text range. If you don't want to replace the contents of the text range, use the collapse range command before using this command.

Syntax

paste object  selection object  Required. An expression that returns a selection object (the location where you want to paste the object).

Example

This example copies and pastes the first table in the active document into a new document.

if (count of tables in active document) ≥ 1 then
  select table 1 of active document
  copy object selection
  set newDoc to make new document
  paste object text object of newDoc
end if

This example copies the first paragraph in the document and pastes it at the insertion point.
copy object text object of paragraph 1 of active document
collapse range text object of selection direction collapse start
paste object text object of selection

This example copies the selection and pastes it at the end of the document.

if selection type of selection is not selection ip then
  copy object selection
  set range2 to text object of active document
  set range2 to collapse range range2 direction collapse end
  paste object range2
end if
**Command: picas to points**

Converts a measurement from picas to points (1 pica = 12 points). Returns the converted measurement as a small real number.

**Syntax**

```
picas to points
```

- `picas` small real  Required. The pica value to be converted to points.

**Example**

This example adds line numbers to the active document and sets the distance between the line numbers and the document text to 4 picas.

```
set active line of line numbering of page setup of active document to true
set distance from text of line numbering of page setup of active document to picas to points picas 4
```

This example sets the first-line indent for the selected paragraphs to 3 picas.

```
set first line indent of paragraph format of selection to picas to points picas 3
```

---

**Command: points to centimeters**

Converts a measurement from points to centimeters (1 centimeter = 28.35 points). Returns the converted measurement as a small real number.

**Syntax**

```
points to centimeters
```

- `points` small real  Required. The measurement, in points.

**Example**

This example converts a measurement of 30 points to the corresponding number of centimeters.

```
display dialog ((points to centimeters points 30) as string) & " centimeters"
```

---

**Command: points to inches**

Converts a measurement from points to inches (1 inch = 72 points). Returns the converted measurement as a small real number.

**Syntax**

```
points to inches
```

- `points` small real  Required. The measurement, in points.

**Example**

This example converts the measurement of the top margin for the active document to inches and displays the result in a message box.

```
display dialog (points to inches points (top margin of page setup of active document))
```
**Command: points to lines**

Converts a measurement from points to lines (1 line = 12 points). Returns the converted measurement as a small real number.

**Syntax**

`points to lines`  
`points`  small real  Required. The measurement, in points.

**Example**

This example converts the line spacing value of the first paragraph in the selection from points to lines.

```
display dialog ((points to lines points (line spacing of paragraph 1 ~ of selection)) as string) & " lines"
```

**Command: points to millimeters**

Converts a measurement from points to millimeters (1 millimeter = 2.835 points). Returns the converted measurement as a small real number.

**Syntax**

`points to millimeters`  
`points`  small real  Required. The measurement, in points.

**Example**

This example converts 72 points to the corresponding number of millimeters.

```
display dialog ((points to millimeters points 72) as string) & " millimeters"
```

**Command: points to picas**

Converts a measurement from points to picas (1 pica = 12 points). Returns the converted measurement as a small real number.

**Syntax**

`points to picas`  
`points`  small real  Required. The measurement, in points.

**Example**

This example converts 36 points to the corresponding number of picas.

```
display dialog ((points to picas points 36) as string) & " picas"
```

**Command: present it**

Opens PowerPoint with the specified Word document loaded.

**Syntax**

`present it`  `document`  Required. An expression that returns a `document` object.
Command: previous for browser

Moves the selection to the previous item indicated by the browser target. Use the browser target property to change the browser target.

Syntax

previous for browser browser Required. An expression that returns a browser object.

Example

This example moves the insertion point into the first cell (the cell in the upper-left corner) of the previous table.

set browser target of browser object to browse table
previous for browser browser object

Command: previous header footer

If the selection is in a header, this command moves to the previous header within the current section (for example, from an even header to an odd header) or to the last header in the previous section. If the selection is in a footer, this command moves to the previous footer.

Note If the selection is in the first header or footer in the first section of the document, or if it's not in a header or footer at all, an error occurs.

Syntax

previous header footer view Required. An expression that returns a view object.

Example

This example inserts an even section break, switches the active window to page layout view, displays the current header, and then switches to the previous header.

collapse range text object of selection direction collapse start
insert break at text object of selection break type section break even page
set view type of view of active window to page view
set seek view of view of active window to seek current page header
previous header footer view of active window

Command: previous revision

Locates and returns the previous tracked change as a revision object.

Syntax

previous revision selection object Required. An expression that returns a selection object.

[wrap Boolean] Optional. True to continue searching for a revision at the end of the document when the beginning of the document is reached. The default value is false.
Command: print out

Prints all or part of the specified document. The arguments for this method correspond to the options in the Print dialog box (File menu).

Syntax

print out application/document/window  Required. An expression that returns an application, document, or window object.

[background  Boolean] Optional. True to have the script continue while Word prints the document.

[append  Boolean] Optional. True to append the specified document to the file name specified by the output file name argument. False to overwrite the contents of output file name.

[print out range  enumeration] Optional. The page range. Can be one of the following: print all document, print current page, print from to, print range of pages, or print selection.

[output file name  Unicode text] Optional. If print to file is true, this argument specifies the path and file name of the output file.

[page from  integer] Optional. The starting page number when print out range is set to print from to.

[page to  integer] Optional. The ending page number when print out range is set to print from to.

[print out item  enumeration] Optional. The item to be printed. Can be one of the following: print auto text entries, print comments, print document content, print key assignments, print properties, print envelope, or print styles.

[print copies  integer] Optional. The number of copies to be printed.

[print out page type  enumeration] Optional. The type of pages to be printed. Can be one of the following: print all pages, print even pages only, or print odd pages only.

[print to file  Boolean] Optional. True to send printer instructions to a file. Make sure to specify a file name with output file name.

[collate  Boolean] Optional. When printing multiple copies of a document, true to print all pages of the document before printing the next copy.

[file name  Unicode text] Optional. The path and file name of the document to be printed. If this argument is omitted, Word prints the active document. Available only with the application object.


Example

This example prints the current page of the active document.

print out active document print out range print current page
This example prints the first three pages of the document in the active window.

This example prints the comments in the active document.

```
if (get count of Word comments of active document) ≥ 1 then
    print out active document print out item print comments
end if
```

**Command: print out envelope**

Prints an envelope without adding it to the active document.

**Syntax**

```
print out envelope  envelope  Required. An expression that returns an envelope object.

[extract address  Boolean] Optional. True to use the text marked by the EnvelopeAddress
    bookmark (a user-defined bookmark) as the recipient's address.

[address  Unicode text] Optional. A string that specifies the recipient's address (ignored if extract
    address is true).

[auto text  Unicode text] Optional. The name of the AutoText entry that includes a recipient's
    address.

[omit return address  Boolean] Optional. True to omit the return address.

[return address  Unicode text] Optional. A string that specifies the return address.

[return autotext  Unicode text] Optional. The name of the AutoText entry that includes a return
    address.

[print bar code  Boolean] Optional. True to add a POSTNET bar code. For U.S. mail only.

[print FIMA  Boolean] Optional. True to add a Facing Identification Mark (FIM-A) for use in
    presorting courtesy reply mail. For U.S. mail only.

[envelope size  Unicode text] Optional. A string that specifies the envelope size. The string
    should match one of the sizes listed on the Envelope size pop-up menu in the Custom Page
    Options dialog box (for example, "Size 10").

[envelope height  integer] Optional. The height of the envelope (in points) when the envelope
    size argument is set to "Custom size."

[envelope width  integer] Optional. The width of the envelope (in points) when the envelope
    size argument is set to "Custom size."

[feed source  Boolean] Optional. True to use the setting of the feed source property of the
    envelope object when printing the envelope.

[address from left  integer] Optional. The distance (in points) between the left edge of the
    envelope and the recipient's address.

[address from top  integer] Optional. The distance (in points) between the top edge of the
    envelope and the recipient's address.
```
[return address from left  integer]  Optional. The distance (in points) between the left edge of the envelope and the return address.
[return address from top  integer]  Optional. The distance (in points) between the top edge of the envelope and the return address.
[default face up  Boolean]  Optional. True to print the envelope face up, false to print it face down.
[default orientation  enumeration]  Optional. The orientation of the envelope. Can be one of the following: left portrait, center portrait, right portrait, left landscape, center landscape, right landscape, left clockwise, center clockwise, or right clockwise.
[size from page setup  Boolean]  Optional. True if the envelope's address areas are sized according to settings in the Page Setup dialog box (Envelopes command, Tools menu). False if they are sized according to custom settings. The default value is true.
[show page setup dialog  Boolean]  Optional. True if the Page Setup dialog box is displayed to allow adjustment of settings. Size from page setup must be set to true for the box to be displayed. The default value is false.

Example
This example prints an envelope using the user address as the return address and a predefined recipient address.

set recip to "Don Funk" & return & "123 Skye St." & return ~ & "Our Town, WA 98107"
print out envelope envelope object of active document address recip ~
  return address user address envelope size "Size 10" ~
  with print bar code

Command: print out mailing label

Prints a label or a page of labels with the same address.

Syntax

print out mailing label  mailing label  Required. An expression that returns a mailing label object.
[name  Unicode text]  Optional. The mailing label name.
[address  Unicode text]  Optional. The text for the label address.
[extract address  Boolean]  Optional. True to use the text marked by the EnvelopeAddress bookmark (a user-defined bookmark) as the label text. If this argument is specified, address is ignored.
[laser tray  enumeration]  Optional. The laser printer tray to be used. Can be one of the following: printer automatic sheet feed, printer default bin, printer envelope feed, printer form source, printer large capacity bin, printer large format bin, printer lower bin, printer manual envelope feed, printer manual feed, printer middle bin, printer only bin, printer paper cassette, printer small format bin, printer tractor feed, or printer upper bin.
[single label  Boolean]  Optional. True to print a single label, false to print an entire page of the same label.
Protects the specified document from changes. When a document is protected, the user can make only limited changes, such as adding annotations, making revisions, or completing a form.

**Note** If the document is already protected when you use this method, an error occurs.

**Syntax**

```plaintext
protect  document  Required. An expression that returns a document object.

protection type  enumeration  Required. The protection type for the specified document. Can be one of the following: allow only comments, allow only form fields, allow only revisions, or do document protection.

[no reset  Boolean] Optional. False to reset form fields to their default values. True to retain the current form field values if the specified document is protected. If protection type isn't allow only form fields, the no reset argument is ignored.

[password  Unicode text] Optional. The password required to "unprotect" the specified document.
```

**Example**

This example switches the active document to print preview if it's currently in some other view.

```plaintext
if do print preview is false then
   print preview active document
end if
```

**Command: print preview**

Toggles the view to or from print preview.

**Note** The do print preview property can be set to true or false to switch to or from print preview, respectively. You can also change the view by setting the view type property for the view object to print preview view.

**Syntax**

```plaintext
print preview  document  Required. An expression that returns a document object.
```

**Example**

This example prints a page of Avery 5664 mailing labels, using the specified address.

```plaintext
set addr to "Denise Smith" & return & "123 Skye St." & return ~
   & "OurTown, WA 98107"
print out mailing label mailing label object name "5664" address addr
```

**Command: protect**

Toggles the view to or from print preview.

**Note** The do print preview property can be set to true or false to switch to or from print preview, respectively. You can also change the view by setting the view type property for the view object to print preview view.

**Syntax**

```plaintext
print preview  document  Required. An expression that returns a document object.
```

**Example**

This example switches the active document to print preview if it's currently in some other view.

```plaintext
if do print preview is false then
   print preview active document
end if
```

**Command: protect**

Toggles the view to or from print preview.

**Note** The do print preview property can be set to true or false to switch to or from print preview, respectively. You can also change the view by setting the view type property for the view object to print preview view.

**Syntax**

```plaintext
print preview  document  Required. An expression that returns a document object.
```

This example prints a page of Avery 5664 mailing labels, using the specified address.

```plaintext
set addr to "Denise Smith" & return & "123 Skye St." & return ~
   & "OurTown, WA 98107"
print out mailing label mailing label object name "5664" address addr
```

**Command: protect**

Toggles the view to or from print preview.

**Note** The do print preview property can be set to true or false to switch to or from print preview, respectively. You can also change the view by setting the view type property for the view object to print preview view.

**Syntax**

```plaintext
print preview  document  Required. An expression that returns a document object.
```

This example switches the active document to print preview if it's currently in some other view.

```plaintext
if do print preview is false then
   print preview active document
end if
```
Example
This example protects the active document for forms without resetting the contents of the form fields.

if protection type of active document is no document protection then
    protect active document protection type allow only form fields
    with no reset
end if

This example protects Monthly Report.doc so that only comments can be added to it. The password "free" is required to unprotected the document.

set myDoc to document "Monthly Report.doc"
protect myDoc protection type allow only comments password "free"

Command: rebind

Changes the command assigned to the specified key binding.

Syntax

rebind  key binding   Required. An expression that returns a key binding object.

    key category   enumeration   Required. The key category of the specified key binding. Can be one of following:

    key category auto text
    key category command
    key category disable
    key category font
    key category macro
    key category nil
    key category prefix
    key category style
    key category symbol

    command   Unicode text   Required. The name of the specified command.

    [command parameter   Unicode text]   Optional. Additional text, if any, required for the command specified by command.

Example
This example reassigns the CTRL+SHIFT+S key binding to the FileSaveAs command.

set customization context to normal template
set myKey to find key key code (build key code key1 control_key
  key2 shift_key key3 s_key)
rebind myKey key category key category command command "FileSaveAs"

This example rebinds all keys assigned to the macro named "Macro1" to the macro named "ReportMacro."

set customization context to attached template of active document
repeat with myKey in (get keys bound to key category key category macro
  command "Macro1")
    rebind myKey key category key category macro command "ReportMacro"
end repeat
Command: redo
Redoes the last action that was undone (reverses the undo command). Returns true if the actions were redone successfully.

Syntax
redo document Required. An expression that returns a document object.
[times integer] Optional. The number of actions to be redone.

Example
This example redoes the last two actions in the Sales.doc redo list.
redo document "Sales.doc" times 2

Command: reject
Rejects the specified tracked change. The revision marks are removed, leaving the original text intact.

Note Formatting changes cannot be rejected.

Syntax
reject revision Required. An expression that returns a revision object.

Example
This example rejects the tracked changes in the first paragraph.
set myRange to text object of paragraph 1 of active document
repeat with myRev in (get revisions of myRange)
    reject myRev
end repeat
This example rejects the first tracked change in the selection.
set myRange to text object of selection
reject revision 1 of myRange

Command: reject all revisions
Rejects all tracked changes in the specified document.

Syntax
reject all revisions document Required. An expression that returns a document object.

Example
This example checks the main story in the active document for tracked changes, and if there are any, the example rejects all revisions in all stories in the document.
if (count of revisions of active document) ≥ 1 then ¬
    reject all revisions active document
**Command: reload**

Reloads a cached document by resolving the hyperlink to the document and downloading it.

**Note**  This command reloads the document asynchronously; that is, statements following the `reload` command in your procedure may execute before the document is actually reloaded. Because of this, you may get unexpected results from using this command in your scripts.

**Syntax**

`reload`  document  Required. An expression that returns a `document` object.

**Example**

This example opens and reloads the hyperlink to the address "main" on a local intranet.

```plaintext
follow hyperlink active document address "http://main"
reload active document
```

**Command: remove numbers**

Removes numbers or bullets from the specified `document`, `Word list`, or `list format` object.

**Syntax**

`remove numbers`  document/list format/Word list  Required. An expression that returns a `document`, `Word list`, or `list format` object.

`[number type enumeration]`  Optional. The type of number to be removed. Can be one of the following: `number paragraph`, `number listnum`, or `number all numbers`. The default value is `number all numbers`.

**Remarks**

When this command is applied to a `Word list` object, it removes numbers only from paragraphs in the specified list, skipping over any interleaved numbers from other lists. If this command is applied to the `list format` object for a range of text, all numbers from all lists in the range are removed.

**Example**

This example removes the bullets or numbers from any numbered paragraphs in the selection.

```plaintext
remove numbers list format of text object of selection
```

This example removes the numbers from the beginning of any numbered paragraphs in the active document.

```plaintext
remove numbers active document number type number paragraph
```

This example removes the LISTNUM fields from the selection.

```plaintext
remove numbers list format of text object of selection ¬
    number type number listnum
```

This example removes the bullets or numbers from the third list in MyDocument.doc.

```plaintext
if (count of Word lists of document "MyDocument.doc") ≥ 3 then
    remove numbers Word list 3 of document "MyDocument.doc"
end if
```
Command: remove theme
Removes the theme from a document.
Syntax
remove theme document Required. An expression that returns a document object.
Example
This example removes the active theme from the current document.
remove theme active document

Command: repaginate
Repaginates the entire document.
Syntax
repaginate document Required. An expression that returns a document object.
Example
This example repaginates the active document if it's changed since the last time it was saved.
if saved of active document is false then repaginate active document
This example repaginates all open documents.
repeat with aDoc in (get documents)
   repaginate aDoc
end repeat

Command: reset
Removes manual character formatting (formatting not applied using a style). For example, if you manually format a word as bold and the underlying style is plain text (not bold), the reset command removes the bold format.
Syntax
reset font Required. An expression that returns a font object.
Example
This example removes manual formatting from the selection.
reset font object of selection

Command: reset continuation notice
Resets the footnote or endnote continuation notice to the default notice. The default notice is blank (no text).
Syntax
reset continuation notice footnote options/endnote options Required. An expression that returns an endnote options or footnote options object.
Example
This example resets the endnote continuation notice for the active document.
reset continuation notice endnote options of active document
This example resets the footnote continuation notice and sets the starting number for footnote reference marks to 2 in Sales.doc.
reset continuation notice footnote options of text object of section 1 of document "Sales.doc"
set footnote numbering rule of footnote options of text object of section 1 of document "Sales.doc" to restart continuous
set footnote starting number of footnote options of text object of section 1 of document "Sales.doc" to 2

Command: reset continuation separator
Resets the footnote or endnote continuation separator to the default separator. The default separator is a long horizontal line that separates document text from notes continued from the previous page.

Syntax
reset continuation separator  footnote options/endnote options  Required. An expression that returns an endnote options or footnote options object.

Example
This example resets the footnote continuation separator to the default separator line.
reset continuation separator footnote options of active document
This example resets the endnote continuation separator for the first section in each open document.
repeat with aDoc in (get documents)
  reset continuation separator endnote options of text object of section 1 of aDoc
end repeat

Command: reset ignore all
Clears the list of words that were previously ignored during a spelling check. After you run this command, previously ignored words are checked along with all the other words.

Syntax
reset ignore all

Example
This example clears the list of words that were ignored during a previous spelling check and then begins a new spelling check on the active document.
reset ignore all
check spelling text object of active document
Command: reset list gallery

Resets the list template specified by index for the specified list gallery to the built-in list template format.

Syntax

reset list gallery  list gallery  Required. An expression that returns a list gallery object.

index  integer  Required. A number from 1 through 7, corresponding to a format in the specified list gallery. Skipping the None option, the formats are numbered from left to right, starting with the top row.

Example

This example sets the fourth format listed on the Numbered tab in the Bullets and Numbering dialog box back to the built-in numbering format, and then it applies the list template to the selection.

reset list gallery (get list gallery number gallery) index 4
apply list format template list format of text object of selection
    list template (list template 4 of (get list gallery number gallery))

Command: reset separator

Resets the footnote or endnote separator to the default separator. The default separator is a short horizontal line that separates document text from notes.

Syntax

reset separator  footnote options/endnote options  Required. An expression that returns an endnote options or footnote options object.

Example

This example resets the footnote separator to the default separator line.
reset separator footnote options of active document

This example resets the endnote separator for the notes in the document where the selection is located.
reset separator endnote options of selection
Command: retrieve language

Returns the language object for the specified language

Syntax

retrieve language  enumeration  Required. The language whose language object you want to return. Can be one of the following:

<table>
<thead>
<tr>
<th>Language</th>
<th>Language</th>
<th>Language</th>
</tr>
</thead>
<tbody>
<tr>
<td>afrikaans</td>
<td>french</td>
<td>russian</td>
</tr>
<tr>
<td>arabic</td>
<td>french canadian</td>
<td>serbian cyrillic</td>
</tr>
<tr>
<td>basque</td>
<td>german</td>
<td>serbian latin</td>
</tr>
<tr>
<td>belgian dutch</td>
<td>greek</td>
<td>sesotho</td>
</tr>
<tr>
<td>belgian french</td>
<td>hebrew</td>
<td>simplified chinese</td>
</tr>
<tr>
<td>brazilian portuguese</td>
<td>hungarian</td>
<td>slovak</td>
</tr>
<tr>
<td>bulgarian</td>
<td>italian</td>
<td>slovenian</td>
</tr>
<tr>
<td>byelorussian</td>
<td>icelandic</td>
<td>spanish</td>
</tr>
<tr>
<td>catalan</td>
<td>japanese</td>
<td>spanish modern sort</td>
</tr>
<tr>
<td>croatian</td>
<td>korean</td>
<td>swedish</td>
</tr>
<tr>
<td>czech</td>
<td>language none</td>
<td>swiss french</td>
</tr>
<tr>
<td>danish</td>
<td>latvian</td>
<td>swiss german</td>
</tr>
<tr>
<td>dutch</td>
<td>macedonian</td>
<td>swiss italian</td>
</tr>
<tr>
<td>english aus</td>
<td>malaysian</td>
<td>traditional chinese</td>
</tr>
<tr>
<td>english canadian</td>
<td>mexicanSpanish</td>
<td>tsonga</td>
</tr>
<tr>
<td>english new zealand</td>
<td>language no proofing</td>
<td>tswana</td>
</tr>
<tr>
<td>english south africa</td>
<td>norwegian bokmol</td>
<td>turkish</td>
</tr>
<tr>
<td>english uk</td>
<td>norwegian nynorsk</td>
<td>ukrainian</td>
</tr>
<tr>
<td>english us</td>
<td>polish</td>
<td>venda</td>
</tr>
<tr>
<td>estonian</td>
<td>portuguese</td>
<td>xhosa</td>
</tr>
<tr>
<td>farsi</td>
<td>romanian</td>
<td>zulu</td>
</tr>
</tbody>
</table>

Command: run auto macro

Runs an auto macro that's stored in the specified document. If the specified auto macro doesn't exist, nothing happens.

Note  Use the run VB macro command to run any macro.

Syntax

run auto macro  document  Required. An expression that returns a document object.

    which auto macro  enumeration  Required. The auto macro to run. Can be one of the following:
    auto close, auto exec, auto exit, auto new, or auto open.

Example

This example runs the AutoOpen macro in the active document.

run auto macro active document which auto macro auto open
Command: run letter wizard

Runs the Letter Wizard on the specified document.

Syntax

run letter wizard  document  Required. An expression that returns a document object.


[wizard mode  Boolean]  Optional. True to display the Letter Wizard dialog box as a series of steps with a Next, Back, and Finish button. False to display the Letter Wizard dialog box as if it were opened from the Tools menu (a properties dialog box with an OK button and a Cancel button). The default value is false.

Example

This example starts the Letter Wizard for the active document.

run letter wizard active document ¬

letter content (letter content of active document)

Command: run VB macro

Runs a Visual Basic macro.

Note  Only public Sub procedures that take no arguments, which includes all procedures generated by the macro recorder and all procedures you can run from the Macros dialog box, can be run by using the run VB macro command.

Syntax

run vb macro

macro name  Unicode text  Required. The name of the macro. Can be any combination of template, module, and macro name. For example, the following statements are all valid.

run VB macro macro name "Normal.Module1.MAIN"
run VB macro macro name "MyProject.MyModule.MyProcedure"
run VB macro macro name "'My Document.doc'!ThisModule.ThisProcedure"

If you specify the document name, your code can only run macros in documents related to the current context, not just any macro in any document.

You cannot pass parameters to a procedure by using the run VB macro command.

Example

This example prompts the user to enter a macro name, and then it runs that macro.

set mn to text returned of (display dialog "Enter the macro name" ¬

default answer ")
run VB macro macro name mn
Command: save as

Saves the specified document with a new name or format. The arguments for this method correspond to the options in the Save As dialog box (File menu).

Syntax

save as  document  Required. An expression that returns a document object.

[file name]  Unicode text] Optional. The name for the document. The default is the current folder and file name. If the document has never been saved, the default name is used (for example, Doc1.doc). If a document with the specified file name already exists, the document is overwritten without the user being prompted first.

[file format] enumeration] Optional. The format in which the document is saved. Can be one of the following: format custom dictionary, format document, format dostext, format dostext line breaks, format exclude dictionary, format rtf, wdFormatStationery, format template, format text, format text line breaks, format stationery, format web archive, format HTML, or format unicode text. To save a document in another format, specify the appropriate value for the save format property of the file converter object.

[lock comments] Boolean] Optional. True to lock the document for comments.


[add to recent files] Boolean] Optional. True to add the document to the list of recently used files on the File menu.


[read only recommended] Boolean] Optional. True to have Word suggest read-only status whenever the document is opened.

[embed true type fonts] Boolean] Optional. True to save TrueType fonts with the document.

[save native picture format] Boolean] Optional. If graphics were imported from another platform, true to save only the Windows version of the imported graphics.

[save forms data] Boolean] Optional. True to save the data entered by a user in a form as a data record.

[HTML display only output] Boolean] Optional. True to save only file information that can be displayed in HTML in a Web browser. Any document options that can’t be displayed in a Web browser will be lost.

Example

This example saves the active document as Test.rtf in rich-text format (RTF).

save as active document file name "Test.rtf" file format format rtf

This example saves MyDoc.doc with a write password and then closes the document.

save as document "MyDoc.doc" write password "pass"

close document "MyDoc.doc"
**Command: save version**

Saves a version of the specified document with a comment.

**Syntax**

\[
\text{save version} \quad \text{document} \quad \text{Required. An expression that returns a document object.}
\]

\[
\text{comment} \quad \text{Unicode text} \quad \text{Required. The comment string that's saved with the version.}
\]

**Example**

This example saves a version of Sales.doc with a comment.

```
save version document "Sales.doc" comment "Minor changes to intro"
```

**Command: screen refresh**

Updates the display on the monitor with the current information in the video memory buffer. You can use this command after using the screen updating property to disable screen updates.

**Syntax**

```
screen refresh
```

**Remarks**

**Screen refresh** turns on screen updating for just one instruction and then immediately turns it off. Subsequent instructions don't update the screen until screen updating is turned on again with the screen updating property of the application class.

Document windows are hidden when screen updating is turned off. The screen refresh command displays updated document windows for an instant, but then they're hidden again. Therefore, you may want to use the screen updating property instead of the screen refresh command to turn screen updating on and off.

**Example**

This example turns off screen updating, opens Test.doc, inserts text, refreshes the screen, and then closes the document (with changes saved).

```
set screen updating to false
open "Macintosh HD:Users:Shared:TEST.DOC"
set myRange to create range active document start 0 end 0
insert text "new" at myRange
screen refresh
close active document saving no
set screen updating to true
```

**Command: select cell**

Selects the entire cell containing the current selection. To use this command, the current selection must be contained within a single cell.

**Syntax**

```
select cell \quad \text{selection object} \quad \text{Required. An expression that returns a selection object.}
```
Example
This example selects the entire cell containing the current selection.
select cell selection

Command: select column
Selects the column that contains the insertion point, or selects all columns that contain the selection. If the selection isn't in a table, an error occurs.

Syntax
select column selection object Required. An expression that returns a selection object.

Example
This example collapses the selection to the ending point and then selects the column that contains the insertion point.
collapse range text object of selection direction collapse end
if (get selection information selection information type with in table) ~
   is "True" then
   select column selection
end if

Command: select current alignment
Extends the selection forward until text with a different paragraph alignment is encountered.

Syntax
select current alignment selection object Required. An expression that returns a selection object.

Remarks
There are four types of paragraph alignment: left, centered, right, and justified.

Example
This example positions the insertion point at the beginning of the first paragraph after the current paragraph that doesn't have the same alignment as the current paragraph. If the alignment is the same from the selection to the end of the document, the example moves the selection to the end of the document and displays a message.
select current alignment selection
collapse range text object of selection direction collapse end
if (end of content of text object of selection) = ((end of content of text object of active document) - 1) then
display dialog "No change in alignment found"
end if
Command: select current color

Extends the selection forward until text with a different color is encountered.

Syntax

select current color selection object Required. An expression that returns a selection object.

Example

This example extends the selection from the beginning of the document to the first character formatted with a different color and then displays the number of characters in the resulting selection.

home key selection move unit a story extend by moving
select current color selection
set n to count of characters of selection
display dialog "Contiguous characters with the same color: " & n

Command: select current font

Extends the selection forward until text in a different font or font size is encountered.

Syntax

select current font selection object Required. An expression that returns a selection object.

Example

This example extends the selection until text in a different font or font size is encountered. The example uses the grow font command to increase the size of the selected text to the next available font size.

select current font selection
grow font font object of selection

Command: select current indent

Extends the selection forward until text with different left or right paragraph indents is encountered.

Syntax

select current indent selection object Required. An expression that returns a selection object.

Example

This example jumps to the beginning of the first paragraph in the document that has different indents than the first paragraph in the active document.

home key move unit a story extend by moving
select current indent selection
collapse range text object of selection direction collapse end
This example determines whether all the paragraphs in the active document are formatted with the same left and right indents and then displays a message box indicating the result.

```plaintext
home key move unit a story extend by moving
select current indent selection
collapse range text object of selection direction collapse end
if (end of content of text object of selection) = ((end of content of text object of active document) - 1) then
    display dialog "All paragraphs share the same left and right indents"
else
    display dialog "Not all paragraphs share the same left and right indents"
end if
```

**Command: select current spacing**

Extends the selection forward until a paragraph with different line spacing is encountered.

**Syntax**

```plaintext
select current spacing  selection object  Required. An expression that returns a selection object.
```

**Example**

This example selects all consecutive paragraphs that have the same line spacing and changes the line spacing to single spacing.

```plaintext
select current spacing selection
space 1 paragraph format of selection
```

**Command: select current tabs**

Extends the selection forward until a paragraph with different tab stops is encountered.

**Syntax**

```plaintext
select current tabs  selection object  Required. An expression that returns a selection object.
```

**Example**

This example selects the second paragraph in the active document and then extends the selection to include all other paragraphs that have the same tab stops.

```plaintext
set myRange to text object of paragraph 2 of active document
select myRange
select current tabs selection
```
This example selects paragraphs that have the same tab stops and retrieves the position of the first tab stop. The example moves the selection to the next range of paragraphs that have the same tab stops. The example then adds the tab stop setting from the first group of paragraphs to the current selection.

```
select current tabs selection
set pos to tab stop position of tab stop 1 of paragraph format of selection
collapse range text object of selection direction collapse end
select current tabs selection
make new tab stop at paragraph format of selection with properties ¬
    {tab stop position:pos}
```

**Command: select row**

Selects the row that contains the insertion point, or selects all rows that contain the selection. If the selection isn't in a table, an error occurs.

**Syntax**

```
select row  selection object  Required. An expression that returns a selection object.
```

**Example**

This example collapses the selection to the starting point and then selects the row that contains the insertion point.

```
collapse range text object of selection direction collapse start
if (get selection information selection information type with in table) ¬
    is "True" then
    select row selection
end if
```

**Command: send mail**

Opens a message window for sending the specified document through an electronic mail system.

**Note** Use the send mail attach property of the Word options class to control whether the document is sent as text in the message window or as an attachment.

**Syntax**

```
send mail  document  Required. An expression that returns a document object.
```

**Example**

This example sends the active document as an attachment to a mail message.

```
set send mail attach of settings to true
send mail active document
```
Command: set active writing style

Sets the writing style for a specified language in the specified document.

Note The writing style list property of the language class returns an array of the names of the available writing styles.

Syntax

set active writing style document Required. An expression that returns a document object.

language ID enumeration Required. The language you want to set the writing style for in the specified document. Can be one of the following:

<table>
<thead>
<tr>
<th>Language</th>
<th>Writing Style</th>
<th>Language</th>
<th>Writing Style</th>
</tr>
</thead>
<tbody>
<tr>
<td>afrikaans</td>
<td>french</td>
<td>russian</td>
<td></td>
</tr>
<tr>
<td>arabic</td>
<td>french canadian</td>
<td>serbian cyrillic</td>
<td></td>
</tr>
<tr>
<td>basque</td>
<td>german</td>
<td>serbian latin</td>
<td></td>
</tr>
<tr>
<td>belgian dutch</td>
<td>greek</td>
<td>sesotho</td>
<td></td>
</tr>
<tr>
<td>belgian french</td>
<td>hebrew</td>
<td>simplified chinese</td>
<td></td>
</tr>
<tr>
<td>brazilian portuguese</td>
<td>hungarian</td>
<td>slovak</td>
<td></td>
</tr>
<tr>
<td>bulgarian</td>
<td>italian</td>
<td>slovenian</td>
<td></td>
</tr>
<tr>
<td>byelorussian</td>
<td>icelandic</td>
<td>spanish</td>
<td></td>
</tr>
<tr>
<td>catalan</td>
<td>japanese</td>
<td>spanish modern sort</td>
<td></td>
</tr>
<tr>
<td>croatian</td>
<td>korean</td>
<td>swedish</td>
<td></td>
</tr>
<tr>
<td>czech</td>
<td>language none</td>
<td>swiss french</td>
<td></td>
</tr>
<tr>
<td>danish</td>
<td>latvian</td>
<td>swiss german</td>
<td></td>
</tr>
<tr>
<td>dutch</td>
<td>macedonian</td>
<td>swiss italian</td>
<td></td>
</tr>
<tr>
<td>english aus</td>
<td>malaysian</td>
<td>traditional chinese</td>
<td></td>
</tr>
<tr>
<td>english canadian</td>
<td>mexicanSpanish</td>
<td>tsonga</td>
<td></td>
</tr>
<tr>
<td>english new zealand</td>
<td>language no proofing</td>
<td>tswana</td>
<td></td>
</tr>
<tr>
<td>english south africa</td>
<td>norwegian bokmol</td>
<td>turkish</td>
<td></td>
</tr>
<tr>
<td>english uk</td>
<td>norwegian nynorsk</td>
<td>ukrainian</td>
<td></td>
</tr>
<tr>
<td>english us</td>
<td>polish</td>
<td>venda</td>
<td></td>
</tr>
<tr>
<td>estonian</td>
<td>portuguese</td>
<td>xhosa</td>
<td></td>
</tr>
<tr>
<td>farsi</td>
<td>romanian</td>
<td>zulu</td>
<td></td>
</tr>
</tbody>
</table>

writing style Unicode text. Required. The writing style you want to set the document to.

Example

This example sets the writing style used for French, German, and U.S. English for the active document. You must have the grammar files installed for French, German, and U.S. English to run this example.

set active writing style active document language ID english us ¬
    writing style "Technical"

set active writing style active document language ID french ¬
    writing style "Commercial"

set active writing style active document language ID german ¬
    writing style "Grammatik"
Command: set all fuzzy options

Activates all nonspecific search options associated with Japanese text.

Syntax

set all fuzzy options  find  Required. An expression that returns a find object.

Remarks

This command sets the following properties to true:

- ay match fuzzy
- bv match fuzzy
- byte match fuzzy
- case match fuzzy
- dash match fuzzy
- dz match fuzzy
- hf match fuzzy
- hiragana match fuzzy
- iteration mark match fuzzy
- kanji match fuzzy
- ki ku match fuzzy
- old kana match fuzzy
- prolonged sound mark match fuzzy
- punctuation match fuzzy
- small kana match fuzzy
- space match fuzzy
- tc match fuzzy
- zj match fuzzy

Example

This example activates all nonspecific options before executing a search in the selected range. If the word "حماية" is formatted as bold, the entire paragraph is selected and copied to the Clipboard.

clear formatting find of selection
set all fuzzy options find of selection
set bold of font object of find object of selection to true
set utxtViolin to <data utxt30D030A430AA30EA30F3>
execute find find object of selection find text utxtViolin ~
    with match forward and find format
if found of find of selection is true then
    expand selection by a paragraph item
    copy selection
end if

Command: set as font template default

Sets the specified font formatting as the default for the active document and all new documents based on the active template. The default font formatting is stored in the Normal style.

Syntax

set as font template default  font  Required. An expression that returns a font object.

Example

This example sets the character formatting in the selection as the default.
set as font template default font object of selection

332
Command: set as page setup template default

Sets the specified page setup formatting as the default for the active document and all new documents based on the active template.

Syntax

set as page setup template default  page setup  Required. An expression that returns a page setup object.

Example

This example changes the left and right margin settings for the active document and then sets the page setup formatting as the default.

set left margin of page setup of active document to inches to points inches 1
set right margin of page setup of active document to ~
   inches to points inches 1
set as page setup template default page setup of active document

Command: set default file path

Returns or sets default folders for items such as documents, templates, and graphics.

Syntax

set default file path

file path type  enumeration  Required. The default folder to set. Can be one of the following:
documents path, pictures path, user templates path, workgroup templates path, user options path, auto recover path, tools path, tutorial path, startup path, program path, graphics filters path, text converters path, proofing tools path, temp file path, current folder path, style gallery path, trash path, office path, type libraries path, or border art path.

path  Unicode text  Required. The new file path.

Remarks

The new setting takes effect immediately.

Example

This example sets the default folder for Word documents.

set default file path file path type documents path ~
   path "Macintosh HD:Users:Shared"
Command: set default theme

Sets a default theme for Word to use with new documents, e-mail messages, or Web pages.

Syntax

set default theme

name  Unicode text  Required. The name of the theme you want to assign as the default theme plus any theme formatting options you want to apply. The format of this string is "theme nnn" where theme and nnn are defined as follows:

<table>
<thead>
<tr>
<th>String</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>theme</td>
<td>The name of the folder that contains the data for the requested theme. (The default location for theme data folders is &quot;Microsoft Office 2004:Office:Themes.&quot;) You must use the folder name for the theme rather than the display name that appears in the Theme dialog box (Theme command, Format menu).</td>
</tr>
<tr>
<td>nnn</td>
<td>A three-digit string that indicates which theme formatting options to activate (1 to activate, 0 to deactivate). The digits correspond to the Vivid Colors, Active Graphics, and Background Image check boxes in the Theme dialog box (Theme command, Format menu). If this string is omitted, the default value for nnn is &quot;011&quot; (Active Graphics and Background Image are activated).</td>
</tr>
</tbody>
</table>

theme type  enumeration  Required. The type of new document to which you are assigning a default theme. Can be one of the following: for a document, for a email message, or for a web page.

Remarks

Setting a default theme will not apply that theme to the blank document automatically created when you start Word. Any new documents you create after that will have the default theme.

Example

This example specifies that Word use the Expedition theme with Active Graphics for all new Web pages.

set default theme name "expeditn 010" theme type for a web page
Command: set document compatibility

True if the compatibility option specified by the type argument is enabled. Compatibility options affect how a document is displayed in Word. These options are listed in the Compatibility pane of the Preferences dialog box (Word menu).

Syntax

```
compatibility item  enumeration  Required. Can be one of the following:
```

<table>
<thead>
<tr>
<th>Constant</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>align tables row by row</td>
<td>True if Word aligns table rows independently.</td>
</tr>
<tr>
<td>Word95 auto space</td>
<td>True if Word auto spaces like Word 95.</td>
</tr>
<tr>
<td>convert data merge escapes</td>
<td>True if Word correctly interprets characters preceded by backslashes () in Word version 2.x mail-merge data sources. For example, &quot; is interpreted as &quot;.</td>
</tr>
<tr>
<td>do not adjust line height in table</td>
<td>True if Word doesn't adjust line height to grid height in tables.</td>
</tr>
<tr>
<td>do not balance SBCS and DBCS characters</td>
<td>True if Word doesn't balance the width of single-byte and double-byte characters.</td>
</tr>
<tr>
<td>do not underline trailing spaces</td>
<td>True if Word doesn't draw underlines on trailing spaces.</td>
</tr>
<tr>
<td>do not use html paragraph autospacing</td>
<td>True if Word doesn't use HTML paragraph auto spacing.</td>
</tr>
<tr>
<td>exact on top</td>
<td>True if Word places text at the top of the space allocated for the line when using exact line spacing.</td>
</tr>
<tr>
<td>expand shift return</td>
<td>True if Word expands character spaces on required line breaks (SHIFT+RETURN).</td>
</tr>
<tr>
<td>Word98 footnote layout</td>
<td>True if Word lays out footnotes like Word 6.x/95/97.</td>
</tr>
<tr>
<td>forget last tab alignment</td>
<td>True if Word forgets the last tab alignment when adding a new tab.</td>
</tr>
<tr>
<td>layout raw table width</td>
<td>True if Word lays out tables with their raw width.</td>
</tr>
<tr>
<td>layout table rows apart</td>
<td>True if Word allows table rows to lay out apart.</td>
</tr>
<tr>
<td>leave backslash alone</td>
<td>True if Word doesn't convert backslash characters into yen signs.</td>
</tr>
<tr>
<td>Constant</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Word6 line wrap</td>
<td>True if Word wraps lines the same way Word 6.0 wraps lines.</td>
</tr>
<tr>
<td>use larger small caps</td>
<td>True if Word applies small capital letter (“small caps”) formatting as in Word version 5.x for the Macintosh, which produces slightly larger small caps.</td>
</tr>
<tr>
<td>no column balance</td>
<td>True if Word doesn’t balance text columns above continuous section breaks.</td>
</tr>
<tr>
<td>no extra line spacing</td>
<td>True if Word handles line spacing as in WordPerfect version 5.x. This argument is available only in Word version 7.0.</td>
</tr>
<tr>
<td>no extra spacing between rows of text</td>
<td>True if Word displays lines of text without leading as in Word version 5.x for the Macintosh.</td>
</tr>
<tr>
<td>no space for underlines</td>
<td>True if Word doesn’t add extra space for underlines. This argument is available only in Word version 7.0.</td>
</tr>
<tr>
<td>no space for raised or lowered characters</td>
<td>True if Word doesn’t add extra line spacing for raised and lowered characters.</td>
</tr>
<tr>
<td>no tab hanging indent</td>
<td>True if Word doesn’t automatically add a tab stop to a paragraph formatted with a hanging indent.</td>
</tr>
<tr>
<td>original word table rules</td>
<td>True if Word combines table borders as in Word version 5.x for the Macintosh.</td>
</tr>
<tr>
<td>print body text before header</td>
<td>True if Word prints the main text layer before the header and footer layer (the reverse of the default order). This allows Word to process PostScript codes in the main text layer the same as in Word version 5.x for the Macintosh.</td>
</tr>
<tr>
<td>print colors black</td>
<td>True if Word prints colors as black on printers that don’t support color.</td>
</tr>
<tr>
<td>Word96 shape layout</td>
<td>True if Word lays out AutoShapes like Word 97.</td>
</tr>
<tr>
<td>show breaks in frames</td>
<td>True if Word displays manual (“hard”) page or column breaks in any frames that contain them.</td>
</tr>
<tr>
<td>spacing in whole points</td>
<td>True if Word rounds character-spacing measurements up or down to the nearest whole number.</td>
</tr>
<tr>
<td>substitute font by size</td>
<td>True if Word substitutes fonts based on the font size in WordPerfect version 6.0 documents. This argument is available only in Word version 7.0.</td>
</tr>
<tr>
<td>suppress bottom spacing</td>
<td>True if Word removes extra line spacing at the bottom of the page.</td>
</tr>
<tr>
<td>Constant</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>suppress space before after page break</td>
<td>True if Word removes space before or after hard page breaks and column breaks.</td>
</tr>
<tr>
<td>suppress top spacing</td>
<td>True if Word removes extra line spacing at the top of the page.</td>
</tr>
<tr>
<td>suppress top spacing Mac Word5</td>
<td>True if Word handles extra line spacing at the top of the page the same as in Word version 5.x for the Macintosh.</td>
</tr>
<tr>
<td>swap borders facing pages</td>
<td>True if Word prints a left paragraph border (not a box) on the right side on odd-numbered pages if either the Different odd and even check box (Layout tab) or the Mirror margins check box (Margins tab) is selected in the Page Setup dialog box (File menu).</td>
</tr>
<tr>
<td>transparent metafiles</td>
<td>True if Word doesn’t “blank” the area behind metafile pictures.</td>
</tr>
<tr>
<td>truncate font height</td>
<td>True if Word rounds the font size up or down as in WordPerfect version 6.x for Windows. This argument is available only in Word version 7.0.</td>
</tr>
<tr>
<td>use printer metrics</td>
<td>True if Word uses printer metrics to lay out the document.</td>
</tr>
<tr>
<td>use Word97 line breaking rules</td>
<td>True if Word uses Word 97 line-breaking rules for Asian text.</td>
</tr>
<tr>
<td>WordPerfect justification</td>
<td>True if Word lays out fully justified text the same as in WordPerfect version 6.x.</td>
</tr>
<tr>
<td>WordPerfect space width</td>
<td>True if Word sets the default width of spaces between words to be the same as in WordPerfect version 5.x for Windows and WordPerfect version 6.0 for DOS.</td>
</tr>
<tr>
<td>wrap trail spaces</td>
<td>True if Word wraps spaces at the end of lines to the next line.</td>
</tr>
<tr>
<td>Word6 border rules</td>
<td>True if Word handles left and right borders of text lines affected by frames the same as in Word version 6.x.</td>
</tr>
</tbody>
</table>

**is compatible**  
Boolean  Required. The value to be set.

**Example**

This example enables the Suppress Space Before after a hard page or column break option in the Compatibility pane of the Preferences dialog box (Word menu) for the active document.

```
set document compatibility active document compatibility item suppress space before after page break with is compatible
```
Microsoft Word Suite

Command: set number to text columns

Arranges text into the specified number of text columns.

Note  You can also use the make command with the text column class to add text columns to a document.

Syntax

set number of text columns  page setup  Required. An expression that returns a page setup object.

   number of columns  integer  Required. The number of columns the text is to be arranged into.

Example

This example arranges the text in the active document into two columns of equal width.

set number of text columns page setup of active document number of columns 2

Command: set private profile string

Sets a string in a settings file.

Syntax

set private profile string  system object  Required. An expression that returns a system object.

   file name  Unicode text  Required. The file name for the settings file. If there's no path specified, the Users/<user name>/Library/Preferences/Microsoft folder is assumed.

   section  Unicode text  Required. The name of the section in the settings file that contains key.

   key  Unicode text  Required. The key whose setting you want to retrieve. Key should be the value in the key specified by section (for example, "Tools-Path").

   private profile string  Unicode text  Required. The string to be set.

Command: set profile string

Sets a setting in the Word Settings (10) file.

Syntax

set profile string  system object  Required. An expression that returns a system object.

   section  Unicode text  Required. A key in the settings file.

   key  Unicode text  Required. The value in the key specified by section (for example, "BackgroundPrint" in the Options key).

   profile string  Unicode text  The string to be set.
Command: show
Displays and carries out actions initiated in the specified built-in Word dialog box. Returns an integer that indicates which button was clicked to close the dialog box.

<table>
<thead>
<tr>
<th>Return value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-2</td>
<td>The Close button.</td>
</tr>
<tr>
<td>-1</td>
<td>The OK button.</td>
</tr>
<tr>
<td>0 (zero)</td>
<td>The Cancel button.</td>
</tr>
<tr>
<td>&gt; 0 (zero)</td>
<td>A command button: 1 is the first button, 2 is the second button, and so on.</td>
</tr>
</tbody>
</table>

**Note** To display a dialog box but not have any actions carried out or settings applied when the dialog box is closed, use the **display Word dialog** command.

**Syntax**

show dialog  Required. An expression that returns a dialog object.

[time out integer]  Optional. The amount of time that Word will wait before closing the dialog box automatically. One unit is approximately 0.001 second. Concurrent system activity may increase the effective time value. If this argument is omitted, the dialog box is closed when the user dismisses it.

**Example**

This example displays and carries out any action initiated in the Zoom dialog box. If there are no actions initiated for approximately 9 seconds, the dialog box is closed.

show (get dialog dialog view zoom) time out 9000

**Command: show all headings**

Toggles between showing all text (headings and body text) and showing only headings.

**Note** This method generates an error if the view isn't outline view or master document view.

**Syntax**

show all headings  view  Required. An expression that returns a view object.

**Example**

This example uses the **show heading** command to show all headings (without any body text) and then toggles the display to show all text (headings and body text) in outline view.

set view type of view of active window to outline view
show heading view of active window level 9
show all headings view of active window
Command: show heading

Shows all headings up to the specified heading level and hides subordinate headings and body text.

Note This command generates an error if the view isn't outline view or master document view.

Syntax

show heading view Required. An expression that returns a view object.

   level integer Required. The outline heading level (a number from 1 to 9).

Example

This example switches the active window to outline view and displays all text that's formatted with the Heading 1 style. Body text and all other types of headings are hidden.

set view type of view of active window to outline view
show heading view of active window level 1

This example switches the window for Document1 to outline view and displays all text that's formatted with the Heading 1, Heading 2, or Heading 3 style.

set view type of view of window "Document1" to outline view
show heading view of window "Document1" level 3

Command: show me

Displays the Office Assistant or the Help window when there's more information available. If additional information isn't available, this command generates a message that no associated Help topic exists.

Syntax

show me

Example

This example completes a TipWizard Show Me action if one is available.

show me

Command: shrink discontiguous selection

Deselects all but the most recently selected text when a selection contains multiple, unconnected selections.

Syntax

shrink discontiguous selection selection Required. An expression that returns a selection object.

Example

This example deselects all but the most recently selected text and formats with bold and small caps the text remaining in the selection. This example assumes there are multiple selections in the document.

shrink discontiguous selection selection
set bold of font object of selection to true
set small caps of font object of selection to true
Command: shrink font
Decreases the font size to the next available size. If the selection or range contains more than one
font size, each size is decreased to the next available setting.

Syntax
`shrink font` font Required. An expression that returns a font object.

Example
This example reduces the font size of the selected text by one size.

```plaintext
if selection type of selection is selection normal then
    shrink font font object of selection
else
    "You need to select some text."
end if
```

Command: shrink selection
Shrinks the selection to the next smaller unit of text. The progression is as follows: entire document,
section, paragraph, sentence, word, insertion point.

Syntax
`shrink selection` selection object Required. An expression that returns a selection object.

Example
This example shrinks the current selection by the next smaller unit.

```
shrink selection selection
```

Command: small scroll
Scrolls a window or pane by the specified number of lines. This command is equivalent to clicking
the scroll arrows on the horizontal and vertical scroll bars.

Syntax
`small scroll` window/pane Required. An expression that returns a pane or window object.

```
[down integer] Optional. The number of lines to scroll the window down. A "line" corresponds
to the distance scrolled by clicking the down scroll arrow on the vertical scroll bar once.

[up integer] Optional. The number of lines to scroll the window up. A "line" corresponds to the
distance scrolled by clicking the up scroll arrow on the vertical scroll bar once.

[to right integer] Optional. The number of lines to scroll the window to the right. A "line"
corresponds to the distance scrolled by clicking the right scroll arrow on the horizontal scroll
bar once.

[to left integer] Optional. The number of lines to scroll the window to the left. A "line"
corresponds to the distance scrolled by clicking the left scroll arrow on the horizontal scroll bar
once.
```
Remarks
If down and up are both specified, the window is scrolled by the difference of the arguments. For example, if down is 3 and up is 6, the window is scrolled up three lines. Similarly, if to left and to right are both specified, the window is scrolled by the difference of the arguments.

Any of these arguments can be a negative number. If no arguments are specified, the window is scrolled down by one line.

Example
This example scrolls the active window down one line.
small scroll active window down 1
This example splits the active window and then scrolls up and over to the left.
set split window of active window to true
small scroll active window up 5 to left 5

Command: speak text

Uses Apple's Text-to-Speech engine to speak aloud the selection.

Syntax
speak text  selection object  Required. An expression that returns a selection object.

Example
This example selects all of the text in the document and reads it aloud.
select text object of active document
speak text selection

Command: split subdocument

Divides an existing subdocument into two subdocuments at the same level in master document view or outline view. The division is at the beginning of the specified range. If the active document isn't in either master document or outline view, or if the range isn't at the beginning of a paragraph in a subdocument, an error occurs.

Syntax
split subdocument  subdocument  Required. An expression that returns a subdocument object.

text range  text range  Required text range object. The range that, when the subdocument is split, becomes a separate subdocument.

Example
This example splits the selection from an existing subdocument into a separate subdocument.
split subdocument subdocument 1 of text object of selection ¬
text range text object of selection
**Command: split table in selection**

Inserts an empty paragraph above the first row in the selection. If the selection isn't in the first row of the table, the table is split into two tables.

**Note** If the selection isn't in a table, an error occurs.

**Syntax**

```plaintext
split table in selection  selection object  Required. An expression that returns a selection object.
```

**Example**

If the selection is in a table, this example splits the table.

```plaintext
if (get selection information selection information type with in table) is "True" then
    split table in selection selection
end if
```

This example splits the first table in the active document between the first and second rows.

```plaintext
select (row 2 of table 1 of active document)

split table in selection selection
```

---

**Command: substitute font**

Sets font-mapping options, which are reflected in the Font Substitution dialog box (Compatibility pane, Preferences dialog box, Word menu).

**Syntax**

```plaintext
substitute font  unavailable font  Required. The name of a font not available on your computer that you want to map to a different font for display and printing.

substitute font  substitute font  Required. The name of a font available on your computer that you want to substitute for the unavailable font.
```

**Example**

This example substitutes Courier for Myfont.

```plaintext
substitute font unavailable font "Myfont" substitute font "Courier"
```

---

**Command: swap with endnotes**

Converts all footnotes in a document to endnotes and vice versa.

**Note** To convert a range of footnotes to endnotes, use the footnote convert command.

**Syntax**

```plaintext
swap with endnotes  footnote options  Required. An expression that returns a footnote options object.
```
**Command: swap with footnotes**

Converts all endnotes in a document to footnotes and vice versa.

**Note**  To convert a range of endnotes to footnotes, use the `endnote convert` command.

**Syntax**

`swap with footnotes`  endnote options  Required. An expression that returns an `endnote options` object.

**Command: toggle portrait**

Switches between portrait and landscape page orientations for a document or section.

**Syntax**

`toggle portrait`  page setup  Required. An expression that returns a `page setup` object.

**Remarks**

If the specified sections have different page orientations, an error occurs.

**Example**

This example changes the page orientation for the active document.

```
toggle portrait page setup of active document
```

This example changes the page orientation for all the sections in the selection. If the initial orientation of each section is not the same as the orientation of the other sections, an error occurs.

```
toggle portrait page setup of selection
```

**Command: type backspace**

Deletes the character preceding a collapsed selection (an insertion point). If the selection isn't collapsed to an insertion point, the selection is deleted.

**Note**  This method corresponds to the functionality of the BACKSPACE key.

**Syntax**

`type backspace`  selection object  Required. An expression that returns a `selection` object.

**Example**

This example deletes the character preceding the insertion point (the collapsed selection).

```
collapse range text object of selection direction collapse end type backspace selection
```

**Command: type paragraph**

Inserts a new, blank paragraph. If the selection isn't collapsed to an insertion point, it's replaced by the new paragraph.

**Note**  This method corresponds to the functionality of the ENTER key.

**Syntax**

`type paragraph`  selection object  Required. An expression that returns a `selection` object.
Example
This example collapses the selection to its end and then inserts a new paragraph following it.

```
collapse range text object of selection direction collapse end
type paragraph selection
```

Command: type text

Inserts the specified text. If the replace selection property of the Word options class is true, the selection is replaced by the specified text. If the replace selection property is false, the specified text is inserted before the selection.

Syntax

```
type text  selection object  Required. An expression that returns a selection object.
   text  Unicode text  Required. The text to be inserted.
```

Example

If Typing replaces selection is selected in the Edit pane of the Preferences dialog box, this example collapses the selection before inserting "Hello." This technique prevents existing document text from being replaced.

```
if replace selection of settings is true then
   collapse range text object of selection direction collapse start
   type text selection text "Hello"
end if
```

This example inserts "Title" followed by a new paragraph.

```
set replace selection of settings to false
set replace selection of settings to false
type text selection text "Title"
type paragraph selection
```

Command: undo

Undoes the last action or a sequence of actions, which are displayed in the Undo list. Returns true if the actions were successfully undone.

Syntax

```
undo document  Required. An expression that returns a document object.
   [times integer]  Optional. The number of actions to be undone.
```

Example

This example undoes the last two actions taken in Sales.doc.

```
undo document "Sales.doc" times 2
```

This example undoes the last action. If the action is successfully undone, a message is displayed in the status bar.

```
set u to undo active document
if u is true then set status bar to "Undo was successful"
```
**Command: undo clear**

Clears the list of actions that can be undone for the specified document. Corresponds to the list of items that appears when you click the arrow beside the **Undo** button on the **Standard** toolbar.

**Note** Include this method at the end of a macro to keep script actions from appearing in the **Undo** box (for example, "VBA-Range.InsertBefore").

**Syntax**

```
undo clear  document  Required. An expression that returns a document object.
```

**Example**

This example clears the list of actions that can be undone for the active document.

```
undo clear active document
```

---

**Command: unlink**

Replaces the specified field with its most recent result.

**Syntax**

```
unlink  field  Required. An expression that returns a field.
```

**Remarks**

When you unlink a field, its current result is converted to text or a graphic and can no longer be updated automatically. Note that some fields, such as XE (Index Entry) fields and SEQ (Sequence) fields, cannot be unlinked.

**Example**

This example unlinks the first field in "Sales.doc."

```
unlink field 1 of document "Sales.doc"
```

This example updates and unlinks all the fields in the first section in the active document.

```
repeat with aField in (get fields of section 1 of active document)
    update field aField
    unlink aField
end repeat
```

---

**Command: unprotect**

Removes protection from the specified document. If the document isn’t protected, this command generates an error.

**Syntax**

```
unprotect  document  Required. An expression that returns a document object.
```

```
[password  Unicode text]  Optional. The password string used to protect the document.
    Passwords are case-sensitive. If the document is protected with a password and the correct password isn’t supplied, a dialog box prompts the user for the password.
```
Example

This example unprotects the active document, using "Blue" as the password.
if protection type of active document is not no document protection then
    unprotect active document password "Blue"
end if

This example unprotects the active document. If the document has a password, a dialog box prompts the user for the password. Text is inserted, and the document is protected for revisions.

set aDoc to active document
if protection type of aDoc is not no document protection then
    unprotect aDoc
    insert text "department six" at text object of selection
    protect aDoc protection type allow only revisions password "Blue"
end if

Command: update

Updates the values shown in a built-in Word dialog box; the entries shown in the specified index, table of authorities, table of figures or table of contents; or the specified link.

Note  To update the page numbers of items in a table of contents or figures, use the update page numbers command.

Syntax

update  link format/table of figures/table of contents/table of authorities/dialog/index  Required.
An expression that returns an object in the list.

Example

This example updates the first table of figures in the active document.
if (count of tables of figures of active document) ≥ 1 then
    update table of figures 1 of active document
end if

This example returns a dialog object that refers to the Font dialog box. It then changes the font applied to the selection object to Arial, updates the dialog box values, and displays the Font dialog box.

set myDialog to (get dialog dialog format font)
set name of font object of selection to "Arial"
update myDialog
show myDialog
This example updates any fields in the active document that aren't updated automatically.

repeat with aField in (get fields of active document)
   if auto update of link format of aField is false then
      update link format of aField
   end if
end repeat

Command: update document

Updates the envelope in the document with the current envelope settings.

Note If you use this command before an envelope has been added to the document, an error occurs.

Syntax

update document envelope Required. An expression that returns an envelope object.

Example

This example formats the envelope in Report.doc to use a custom envelope size (4.5 inches by 7.5 inches).

try
   set myEnv to envelope object of document "Report.doc"
   set default height of myEnv to inches to points inches 4.5
   set default width of myEnv to inches to points inches 7.5
   update document myEnv
on error
   display dialog "Report.doc doesn't include an envelope"
end try

This example adds an envelope to the active document, using predefined addresses. The default envelope bar code and Facing Identification Mark (FIM-A) settings are set to true, and the envelope in the active document is updated.

set addr to "Luciano Ravenna" & return & "1234 E. Main St." & return ¬
   & "Our Town, WA 98004"
set ret to "Jacob Sanjay" & return & "N. 33rd St." & return ¬
   & "Other Town, WA 98040"
insert envelope data envelope object of active document address addr ¬
   return address ret
set default print bar code of envelope object of active document to true
set default print FIMA of envelope object of active document to true
update document envelope object of active document
**Command: update field**

Updates the result of the field object. Returns `true` if the field is updated successfully.

**Syntax**

`update field field`  
Required. An expression that returns a `field` object.

**Example**

This example updates the first field in the active document and displays a message in the status bar indicating whether or not the field was updated successfully.

```plaintext
set uField to update field field 1 of active document
if uField is true then
   set status bar to "Field updated"
else
   set status bar to "Error, field not updated"
end if
```

**Command: update page numbers**

Updates the page numbers for items in the specified table of contents or table of figures.

**Syntax**

`update page numbers table of contents/table of figures`  
Required. An expression that returns a `table of contents` or `table of figures` object.

**Example**

This example updates all tables of figures in Sales.doc.

```plaintext
repeat with figs in (get tables of figures of document "Sales.doc")
   update page numbers figs
end repeat
```

This example inserts a page break at the insertion point and then updates the page numbers for the first table of contents in the active document.

```plaintext
collapse range text object of selection direction collapse start
insert break at text object of selection break type page break
update page numbers table of contents 1 of active document
```

**Command: update source**

Replaces the specified field with its most recent result.

**Syntax**

`update source field`  
Required. An expression that returns a `field`. 
**Example**
This example updates the INCLUDETEXT fields in the active document.
```
repeat with aField in (get fields of active document)
    if field type of aField is field include text then
        update source aField
    end if
end repeat
```

**Command: update styles**
Copies all styles from the attached template into the document, overwriting any existing styles in the document that have the same name.

**Syntax**
```
update styles  document  Required. An expression that returns a document object.
```

**Example**
This example copies the styles from the attached template into each open document, and then it closes each document.
```
repeat with aDoc in (get documents)
    update styles aDoc
    close aDoc saving yes
end repeat
```

This example changes the formatting of the Heading 1 style in the template attached to the active document. The **update styles** command updates the styles in the active document, including the Heading 1 style.
```
set aDoc to open as document attached template of active document
set name of font object of Word style style heading1 of aDoc to "Arial"
set bold of font object of Word style style heading1 of aDoc to False
close aDoc saving yes
update styles active document
```

**Command: update summary properties**
Updates the keyword and comment text in the Properties dialog box (File menu) to reflect the AutoSummary content for the specified document.

**Syntax**
```
update summary properties  document  Required. An expression that returns a document object.
```
Example

This example highlights key points in the active document and updates the summary information in the Properties dialog box (File menu).

auto summarize active document auto summarize length 25 ~
    summary mode summary mode highlight
update summary properties active document

Command: use address book

Selects the address book that’s used as the data source for a data merge operation.

Syntax

use address book  data merge  Required. An expression that returns a data merge object.
    book type  Unicode text  Required. The address book to be used as the data merge data source.

Example

This example creates a new data merge main document that uses the Personal Address Book at its data source.

set myDoc to make new document
set main document type of data merge of myDoc to document type form letters
use address book data merge of myDoc book type "pab"

Command: use default folder suffix

Sets the folder suffix for the specified document to the default suffix for the language support you have selected or installed.

Syntax

use default folder suffix  web options  Required. An expression that returns a web options object.

Example

This example sets the folder suffix for the active document to the default suffix.

use default folder suffix web options of active document

Command: view property browser

Displays the property window for the selected control in the specified document.

Note  This command is available only from outside of Word.

Syntax

view property browser  document  Required. An expression that returns a document object.
Command: web page preview

Previews a document in a Web browser.

Syntax

web page preview  document  Required. An expression that returns a document object.

Example

This example displays the current document as it would appear if saved as a Web page.

web page preview active document

Command: Word help

Displays on-line Help information.

Syntax

Word help

help type  enumeration  Required. The on-line Help topic or window. Can be one of the following:

<table>
<thead>
<tr>
<th>Constant</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>help</td>
<td>Displays the Help window.</td>
</tr>
<tr>
<td>help about</td>
<td>Displays the About Microsoft Word dialog box (Word menu).</td>
</tr>
<tr>
<td>help contents</td>
<td>Displays the Help window.</td>
</tr>
<tr>
<td>help index</td>
<td>Displays the Help window.</td>
</tr>
<tr>
<td>help psshelp</td>
<td>Displays product support information.</td>
</tr>
<tr>
<td>help search</td>
<td>Displays the Help window.</td>
</tr>
</tbody>
</table>

Example

This example displays the Help window.

Word help help type help
Classes

callout............................................................................................................................................. 353
　callout format .................................................................................................................................. 355
　fill format ......................................................................................................................................... 356
　horizontal line format ......................................................................................................................... 357
　inline horizontal line ............................................................................................................................ 357
　inline picture ....................................................................................................................................... 358
　inline picture bullet .............................................................................................................................. 358
　inline shape ......................................................................................................................................... 359
　line format ......................................................................................................................................... 360
　line shape ......................................................................................................................................... 361
　picture ................................................................................................................................................ 362
　picture format ..................................................................................................................................... 363
　shadow format .................................................................................................................................... 364
　shape .................................................................................................................................................. 365
　standard inline horizontal line ........................................................................................................... 366
　text box .............................................................................................................................................. 367
　text frame .......................................................................................................................................... 368
　threeD format ..................................................................................................................................... 369
　word art ............................................................................................................................................ 370
　word art format .................................................................................................................................. 371
　wrap format ....................................................................................................................................... 372

Class: callout

Plural

callouts

Represents a borderless line callout in the drawing layer. The callout class inherits all the properties of the shape class, which includes all the shapes in the main story of a document or in all the headers and footers of a document.

Use callout index, where index is the name or the index number, to return a single callout object. To set the formatting options for a callout, use the callout format class.
This example adds a callout to a newly created document and then sets the callout angle to 45 degrees.

```javascript
set myDocument to make new document
set myCall to make new callout at myDocument with properties ¬
   {callout type: callout two, left position: 72, top: 36, width: 25, ¬
    height: 25}
set angle of callout format of myCall to angle45
```

**Properties**

<Inheritance> **shape**

Inherits the properties and elements of the **shape** class.

**callout format**

Returns a **callout format** object that contains callout formatting properties for the specified shape. Read-only.

**callout type**

Returns or sets the callout type. Read only.

Can be one of the following:

- **callout unset**
- **callout one** (a single-segment callout line that can be either horizontal or vertical)
- **callout two** (a single-segment callout line that rotates freely)
- **callout three** (a two-segment line)
- **callout four** (a three-segment line)

**Class: callout format**

**Plural**

**callout formats**

Represents the formatting options for a callout in the drawing layer.

To return or set formatting options for a callout, use **callout format** of **callout index**, where **index** is the name or the index number of a single **callout** object.

The example below specifies the following attributes of callout 3 in the active document:

- The callout will have a vertical accent bar that separates the text from the callout line.
- The angle between the callout line and the side of the callout text box will be 30 degrees.
- There will be no border around the callout text.
- The callout line will be attached to the top of the callout text box.
- The callout line will contain two segments.
For this example to work, shape three must be a callout.

```plaintext
set accent of callout format of callout 3 of active document to true
set angle of callout format of callout 3 of active document to angle30
set callout has border of callout format of callout 3 of active document to false
set callout type of callout format of callout 3 of active document to callout three
preset drop callout 3 of active document DropType drop top
```

**Properties**

**accent**

*True* if a vertical accent bar separates the callout text from the callout line. Read/write.

**angle**

Returns or sets the angle of the callout line. If the callout line contains more than one line segment, this property returns or sets the angle of the segment that is farthest from the callout text box. Read/write.

Can be one of the following:

- angle30
- angle45
- angle60
- angle90
- angle automatic
- angle unset

If you set the value of this property to anything other than *angle automatic*, the callout line maintains a fixed angle as you drag the callout.

**auto attach**

*True* if the place where the callout line attaches to the callout text box changes depending on whether the origin of the callout line (where the callout points to) is to the left or right of the callout text box. Read/write.

When the value of this property is *true*, the drop value (the vertical distance from the edge of the callout text box to the place where the callout line attaches) is measured from the top of the text box when the text box is to the right of the origin, and it's measured from the bottom of the text box when the text box is to the left of the origin. When the value of this property is *false*, the drop value is always measured from the top of the text box, regardless of the relative positions of the text box and the origin. To set the drop value, use the *custom drop* command, and use the *drop* property to return the drop value.

Setting this property affects a callout only if it has an explicitly set drop value, that is, if the value of the *drop type* property is *drop custom*. By default, callouts have explicitly set drop values when they're created.
**Drawing Suite**

**auto length**

*True* if the first segment of the callout line (the segment attached to the text callout box) is scaled automatically whenever the callout is moved. *False* if the first segment of the callout retains the fixed length specified by the *callout format length* property whenever the callout is moved. Applies only to callouts whose lines consist of more than one segment (types *callout three* and *callout four*). Read-only.

This property is read-only. To set this property to *true*, use the *automatic length* command. To set this property to *false*, use the *custom length* command.

**callout format length**

When the *auto length* property of the specified callout is set to *false*, the *callout format length* property returns the length (in points) of the first segment of the callout line (the segment attached to the text callout box). Applies only to callouts whose lines consist of more than one segment (types *callout three* and *callout four*). Read-only.

This property is read-only. To set the value of this property, use the *custom length* command.

**callout has border**

*True* if the text in the specified callout is surrounded by a border. Read/write.

**callout type**

Returns or sets the callout type. Read only. Can be one of the following:

- *callout unset*
- *callout one*
- *callout two*
- *callout three*
- *callout four*

**drop**

For callouts with an explicitly set drop value, this property returns the vertical distance (in points) from the edge of the text bounding box to the place where the callout line attaches to the text box. This distance is measured from the top of the text box unless the *auto attach* property is set to *true* and the text box is to the left of the origin of the callout line (the place that the callout points to), in which case the drop distance is measured from the bottom of the text box. Read-only.

To set the value of this property, use the *custom drop* command.

The value of this property accurately reflects the position of the callout line attachment to the text box only if the callout has an explicitly set drop value, that is, if the value of the *drop type* property is *drop custom*. To set the *drop type* property to *drop custom*, use the *preset drop* command.
Drawing Suite

*drop type*

Returns a value that indicates where the callout line attaches to the callout text box. Read-only.

Can be one of the following:

- drop bottom
- drop center
- drop custom
- drop unset
- drop top

If the callout drop type is **drop custom**, the values of the *drop* and *auto attach* properties and the relative positions of the callout text box and callout line origin (the place that the callout points to) are used to determine where the callout line attaches to the text box.

This property is read-only. To set the value of this property, use the `preset drop` command.

*gap*

Returns or sets the horizontal distance (in points) between the end of the callout line and the text bounding box. Read/write.

**Class: fill format**

**Plural**

fill formats

Represents fill formatting for a shape. A shape can have a solid, gradient, texture, pattern, picture, or semi-transparent fill.

The following example adds a rectangle to the active document and then sets the gradient and color for the rectangle's fill.

```qscript
set newRect to make new shape at active document with properties ¬
{ auto shape type: autoShape rectangle, left position: 90, top: 90, ¬
  width: 90, height: 80 }
set fore color of fill format of newRect to ({0, 128, 128} as RGB color)
one color gradient newRect gradient style horizontal gradient gradient ¬
  variant 1 gradient degree 1.0
```

**Remarks**

Many of the properties of the *fill format* object are read-only. To set one of these properties, you must apply the corresponding command.
Properties

back color

Returns or sets an RGB color that represents the background color for the specified fill format. Read/write.

fill type

Returns the shape fill format type. Read-only.
Can be one of the following:

- fill unset
- fill solid
- fill patterned
- fill gradient
- fill textured
- fill background
- fill picture

fore color

Returns or sets an RGB color that represents the foreground color for the fill format. Read-only.

gradient color type

Returns the gradient color type for the specified fill. Read-only.
Can be one of the following:

- gradient type unset
- single shade gradient type
- two colors gradient type
- preset colors gradient type

To set the gradient type for the fill format, use the one color gradient, preset gradient, or two color gradient command.
**Drawing Suite**

*gradient degree*

Returns a value that indicates how dark or light a one-color gradient fill format is. A value of 0 (zero) means that black is mixed in with the shape's foreground color to form the gradient; a value of 1 means that white is mixed in; and values between 0 and 1 mean that a darker or lighter shade of the foreground color is mixed in. Read-only.

To set the gradient degree for the fill format, use the `one color gradient` command.

*gradient style*

Returns the gradient style for the specified fill. Read-only.

Can be one of the following:

- gradient unset
- horizontal gradient
- vertical gradient
- diagonal up gradient
- diagonal down gradient
- from corner gradient
- from center gradient
- from title gradient (used only in PowerPoint)

To set the gradient style for the fill, use the `one color gradient` or `two color gradient` command.

**Note** Attempting to return this property for a fill that doesn't have a gradient generates an error. To determine whether the fill has a gradient, use the `fill type` property.

*gradient variant*

Returns the gradient variant for the specified fill format as an integer value from 1 to 4 for most gradient fills. If the gradient style is `from center gradient`, this property returns either 1 or 2. The values for this property correspond to the gradient variants (numbered from left to right and from top to bottom) on the Gradient tab in the Fill Effects dialog box. Read-only.

To set the gradient variant for the fill format, use the `one color gradient` or `two color gradient` command.
pattern

Returns a value that represents the pattern applied to the specified fill format. Read-only.

Can be one of the following:

- five percent pattern
- ten percent pattern
- twenty percent pattern
- twenty five percent pattern
- thirty percent pattern
- forty percent pattern
- fifty percent pattern
- sixty percent pattern
- seventy percent pattern
- seventy five percent pattern
- eighty percent pattern
- ninety percent pattern
- dark downward diagonal pattern
- dark horizontal pattern
- dark upward diagonal pattern
- dark vertical pattern
- dashed downward diagonal pattern
- dashed horizontal pattern
- dashed upward diagonal pattern
- dashed vertical pattern
- diagonal brick pattern
- divot pattern
- dotted diamond pattern
- dotted grid pattern
- horizontal brick pattern
- large checker board pattern
- large confetti pattern
- large grid pattern
- light downward diagonal pattern
- light horizontal pattern
- light upward diagonal pattern
- light vertical pattern
- unset pattern
- narrow horizontal pattern
- outlined diamond pattern
- plaid pattern
- plaid pattern
- plaid pattern
- plaid pattern
- plaid pattern
- plaid pattern
- plaid pattern
- solid diamond pattern
- sphere pattern
- trellis pattern
- wave pattern
- weave pattern
- wide downward diagonal pattern
- wide upward diagonal pattern
- zig zag pattern

To set the pattern for the fill format, use the **patterned** command.

To set the colors used in the pattern, use the **back color** and **fore color** properties.

preset gradient type

Returns the preset gradient type for the specified fill format. Read-only.

Can be one of the following:

- gradient brass
- gradient calm water
- gradient chrome
- gradient chrome2
- gradient daybreak
- gradient desert
- gradient early sunset
- gradient fire
- gradient fog
- gradient gold
- gradient gold2I
- gradient horizon
- gradient late sunset
- gradient mahogany
- gradient moss
- gradient nightfall
- gradient ocean
- gradient parchment
- gradient peacock
- gradient rainbow
- gradient rainbow2
- gradient sapphire
- gradient silver
- gradient wheat
- preset gradient unset

To set the preset gradient type for the fill format, use the **preset gradient** command.
Drawing Suite

**preset texture**

Returns the preset texture for the specified fill format. Read-only.

Can be one of the following MsoPresetTexture constants:

- `preset texture unset`
- `texture blue tissue paper`
- `texture bouquet`
- `texture brown marble`
- `texture canvas`
- `texture cork`
- `texture denim`
- `texture fish fossil`
- `texture granite`
- `texture green marble`
- `texture medium wood`
- `texture newsprint`
- `texture oak`
- `texture paper bag`
- `texture papyrus`
- `texture parchment`
- `texture pink tissue paper`
- `texture purple mesh`
- `texture recycled paper`
- `texture sand`
- `texture stationery`
- `texture walnut`
- `texture water droplets`
- `texture white marble`
- `texture woven mat`

To set the preset texture for the fill format, use the `preset textured` command.

**texture name**

Returns the name of the custom texture file for the specified fill format. Read-only.

To set the texture file for the fill format, use the `user textured` command.

**texture type**

Returns the texture type for the specified fill format. Read-only.

Can be one of the following:

- `texture type texture type unset`
- `texture type preset texture`
- `texture type user defined texture`

To set the texture type for the fill format, use the `preset textured`, `user picture`, or `user textured` command.

**transparency**

Returns or sets the degree of transparency of the specified fill format as a value between 0.0 (opaque) and 1.0 (clear). Read/write.

The value of this property affects the appearance of solid-colored fills and lines only; it has no effect on the appearance of patterned lines or patterned, gradient, picture, or textured fills.

**visible**

True if the specified object, or the formatting applied to it, is visible. Read/write.
Class: horizontal line format

Plural

horizontal line formats

Represents horizontal line formatting.

To return a horizontal line format object, use the horizontal line format property of either the inline horizontal line or standard inline horizontal line object. This example sets the alignment for a new horizontal line.

make new standard inline horizontal line at the end of the active document

set alignment of horizontal line format of ~

standard inline horizontal line 1 of active document ~
to horizontal line align left

This example adds a horizontal line without any 3-D shading.

make new standard inline horizontal line at the end of the active document

set no shade of horizontal line format of standard inline horizontal line 1 ~
of active document to true

This example adds a horizontal line and sets its length to 50% of the window width.

make new standard inline horizontal line at the end of the active document

set percent width of horizontal line format of ~

standard inline horizontal line 1 of active document to 50

Properties

alignment

Returns or sets the alignment for the specified horizontal line. Read/write.

Can be one of the following:

- horizontal line align left
- horizontal line align center
- horizontal line align right

no shade

True if Word draws the specified horizontal line without 3-D shading. Read/write.

You can use this property only with horizontal lines that are not based on an existing image file.

percent width

Returns or sets the length of the specified horizontal line expressed as a percentage of the window width. Read/write.

Setting this property also sets the width type property to horizontal line percent width.
width type

Returns or sets the width type for the specified horizontal line format object. Read/write.

This property can be either of the following:

- **horizontal line fixed width**: Word interprets the width (length) of the specified horizontal line as a fixed value (in points). This is the default value for new inline horizontal lines. Setting the width property for the inline shape object associated with a horizontal line sets the width type property to this value.

- **horizontal line percent width**: Word interprets the width (length) of the specified horizontal line as a percentage of the screen width. This is the default value for new standard inline horizontal lines. Setting the percent width property on a horizontal line sets the width type property to this value.

**Class: inline horizontal line**

**Plural**

inline horizontal lines

Represents a horizontal line that is based on an image file and that is located in the text layer of a document. The inline horizontal line class inherits all the properties of the inline shape class, which includes all the inline shapes in the main story of a document or in all the headers and footers of a document.

Use inline horizontal line index, where index is the name or the index number, to return a single inline horizontal line object. The following example sets the width of the first inline horizontal line in the active document to half the size of the window.

```plaintext
set percent width of horizontal line format of inline horizontal line 1 of active document to 50
```

**Properties**

<Inheritance> inline shape

Inherits the properties and elements of the inline shape class.

**file name**

Returns the file name of the image used for the inline horizontal line. Read-only.

**Class: inline picture**

**Plural**

inline pictures

Represents a picture object in the text layer of a document. The inline picture class inherits all the properties of the inline shape class, which includes all the inline shapes in the main story of a document or in all the headers and footers of a document.
Use **inline picture index**, where *index* is the index number of the picture, to return a single **inline picture** object.

The following example adds a new picture to `myDoc`. The inserted picture is linked to the file from which it was created and is saved with the document.

```plaintext
set myDoc to active document
make new inline picture at myDoc with properties -
    {file name:"Users:Shared:clipart:音乐.gif", link to file:true -
        save with document:true}
```

This example sets the lock aspect ratio property of the picture to **true**, allowing the picture to retain its original proportions when it is resized.

```plaintext
set lock aspect ratio of inline picture 1 of active document to true
```

**Properties**

<**Inheritance**> **inline shape**

Inherits the properties and elements of the **inline shape** class.

- **file name**
  Returns the file name of the picture used for the inline picture. Read-only.

- **link to file**
  **True** if the inline picture is linked to the file. Read-only.

- **picture format**
  Returns a **picture format** object that contains picture formatting properties for the specified **inline picture** object. Read-only.

- **save with document**
  **True** if the picture is saved with the document. Read-only.

**Class: inline picture bullet**

**Plural**

**inline picture bullets**

Represents a picture bullet based on an image file in the text layer of a document. The **inline picture bullet** class inherits all the properties of the **inline shape** class, which includes all the inline shapes in the main story of a document or in all the headers and footers of a document.

Use **inline picture bullet index**, where *index* is the index number of the picture, to return a single **inline picture bullet** object.

This example adds a picture bullet to each paragraph in the selected text using a file named "RedBullet.gif."

```plaintext
repeat with aPara in (get paragraphs of selection)
    make new inline picture bullet at the beginning of the text object of -
        aPara with properties {file name:"MacintoshHD:Pics:RedBullet.gif"}
end repeat
```
Properties

<Inheritance> inline shape

Inherits the properties and elements of the inline shape class.

file name

Returns the file name of the picture used for the inline picture. Read-only.

Class: inline shape

Plural

inline shapes

Represents an object in the text layer of a document. An inline shape can only be a picture or an OLE object. Inline shape objects are treated like characters and are positioned as characters within a line of text. The inline shapes list contains all the shapes in a document, text range, or selection.

Use inline shape index, where index is the index number, to return a single inline shape object.

Inline shapes don't have names. The following example activates the first inline shape in the active document.

activate object inline shape 1 of active document

Note

There are two classes that represent shapes: the shape class, which represents shapes in the drawing layer of a document, and the inline shape class, which represents shapes in the text layer of a document. The inline shape class includes the following subclasses: inline horizontal line, inline picture, inline picture bullet, and standard inline horizontal line.

Remarks

Inline shape objects are anchored to a text range but are free-floating and can be positioned anywhere on the page. To convert shapes from one type to the other, you can use the convert to inline shape command and the convert to shape command. You can convert only pictures and OLE objects to inline shapes. To return the type of inline shape, use the inline shape type property.

When you open a document created in an earlier version of Word, pictures are converted to inline shapes.

Properties

alternative text

Returns or sets the alternative text associated with a shape on a Web page. Read/write.

border options

Returns a border options object that represents all the borders for the specified inline shape. Read-only.

field

Returns a field object that represents the field associated with the specified shape. Read-only.

fill format

Returns a fill format object that contains fill formatting properties for the specified shape. Read-only.
**height**

Returns or sets the height of the specified inline shape. Read/write.

**horizontal line format**

Returns a horizontal line format object that contains the horizontal line formatting for the specified inline shape object. Read-only.

**hyperlink**

Returns a hyperlink object that represents the hyperlink associated with the specified inline shape object. Read-only.

**Note** If no hyperlink is associated with the specified shape, an error occurs.

**inline shape scale height**

Scales the height of the specified inline shape relative to its original size. Read/write.

**inline shape type**

Returns the type of inline shape. Read-only.

Can be one of the following:

- inline shape embedded oleobject
- inline shape linked oleobject
- inline shape picture
- inline shape linked picture
- inline shape olecontrol object
- inline shape horizontal line
- inline shape picture horizontal line
- inline shape linked picture horizontal line
- inline shape picture bullet

**line format**

Returns a line format object that contains line formatting properties for the specified shape. (For a line, the line format object represents the line itself; for a shape with a border, the line format object represents the border.) Read-only.

**link format**

Returns a link format object that represents the link options of the specified inline shape that's linked to a file. Read-only.

**lock aspect ratio**

**True** if the specified shape retains its original proportions when you resize it. **False** if you can change the height and width of the shape independently of one another when you resize it. Read/write.
picture format

Returns a picture format object that contains picture formatting properties for the specified inline shape object. Read-only.

text object

Returns a text range object that represents the portion of a document that's contained in the specified inline shape object. Read-only.

width

Returns or sets the width (in points) of the specified inline shape object. Read/write.

Class: line format

Plural

line formats

Represents line and arrowhead formatting. For a line, the line format object contains formatting information for the line itself; for a shape with a border, this object contains formatting information for the shape's border.

To return a line format object, use the line format property of a shape or inline shape. The following example adds a blue, dashed line to the active document. There's a short, narrow oval at the line's starting point and a long, wide triangle at its end point.

make new line shape at active document with properties {begin line X:100 ¬
begin line Y:100, end line X:200, end line Y:300} ¬
set dash style of line format of line shape 1 of active document to ¬
dash style dash dot dot
set fore color of line format of line shape 1 of active document to ¬
{(50, 0, 128) as RGB color}
set begin arrowhead length of line format of line shape 1 of active document ¬
to short arrowhead
set begin arrowhead style of line format of line shape 1 of active document ¬
to oval arrowhead
set begin arrowhead width of line format of line shape 1 of active document ¬
to narrow width arrowhead
set end arrowhead length of line format of line shape 1 of active document ¬
to long arrowhead
set end arrowhead style of line format of line shape 1 of active document ¬
to triangle arrowhead
set end arrowhead width of line format of line shape 1 of active document ¬
to wide arrowhead
Properties

back color
Returns or sets an RGB color that represents the background color for the specified line format. Read/write.

begin arrowhead length
Returns or sets the length of the arrowhead at the beginning of the specified line. Read/write. Can be one of the following:

- arrowhead length unset
- short arrowhead
- medium arrowhead
- long arrowhead

begin arrowhead style
Returns or sets the style of the arrowhead at the beginning of the specified line. Read/write. Can be one of the following:

- arrowhead style unset
- no arrowhead
- triangle arrowhead
- open_arrowhead
- stealth arrowhead
- diamond arrowhead
- oval arrowhead

begin arrowhead width
Returns or sets the width of the arrowhead at the beginning of the specified line. Read/write. Can be one of the following:

- arrowhead width unset
- narrow width arrowhead
- medium width arrowhead
- wide arrowhead
**dash style**

Returns or sets the dash style for the specified line. Read/write.

Can be one of the following:

- line dash style unset
- line dash style solid
- line dash style square dot
- line dash style round dot
- line dash style dash
- line dash style dash dot
- line dash style long dash
- line dash style dash dot
- line dash style long dash dot

**end arrowhead length**

Returns or sets the length of the arrowhead at the end of the specified line. Read/write.

Can be one of the following:

- arrowhead length unset
- short arrowhead
- medium arrowhead
- long arrowhead

**end arrowhead style**

Returns or sets the style of the arrowhead at the end of the specified line. Read/write.

Can be one of the following:

- arrowhead style unset
- no arrowhead
- triangle arrowhead
- open_arrowhead
- stealth arrowhead
- diamond arrowhead
- oval arrowhead
end arrowhead width

Returns or sets the width of the arrowhead at the end of the specified line. Read/write.
Can be one of the following:

- arrowhead width unset
- narrow width arrowhead
- medium width arrowhead
- wide arrowhead

fore color

Returns or sets an RGB color that represents the foreground color for the line format. Read-only.

line style

Returns or sets the line style for the fill format object. Read/write.
Can be one of the following:

- line style unset
- single line
- thin thin line
- thin thick line
- thick thin line
- thick between thin line
### Pattern

Returns or sets a value that represents the pattern applied to the specified line format. Read/write.

Can be one of the following:

<table>
<thead>
<tr>
<th>Pattern</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>five percent pattern</td>
<td>Large checker board pattern</td>
</tr>
<tr>
<td>ten percent pattern</td>
<td>Large confetti pattern</td>
</tr>
<tr>
<td>twenty percent pattern</td>
<td>Large grid pattern</td>
</tr>
<tr>
<td>twenty five percent pattern</td>
<td>Light downward diagonal pattern</td>
</tr>
<tr>
<td>thirty percent pattern</td>
<td>Light horizontal pattern</td>
</tr>
<tr>
<td>forty percent pattern</td>
<td>Light upward diagonal pattern</td>
</tr>
<tr>
<td>fifty percent pattern</td>
<td>Light vertical pattern</td>
</tr>
<tr>
<td>sixty percent pattern</td>
<td>Unset pattern</td>
</tr>
<tr>
<td>seventy percent pattern</td>
<td>Narrow horizontal pattern</td>
</tr>
<tr>
<td>seventy five percent pattern</td>
<td>Narrow vertical pattern</td>
</tr>
<tr>
<td>eighty percent pattern</td>
<td>Outlined diamond pattern</td>
</tr>
<tr>
<td>ninety percent pattern</td>
<td>Plaid pattern</td>
</tr>
<tr>
<td>dark downward diagonal pattern</td>
<td>Shingle pattern</td>
</tr>
<tr>
<td>dark horizontal pattern</td>
<td>Small checker board pattern</td>
</tr>
<tr>
<td>dark upward diagonal pattern</td>
<td>Small confetti pattern</td>
</tr>
<tr>
<td>dark vertical pattern</td>
<td>Small grid pattern</td>
</tr>
<tr>
<td>dashed downward diagonal pattern</td>
<td>Solid diamond pattern</td>
</tr>
<tr>
<td>dashed horizontal pattern</td>
<td>Sphere pattern</td>
</tr>
<tr>
<td>dashed upward diagonal pattern</td>
<td>Trellis pattern</td>
</tr>
<tr>
<td>dashed vertical pattern</td>
<td>Wave pattern</td>
</tr>
<tr>
<td>diagonal brick pattern</td>
<td>Weave pattern</td>
</tr>
<tr>
<td>divot pattern</td>
<td>Wide downward diagonal pattern</td>
</tr>
<tr>
<td>dotted diamond pattern</td>
<td>Wide upward diagonal pattern</td>
</tr>
<tr>
<td>dotted grid pattern</td>
<td>Zig zag pattern</td>
</tr>
<tr>
<td>horizontal brick pattern</td>
<td></td>
</tr>
</tbody>
</table>

### Transparency

Returns or sets the degree of transparency of the specified line as a value between 0.0 (opaque) and 1.0 (clear). Read/write.

The value of this property affects the appearance of solid-colored lines only; it has no effect on the appearance of patterned lines.

### Visible

**True** if the specified object, or the formatting applied to it, is visible. Read/write.

### Weight

Returns or sets the thickness (in points) of the specified line. Read/write.
Class: line shape

Plural
line shapes

Represents a line drawn in the drawing layer of the document. The line shape class inherits all the properties of the shape class, which includes all of the shapes in the main story of a document or in all the headers and footers of a document.

Use line shape index, where index is the name or the index number, to return a single line shape object.

This example adds a dashed, green line to a new document.

```plaintext
set myDocument to make new document
set myLine to make new line shape at myDocument with properties ¬

  {begin line X:100, begin line Y:100, end line X:60, end line Y:20}

set dash style of line format of myLine to line dash style dash
set fore color of line format of myLine to ({0, 128, 0} as RGB color)
```

This example formats the first line in the document as a red arrow.

```plaintext
set begin arrowhead style of line format of line shape 1 of active document ¬

to no arrowhead
set end arrowhead style of line format of line shape 1 of active document ¬

to triangle arrowhead
set fore color of line format of line shape 1 of active document ¬

to ({128, 0, 0} as RGB color)
```

Properties

<Inheritance> shape

Inherits the properties and elements of the shape class.

`begin line X`

Returns or sets the starting X coordinate for the line shape. Read/write.

`begin line Y`

Returns or sets the starting Y coordinate for the line shape. Read/write.

`end line X`

Returns or sets the ending X coordinate for the line shape. Read/write.

`end line Y`

Returns or sets the ending Y coordinate for the line shape. Read/write.
Class: picture

Plural
pictures

Represents a picture object in the drawing layer of a document. The picture class inherits all the properties of the shape class, which includes all the shapes in the main story of a document or in all the headers and footers of a document.

Use picture index, where index is the name or the index number, to return a single picture object.

The following example adds a new picture to the drawing layer of myDoc. The inserted picture is linked to the file from which it was created and is saved with the document.

set myDoc to active document
make new picture at myDoc with properties ¬
    {file name:"Users:Shared:clipart:music.gif", link to file:true ¬
    save with document:true}

Properties

<Inheritance> shape
    Inherits the properties and elements of the shape class.

file name
    Returns the file name of the picture used for the inline horizontal line. Read-only.

link to file
    True if the inline picture is linked to the file. Read-only.

picture format
    Returns a picture format object that contains picture formatting properties for the specified picture object. Read-only.

save with document
    True if the picture is saved with the document. Read-only.

Class: picture format

Plural
picture formats

Contains properties that apply to pictures.

Use the picture format property to return a picture format object. The following example sets the brightness and contrast for picture 1 of the active document and crops 18 points off the bottom of the picture. This example will only work on a picture object.

set brightness of picture format of picture 1 of active document to 0.3
set contrast of picture format of picture 1 of active document to 0.7
set crop bottom of picture format of picture 1 of active document to 18
Note Cropping is calculated relative to the original size of the picture. For example, if you insert a picture that is originally 100 points high, rescale it so that it’s 200 points high, and then set the crop bottom property to 50, 100 points (not 50) will be cropped off the bottom of your picture. However, while cropping measurements are calculated relative to the object’s original scale, those measurements reflect absolute distances from the edge of the object. You cannot set any of the cropping properties to a negative number.

Properties

brightness
Returns or sets the brightness of the specified picture. The value for this property must be a number from 0.0 (dimmest) to 1.0 (brightest). Read/write.

contrast
Returns or sets the contrast for the specified picture. The value for this property must be a number from 0.0 (the least contrast) to 1.0 (the greatest contrast). Read/write.

crop bottom
Returns or sets the number of points that are cropped off the bottom of the specified picture. Read/write.

crop left
Returns or sets the number of points that are cropped off the left side of the specified picture. Read/write.

crop right
Returns or sets the number of points that are cropped off the right side of the specified picture. Read/write.

crop top
Returns or sets the number of points that are cropped off the top of the specified picture. Read/write.

transparency color
Returns or sets the transparent color for the specified picture as an RGB value. For this property to take effect, the transparent background property must be set to true. Read/write.

transparent background
True if the parts of the picture that are the color defined as the transparent color appear transparent. To set the transparent color, use the transparency color property. Read/write.
Class: shadow format

Plural

shadow formats

Represents shadow formatting for a shape.

Use the shadow property to return a shadow format object. The following example adds a shadowed rectangle to the active document. The semitransparent, blue shadow is offset 5 points to the right of the rectangle and 3 points above it.

```
set newShape to make new shape at active document with properties
  { auto shape type: autoshape rectangle, left position: 50, top: 50,
    width: 100, height: 200 }
set fore color of shadow of newShape to ({0, 0, 128} as RGB color)
set offset x of shadow of newShape to 5
set offset y of shadow to newShape to -3
set transparency of shadow of newShape to 0.5
set visible of shadow of newShape to true
```

Properties

fore color

Returns or sets an RGB color that represents the foreground color for the line format. Read-only.

obscured

True if the shadow of the specified shape appears filled in and is obscured by the shape, even if the shape has no fill. False if the shadow has no fill and the outline of the shadow is visible through the shape if the shape has no fill. Read/write.

offset x

Returns or sets the horizontal offset (in points) of the shadow from the specified shape. A positive value offsets the shadow to the right of the shape; a negative value offsets it to the left. Read/write.

offset y

Returns or sets the vertical offset (in points) of the shadow from the specified shape. A positive value offsets the shadow to the right of the shape; a negative value offsets it to the left. Read/write.
shadow type

Returns or sets the shape shadow type. Read/write.

Can be one of the following:

- shadow unset
- shadow1
- shadow2
- shadow3
- shadow4
- shadow5
- shadow6
- shadow7
- shadow8
- shadow9
- shadow10
- shadow11
- shadow12
- shadow13
- shadow14
- shadow15
- shadow16
- shadow17
- shadow18
- shadow19
- shadow20

transparency

Returns or sets the degree of transparency of the specified shadow as a value between 0.0 (opaque) and 1.0 (clear). Read/write.

visible

True if the shadow format object, or the formatting applied to it, is visible. Read/write.

Class: shape

Plural
shapes

Elements
shape

Represents an object in the drawing layer, such as an AutoShape, freeform, OLE object, or picture. The shapes list includes all the shapes in the main story of a document or in all the headers and footers of a document.

A shape is always attached to an anchoring text range. You can position the shape anywhere on the page that contains the anchor.

Note  There are two classes that represent shapes: the shape class, which represents shapes in the drawing layer of a document, and the inline shape class, which represents shapes in the text layer of a document. The shape class includes the following subclasses: callout, line shape, picture, text box, and word art.

Returning an existing shape on a document

Use shape index, where index is the name or the index number, to return a single shape object. The following example horizontally flips shape 1 on the active document.

flip shape 1 of active document flip command flip horizontal
The following example horizontally flips the shape named "Rectangle 1" on the active document.

`flip shape "Rectangle 1" of active document flip command flip horizontal`

Each shape is assigned a default name when it is created. For example, if you add three different shapes to a document, they might be named "Rectangle 2," "TextBox 3," and "Oval 4." To give a shape a more meaningful name, set the name property.

**Returning a newly created shape**

To add a shape object to the list of shapes for the specified document and return a shape object that represents the newly created shape, use the `make` command and specify the type of shape you want to create with the auto shape type property. The following example adds a rectangle to the active document.

`make new shape at active document with properties {auto shape type:autoshape ¬ rectangle, left position:50, top:50, width:100, height:200}`

**Anchoring and positioning a shape**

Every shape object is anchored to a range of text. A shape is anchored to the beginning of the first paragraph that contains the anchoring range. The shape will always remain on the same page as its anchor.

You can view the anchor itself by selecting the shape. The shape's top and left position properties determine its vertical and horizontal positions. The shape's relative horizontal position and relative vertical position properties determine whether the position is measured from the anchoring paragraph, the column that contains the anchoring paragraph, the margin, or the edge of the page.

If the lock anchor property for the shape is set to true, you cannot drag the anchor from its position on the page.

**Formatting a shape**

Use the fill format property to return the fill format object, which contains all the properties for formatting the fill of a closed shape. The shadow property returns the shadow format object, which you use to format a shadow. Use the line format property to return the line format object, which contains properties for formatting lines and arrows. The wrap format property returns the wrap format object, which you use to define how text wraps around shapes. The threeD format property returns the threeD format object, which you use to create 3-D shapes. You can use the pick up and apply commands to transfer formatting from one shape to another.

Use the set shapes default properties command for a shape object to set the formatting for the default shape for the document. New shapes inherit many of their attributes from the default shape.

**Other important shape properties**

Use the shape type property to specify the type of shape: freeform, AutoShape, OLE object, callout, or linked picture, for instance. Use the auto shape type property to specify the type of AutoShape: oval, rectangle, or balloon, for instance.

Use the width and height properties to specify the size of the shape.

The text frame property returns the text frame object, which contains all the properties for attaching text to shapes and linking the text between text frames.
Remarks

Shape objects are anchored to a range of text but are free-floating and can be positioned anywhere on the page. Inline shape objects are treated like characters and are positioned as characters within a line of text. You can use the convert to inline shape command and the convert to shape command to convert shapes from one type to the other. You can convert only pictures to inline shapes.

Properties

anchor

Returns a text range object that represents the anchoring range for the specified shape or shape range. Read-only.

All shape objects are anchored to a range of text but can be positioned anywhere on the page that contains the anchor. If you specify the anchoring range when you create a shape, the anchor is positioned at the beginning of the first paragraph that contains the anchoring range. If you don’t specify the anchoring range, the anchoring range is selected automatically and the shape is positioned relative to the top and left edges of the page.

The shape will always remain on the same page as its anchor. If the lock anchor property for the shape is set to true, you cannot drag the anchor from its position on the page.

auto shape type

Returns or sets the type of AutoShape. Read/write.

Can be one of the following:

- autoshape rectangle
- autoshape trapezoid
- autoshape rounded rectangle
- autoshape isosceles triangle
- autoshape oval
- autoshape cross
- autoshape can
- autoshape bevel
- autoshape smiley face
- autoshape no symbol
- autoshape heart
- autoshape sun
- autoshape arc
- autoshape double brace
- autoshape left bracket
- autoshape left brace
- autoshape right arrow
- autoshape up arrow
- autoshape left right arrow
- autoshape quad arrow
- autoshape bent arrow
- autoshape left up arrow
- autoshape curved right arrow
- autoshape curved up arrow
- autoshape striped right arrow
- autoshape pentagon
- autoshape right arrow callout
- autoshape up arrow callout
<table>
<thead>
<tr>
<th>Autoshape Type</th>
<th>Autoshape Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>left right arrow callout</td>
<td>up down arrow callout</td>
</tr>
<tr>
<td>quad arrow callout</td>
<td>circular arrow</td>
</tr>
<tr>
<td>flowchart process</td>
<td>flowchart alternate process</td>
</tr>
<tr>
<td>flowchart decision</td>
<td>flowchart data</td>
</tr>
<tr>
<td>predefined process</td>
<td>flowchart internal storage</td>
</tr>
<tr>
<td>flowchart document</td>
<td>flowchart multi document</td>
</tr>
<tr>
<td>terminator</td>
<td>flowchart preparation</td>
</tr>
<tr>
<td>manual input</td>
<td>flowchart manual operation</td>
</tr>
<tr>
<td>flowchart card</td>
<td>flowchart offpage connector</td>
</tr>
<tr>
<td>flowchart summing junction</td>
<td>flowchart punched tape</td>
</tr>
<tr>
<td>flowchart stored data</td>
<td>flowchart or</td>
</tr>
<tr>
<td>flowchart sequential access storage</td>
<td>flowchart sort</td>
</tr>
<tr>
<td>flowchart direct access storage</td>
<td>flowchart merge</td>
</tr>
<tr>
<td>explosion one</td>
<td>flowchart delay</td>
</tr>
<tr>
<td>four point star</td>
<td>flowchart magnetic disk</td>
</tr>
<tr>
<td>eight point star</td>
<td>flowchart display</td>
</tr>
<tr>
<td>twenty four point star</td>
<td>flowchart explosion two</td>
</tr>
<tr>
<td>up ribbon</td>
<td>flowchart five point star</td>
</tr>
<tr>
<td>curved up ribbon</td>
<td>flowchart sixteen point star</td>
</tr>
<tr>
<td>vertical scroll</td>
<td>flowchart thirty two point star</td>
</tr>
<tr>
<td>wave</td>
<td>flowchart down ribbon</td>
</tr>
<tr>
<td>rectangular callout</td>
<td>flowchart curved down ribbon</td>
</tr>
<tr>
<td>oval callout</td>
<td>flowchart magnetic disk</td>
</tr>
<tr>
<td>line callout one</td>
<td>flowchart horizontal scroll</td>
</tr>
<tr>
<td>line callout three</td>
<td>flowchart double wave</td>
</tr>
<tr>
<td>line callout one accent bar</td>
<td>flowchart rounded rectangular callout</td>
</tr>
<tr>
<td>line callout three accent bar</td>
<td>flowchart cloud callout</td>
</tr>
<tr>
<td>line callout one no border</td>
<td>flowchart line callout two</td>
</tr>
<tr>
<td>line callout three no border</td>
<td>flowchart line callout four</td>
</tr>
<tr>
<td>line callout three accent bar</td>
<td>flowchart line callout four accent bar</td>
</tr>
<tr>
<td>line callout three no border</td>
<td>flowchart line callout two no border</td>
</tr>
<tr>
<td>line callout three border and accent bar</td>
<td>flowchart line callout four no border</td>
</tr>
<tr>
<td>action button custom</td>
<td>flowchart line callout two border and accent bar</td>
</tr>
<tr>
<td>action button help</td>
<td>flowchart line callout four border and accent bar</td>
</tr>
<tr>
<td>action button back or previous</td>
<td>flowchart line callout four border and accent bar</td>
</tr>
<tr>
<td>action button beginning</td>
<td>action button home</td>
</tr>
<tr>
<td>action button return</td>
<td>action button information</td>
</tr>
<tr>
<td>action button sound</td>
<td>action button forward or next</td>
</tr>
<tr>
<td>balloon</td>
<td>action button end</td>
</tr>
<tr>
<td></td>
<td>action button document</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**fill format**

Returns a [fill format](#) object that contains fill formatting properties for the specified shape. Read-only.

**height**

Returns or sets the height of the specified shape. Read/write.

**horizontal flip**

True if the specified shape is flipped around the horizontal axis. Read-only.
hyperlink

Returns a hyperlink object that represents the hyperlink associated with the specified shape object. Read-only.

Note If there's no hyperlink associated with the specified shape, an error occurs.

left position

Returns or sets the horizontal position (in points) of the specified shape or shape range. Read/write.

line format

Returns a line format object that contains line formatting properties for the specified shape. (For a line, the line format object represents the line itself; for a shape with a border, the line format object represents the border.) Read-only.

link format

Returns a link format object that represents the link options of the specified field, inline shape, or shape that's linked to a file. Read-only.

lock anchor

True if the specified shape object's anchor is locked to the anchoring range. When a shape has a locked anchor, you cannot move the shape's anchor by dragging it (the anchor doesn't move as the shape is moved). Read/write.

lock aspect ratio

True if the specified shape retains its original proportions when you resize it. False if you can change the height and width of the shape independently of one another when you resize it. Read/write.

name

Returns or sets the shape name. Read/write.

relative horizontal position

Specifies what the horizontal position of a frame or shape is relative to. Read/write.

Can be one of the following:

- relative horizontal position margin
- relative horizontal position page
- relative horizontal position column

relative vertical position

Specifies what the vertical position of a frame or shape is relative to. Read/write.

Can be one of the following:

- relative vertical position margin
- relative vertical position page
- relative vertical position paragraph
Drawing Suite

**rotation**

Returns or sets the number of degrees the specified shape is rotated around the z-axis. A positive value indicates clockwise rotation; a negative value indicates counterclockwise rotation. Read/write.

To set the rotation of a three-dimensional shape around the x-axis or the y-axis, use the rotation x property or the rotation y property of the **threeD format** object.

**shadow**

Returns a **shadow format** object that represents the shadow formatting for the specified shape. Read-only.

**shape type**

Returns the shape type. Read-only.

Can be one of the following:

- shape type auto
- shape type callout
- shape type chart
- shape type comment
- shape type embedded OLE object
- shape type form control
- shape type free form
- shape type group
- shape type line
- shape type linked OLE object
- shape type linked picture
- shape type media
- shape type embedded OLE control
- shape type picture
- shape type placeholder
- shape type text box
- shape type placeholder
- shape type text effect

**text frame**

Returns a **text frame** object that contains the text for the specified shape. Read-only.

**threeD format**

Returns a **threeD format** object that contains 3-D-effect formatting properties for the specified shape. Read-only.

**top**

Returns or sets the vertical position (in points) of the specified shape or shape range. Read/write.

**vertical flip**

True if the specified shape is flipped around the vertical axis. Read-only.

**visible**

True if the specified object, or the formatting applied to it, is visible. Read/write.

**width**

Returns or sets the width (in points) of the specified object. Read/write.

**wrap format**

Returns a **wrap format** object that contains the properties for wrapping text around the specified shape or shape range. Read-only.
**z order position**

Returns the position of the specified shape in the z-order. Shape 1 returns the shape at the back of the z-order, and shape (count of shapes) returns the shape at the front of the z-order. Read-only.

To set the shape's position in the z-order, use the **z order** command.

A shape's position in the z-order corresponds to the shape's index number in the **shapes** list. For example, if there are four shapes on myDocument, the expression `shape 1 of myDocument` returns the shape at the back of the z-order, and the expression `shape 4 of myDocument` returns the shape at the front of the z-order.

Whenever you add a new shape to a document, it's added to the front of the z-order by default.

**Class: standard inline horizontal line**

**Plural**

standard inline horizontal lines

Represents a standard horizontal line in the text layer of a document. The **standard inline horizontal line** class inherits all the properties of the **inline shape** class, which includes all the inline shapes in the main story of a document or in all the headers and footers of a document.

Use **standard inline horizontal line index**, where index is the name or the index number, to return a single **standard inline horizontal line** object.

This example adds a horizontal line above the fifth paragraph in the active document.

```
make new standard inline horizontal line at the beginning of text object of paragraph 5 of active document
```

**Properties**

<Inheritance> **inline shape**

Inherits the properties and elements of the **inline shape** class.

**Class: text box**

**Plural**

text boxes

Represents a text box object in the drawing layer of a document. The **text box** class inherits all the properties of the **shape** class, which includes all the shapes in the main story of a document or in all the headers and footers of a document.

Use **text box index**, where index is the name or the index number, to return a single **text box** object.

This example adds a text box that contains the text "Test Box" to a new document.

```
set myDocument to make new document
set myTBox to make new text box at myDocument with properties ¬
    {text orientation:horizontal, left position:100, top:100, width:300, ¬
    height:200}
set content of text range of text frame of myTBox to "Test Box"
```
Properties

<Inheritance> shape

Inherits the properties and elements of the shape class.

text orientation

Returns the orientation of the text inside the text box.
Can be one of the following:

- downward
- horizontal
- text orientation unset
- vertical east asian
- vertical, horizontal rotated east asian
- upward

Class: text frame

Plural

text frames

Represents the text frame in a shape object. Contains the text in the text frame as well as the properties that control the margins and orientation of the text frame.

Use the text frame property to return the text frame object for a shape. The text range property returns a text range object that represents the range of text inside the specified text frame. The following example adds text to the text frame of shape 1 in the active document.

set content of text range of text frame of shape 1 of active document to "My text"

Note Some shapes don’t support attached text (lines, freeforms, pictures, and OLE objects, for example). If you attempt to return or set properties that control text in a text frame for those objects, an error occurs.

Use the has text property to determine whether the text frame contains text, as shown in the following example.

repeat with s in (get shapes in active document)
    if has text of text frame of s is true then
        display dialog (content of text range of text frame of s) as string
    end if
end repeat
Text frames can be linked together so that the text flows from the text frame of one shape into the text frame of another shape. To link text frames, use the next text frame and previous text frame properties. The following example creates a text box (a rectangle with a text frame) and adds some text to it. It then creates another text box and links the two text frames together so that the text flows from the first text frame into the second one.

```
set myTB1 to make new text box at active document with properties {text ¬
    orientation:horizontal, left position:72, top:72, width:144, ¬
    height:36}
set content of text range of text frame of myTB1 to "This is some text. ¬
    This is some more text. And this is the text that should flow into ¬
    the next text box"
set myTB2 to make new text box at active document with properties {text ¬
    orientation:horizontal, left position:72, top:144, width:72, ¬
    height:36}
set next text frame of text frame of myTB1 to text frame of myTB2
```

Use the containing range property to return a text range object that represents the entire story that flows between linked text frames. The following example checks the spelling in the text in the third text box and any other text that's linked to it.

```
set myStory to (containing range of text frame of text box 3 ¬
    of active document)
check spelling myStory
```

**Properties**

**containing range**

Returns a text range object that represents the entire story in a series of shapes with linked text frames that the specified text frame belongs to. Read-only.

**has text**

True if the specified shape has text associated with it. Read-only.

**margin bottom**

Returns or sets the distance (in points) between the bottom of the text frame and the bottom of the inscribed rectangle of the shape that contains the text. Read/write.

**margin left**

Returns or sets the distance (in points) between the left edge of the text frame and the left edge of the inscribed rectangle of the shape that contains the text. Read/write.

**margin right**

Returns or sets the distance (in points) between the right edge of the text frame and the right edge of the inscribed rectangle of the shape that contains the text. Read/write.

**margin top**

Returns or sets the distance (in points) between the top of the text frame and the top of the inscribed rectangle of the shape that contains the text. Read/write.
next textframe

Returns the next text frame object. Read-only.

orientation

Returns or sets the orientation of the text inside the frame. Read/write.
Can be one of the following:

- downward
- horizontal
- text orientation unset
- upward
- vertical east asian
- vertical
- horizontal rotated east asian

overflowing

True if the text inside the specified text frame doesn't all fit within the frame. Read-only.

previous textframe

Returns the previous text frame object. Read-only.

text range

Returns a text range object that represents the portion of a document that's contained in the specified object. Read-only.

Class: threeD format

Represents a shape's three-dimensional formatting.

Use the threeD format property to return a threeD format object. The following example adds an oval to the active document and then specifies that the oval be extruded to a depth of 50 points and that the extrusion be purple.

set myShape to make new shape at active document with properties ¬
   { auto shape type:autoshape oval, left position:90, top:90, ¬
     width:90, height:90}
set visible of threeD format of myShape to true
set depth of threeD format of myShape to 50
set extrusion color of threeD format of myShape to (255, 100, 255) as RGB color
Remarks
You cannot apply three-dimensional formatting to some kinds of shapes, such as beveled shapes or multiple-disjoint paths. Most of the properties and commands of the threeD format object for such a shape will fail.

Properties

- **depth**
  Returns or sets the depth of the shape's extrusion. Can be a value from -600 through 9600 (positive values produce an extrusion whose front face is the original shape; negative values produce an extrusion whose back face is the original shape). Read/write.

- **extrusion color**
  Returns or sets an RGB color that represents the color of the shape's extrusion. Read/write.

- **perspective**
  True if the extrusion appears in perspective — that is, if the walls of the extrusion narrow toward a vanishing point. False if the extrusion is a parallel, or orthographic, projection — that is, if the walls don't narrow toward a vanishing point. Read/write.

- **preset extrusion direction**
  Returns the direction that the extrusion's sweep path takes away from the extruded shape (the front face of the extrusion). Read-only.
  Can be one of the following:
  - extrude bottom
  - extrude bottom left
  - extrude bottom right
  - extrude left
  - extrude none
  - extrude right
  - extrude top
  - extrude top left
  - extrude top right
  - preset extrusion direction unset

To set the value of this property, use the set extrusion direction command.
**preset lighting direction**

Returns or sets the position of the light source relative to the extrusion. Read/write.

Can be one of the following:

- light from bottom
- light from bottom left
- light from bottom right
- light from left
- light from none
- light from right
- light from top
- light from top left
- light from top right
- preset lighting direction unset

**Note** You won't see the lighting effects you set if the extrusion has a wireframe surface.

**preset lighting softness**

Returns or sets the intensity of the extrusion lighting. Read/write.

Can be one of the following:

- lighting bright
- lighting dim
- lighting normal
- lighting softness unset

**preset material**

Returns or sets the extrusion surface material. Read/write.

Can be one of the following:

- matte
- metal
- plastic
- wireframe
- preset material unset
**preset threeD format**

Returns the preset extrusion format. Each preset extrusion format contains a set of preset values for the various properties of the extrusion. If the extrusion has a custom format rather than a preset format, this property returns **preset threeD format unset**. Read-only.

Can be one of the following:

- **preset threeD format unset**
- format1
- format2
- format3
- format4
- format5
- format6
- format7
- format8
- format9
- format10
- format11
- format12
- format13
- format14
- format15
- format16
- format17
- format18
- format19
- format20

The values for this property correspond to the options (numbered from left to right, top to bottom) displayed when you click the 3-D button on the **Drawing** toolbar.

**Remarks**

To set the preset extrusion format, use the **set threeD format** command.

**rotation x**

Returns or sets the rotation of the extruded shape around the x-axis, in degrees. Can be a value from -90 through 90. A positive value indicates upward rotation; a negative value indicates downward rotation. Read/write.

To set the rotation of the extruded shape around the y-axis, use the **rotation y** property. To set the rotation of the extruded shape around the z-axis, use the **rotation** property of the shape object. To change the direction of the extrusion's sweep path without rotating the front face of the extrusion, use the **set extrusion direction** command.

**rotation y**

Returns or sets the rotation (in degrees) of the extruded shape around the y-axis. Can be a value from -90 through 90. A positive value indicates rotation to the left; a negative value indicates rotation to the right. Read/write.

To set the rotation of the extruded shape around the x-axis, use the **rotation x** property. To set the rotation of the extruded shape around the z-axis, use the **rotation** property of the shape object. To change the direction of the extrusion's sweep path without rotating the front face of the extrusion, use the **set extrusion direction** command.

**visible**

**True** if the specified object, or the formatting applied to it, is visible. Read/write.
Class: word art

Plural
word arts

Represents a word art object in the drawing layer of a document. The word art class inherits all the properties of the shape class, which includes all the inline shapes in the main story of a document or in all the headers and footers of a document.

Use word art index, where index is the name or the index number, to return a single word art object. This example adds WordArt that contains the text "Test Text" at the beginning of the active document.

make new word art at beginning of active document with properties {word art ¬
  text: "Test text", left position: (inches to points inches 1), ¬
  top: (inches to points inches 1)}

set preset word art effect of word art format of word art 1 of active ¬
  document to text effect11

set font name of word art format of word art 1 of active document ¬
  to "Arial Black"

set font size of word art format of word art 1 of active document to 36

set bold of word art format of word art 1 of active document to true

Properties
<Inheritance>  shape

Inherits the properties and elements of the shape class.

bold

True to set the font used in the WordArt to bold. Read-only.

font name

The name of the font used in the WordArt. Read-only.

font size

The size (in points) of the font used in the WordArt. Read-only.

italic

True to set the font used in the WordArt to italic. Read-only.
**Drawing Suite**

**preset word art effect**

Returns the style of the specified WordArt. The values for this property correspond to the formats in the **WordArt Gallery** dialog box (numbered from left to right, top to bottom). Read-only.

Can be one of the following:

<table>
<thead>
<tr>
<th>text effect1</th>
<th>text effect12</th>
<th>text effect22</th>
</tr>
</thead>
<tbody>
<tr>
<td>text effect2</td>
<td>text effect13</td>
<td>text effect23</td>
</tr>
<tr>
<td>text effect3</td>
<td>text effect14</td>
<td>text effect24</td>
</tr>
<tr>
<td>text effect4</td>
<td>text effect15</td>
<td>text effect25</td>
</tr>
<tr>
<td>text effect5</td>
<td>text effect16</td>
<td>text effect26</td>
</tr>
<tr>
<td>text effect6</td>
<td>text effect17</td>
<td>text effect27</td>
</tr>
<tr>
<td>text effect7</td>
<td>text effect18</td>
<td>text effect28</td>
</tr>
<tr>
<td>text effect8</td>
<td>text effect19</td>
<td>text effect29</td>
</tr>
<tr>
<td>text effect9</td>
<td>text effect20</td>
<td>text effect30</td>
</tr>
<tr>
<td>text effect10</td>
<td>text effect21</td>
<td>text effect unset</td>
</tr>
</tbody>
</table>

**word art format**

Returns the **word art format** object associated with the word art shape object. Read-only.

**word art text**

The text in the WordArt. Read-only

**Class: word art format**

Represents formatting options for WordArt objects.

Use the **word art format** property to return a **word art format** object. The following example sets the font name and formatting for word art 1 on the active document.

```plaintext
set font name of word art format of word art 1 of active document to "Courier New"
set bold of word art format of word art 1 of active document to true
set italic of word art format of word art 1 of active document to true
```

**Properties**

**alignment**

Returns or sets the alignment for the specified text effect. Read/write.

Can be one of the following:

- text effect alignment unset
- left text effect alignment
- centered text effect alignment
- right text effect alignment
- justify text effect alignment
- word justify text effect alignment
- stretch justify text alignment
**bold**

*True* if the text of the WordArt shape is formatted as bold. Read/write.

**font name**

Returns or sets the name of the font used by the WordArt shape. Read/write.

**font size**

Returns or sets the font size for the specified WordArt, in points. Read/write.

**italic**

*True* if the text of the WordArt shape is formatted as italic. Read/write.

**kerned pairs**

*True* if character pairs in the specified WordArt are kerned. Read/write.

**normalized height**

*True* if all characters (both uppercase and lowercase) in the specified WordArt are the same height. Read/write.

**preset shape**

Returns or sets the shape of the specified WordArt. Read/write.

Can be one of the following:

- **arch down curve**
- **arch down pour**
- **arch up curve**
- **arch up pour**
- **button curve**
- **button pour**
- **can down**
- **can up**
- **cascade down**
- **cascade up**
- **chevron down**
- **chevron up**
- **circle curve**
- **circle Pour**

<table>
<thead>
<tr>
<th>shape</th>
<th>effect</th>
<th>effect</th>
</tr>
</thead>
<tbody>
<tr>
<td>curve down</td>
<td>curve up</td>
<td>inflate bottom</td>
</tr>
<tr>
<td>deflate</td>
<td>deflate bottom</td>
<td>plain text</td>
</tr>
<tr>
<td>deflate</td>
<td>deflake top</td>
<td>slant down</td>
</tr>
<tr>
<td>deflate top</td>
<td>double wave1</td>
<td>sloan up</td>
</tr>
<tr>
<td>deflate</td>
<td>double wave2</td>
<td>stop</td>
</tr>
<tr>
<td>deflate</td>
<td>fade down</td>
<td>triangle down</td>
</tr>
<tr>
<td>deflate</td>
<td>fade left</td>
<td>triangle up</td>
</tr>
<tr>
<td>deflate</td>
<td>fade right</td>
<td>wave1</td>
</tr>
<tr>
<td>deflate</td>
<td>fade up</td>
<td>wave2</td>
</tr>
<tr>
<td>deflate</td>
<td>inflate</td>
<td></td>
</tr>
</tbody>
</table>

Setting the *preset word art effect* property automatically sets the *preset shape* property.
*preset word art effect*

Returns or sets the style of the specified WordArt. The values for this property correspond to the formats in the **WordArt Gallery** dialog box (numbered from left to right, top to bottom). Read/write.

Can be one of the following:

<table>
<thead>
<tr>
<th>text effect1</th>
<th>text effect2</th>
<th>text effect3</th>
<th>text effect4</th>
<th>text effect5</th>
<th>text effect6</th>
</tr>
</thead>
<tbody>
<tr>
<td>text effect7</td>
<td>text effect8</td>
<td>text effect9</td>
<td>text effect10</td>
<td>text effect11</td>
<td>text effect12</td>
</tr>
<tr>
<td>text effect13</td>
<td>text effect14</td>
<td>text effect15</td>
<td>text effect16</td>
<td>text effect17</td>
<td>text effect18</td>
</tr>
<tr>
<td>text effect19</td>
<td>text effect20</td>
<td>text effect21</td>
<td>text effect22</td>
<td>text effect23</td>
<td>text effect24</td>
</tr>
<tr>
<td>text effect25</td>
<td>text effect26</td>
<td>text effect27</td>
<td>text effect28</td>
<td>text effect29</td>
<td>text effect30</td>
</tr>
</tbody>
</table>

Setting the *preset word art effect* property automatically sets many other formatting properties of the specified shape.

*rotated chars*

**True** if characters in the specified WordArt are rotated 90 degrees relative to the WordArt's bounding shape. **False** if characters in the specified WordArt retain their original orientation relative to the bounding shape. Read/write.

If the WordArt has horizontal text, setting the *rotated chars* property to **true** rotates the characters 90 degrees counterclockwise. If the WordArt has vertical text, setting the *rotated chars* property to **false** rotates the characters 90 degrees clockwise. To switch between horizontal and vertical text flow, use the **toggle vertical text** command.

The **flip** command and **rotation** property of the **shape** object and the *rotated chars* property and **toggle vertical text** command all affect the character orientation and direction of text flow in a **word art** object. You may have to experiment to find out how to combine the effects of these properties and commands to get the result you want.

*tracking*

Returns or sets the ratio of the horizontal space allotted to each character in the specified WordArt to the width of the character. Can be a value from 0 (zero) through 5. (Large values for this property specify ample space between characters; values less than 1 can produce character overlap.) Read/write.
The following table gives the values of the *tracking* property that correspond to the settings available in the user interface.

<table>
<thead>
<tr>
<th>User interface setting</th>
<th>Equivalent tracking property value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Very Tight</td>
<td>0.8</td>
</tr>
<tr>
<td>Tight</td>
<td>0.9</td>
</tr>
<tr>
<td>Normal</td>
<td>1.0</td>
</tr>
<tr>
<td>Loose</td>
<td>1.2</td>
</tr>
<tr>
<td>Very Loose</td>
<td>1.5</td>
</tr>
</tbody>
</table>

*word art text*

The text in the WordArt. Read-only

**Class: wrap format**

Represents all the properties for wrapping text around a shape.

Use the *wrap format* property to return the *wrap format* object. The following example adds an oval to the active document and specifies that document text wrap around the left and right sides of the square that circumscribes the oval. There will be a 0.1-inch margin between the document text and the top, bottom, left side, and right side of the square.

```plaintext
set myOval to make new oval at beginning of active document with properties ¬
    { auto shape type: autoshape oval, left position: 36, top: 36, ¬
      width: 100 height: 35 }
set wrap type of wrap format of myOval to wrap square
set wrap side of wrap format of myOval to wrap both
set distance top of wrap format of myOval to (inches to points inches 0.1)
set distance bottom of wrap format of myOval to (inches to points inches 0.1)
set distance left of wrap format of myOval to (inches to points inches 0.1)
set distance right of wrap format of myOval to (inches to points inches 0.1)
```

**Properties**

*allow overlap*

Returns or sets a value that specifies whether a given shape can overlap other shapes. Read/write.

*distance bottom*

Returns or sets the distance (in points) between the document text and the bottom edge of the text-free area surrounding the specified shape. The size and shape of the specified shape, together with the values of the *wrap type* and *wrap side* properties, determine the size and shape of this text-free area. Read/write.
returns or sets the distance (in points) between the document text and the left edge of the text-free area surrounding the specified shape. the size and shape of the specified shape, together with the values of the wrap type and wrap side properties, determine the size and shape of this text-free area. read/write.

distance right

returns or sets the distance (in points) between the document text and the right edge of the text-free area surrounding the specified shape. the size and shape of the specified shape, together with the values of the wrap type and wrap side properties, determine the size and shape of this text-free area. read/write.

distance top

returns or sets the distance (in points) between the document text and the top edge of the text-free area surrounding the specified shape. the size and shape of the specified shape, together with the values of the wrap type and wrap side properties, determine the size and shape of this text-free area. read/write.

wrap side

returns or sets a value that indicates whether the document text should wrap on both sides of the specified shape, on either the left or right side only, or on the side of the shape that’s farthest from the page margin. if the text wraps on only one side of the shape, there’s a text-free area between the other side of the shape and the page margin. read/write.

can be one of the following:

- wrap both
- wrap largest
- wrap left
- wrap right

wrap type

returns the wrap type for the specified shape. read/write.

can be one of the following:

- wrap none
- wrap square
- wrap through
- wrap tight
- wrap top bottom.
**Commands**

activate object ........................................................................................................................................... 395
apply .......................................................................................................................................................... 396
automatic length .................................................................................................................................... 396
break forward link ................................................................................................................................ 397
convert to frame ..................................................................................................................................... 398
convert to inline shape ........................................................................................................................... 398
convert to shape .................................................................................................................................... 398
custom drop ........................................................................................................................................... 399
custom length ......................................................................................................................................... 399
flip ........................................................................................................................................................ 400
get border ............................................................................................................................................... 400
one color gradient ................................................................................................................................... 401
patterned ................................................................................................................................................ 401
pick up ................................................................................................................................................... 402
preset drop .............................................................................................................................................. 402
preset gradient ...................................................................................................................................... 403
preset textured ....................................................................................................................................... 404
reroute connections .............................................................................................................................. 404
reset ...................................................................................................................................................... 405
reset rotation ........................................................................................................................................... 405
scale height ............................................................................................................................................ 406
scale width ............................................................................................................................................. 406
set extrusion direction ............................................................................................................................ 407
set shapes default properties ................................................................................................................. 407
set threeD format .................................................................................................................................... 407
solid ........................................................................................................................................................ 408
toggle vertical text ................................................................................................................................. 409
two color gradient ................................................................................................................................... 410
user picture .............................................................................................................................................. 411
user textured .......................................................................................................................................... 411
valid link target ..................................................................................................................................... 412
z order .................................................................................................................................................... 412

**Command: activate object**

Activates the specified object.

**Syntax**

activate object  shape/inline shape  Required. An expression that returns a *shape* or *inline shape* object.

**Example**

The following example activates the first inline shape in the active document.

activate object inline shape 1 of active document
Command: apply

Applies to the specified shape formatting that has been copied using the pick up command.

Syntax

apply  shape  Required. An expression that returns a shape object.

If formatting for the shape object has not previously been copied using the pick up command, using the apply command generates an error.

Example

This example copies the formatting of shape 1 on the active document and applies the copied formatting to shape 2 on the same document.

pick up shape 1 of active document
apply shape 2 of active document

Command: automatic length

Specifies that the first segment of the callout line (the segment attached to the text callout box) be scaled automatically when the callout is moved. To specify that the first segment of the callout line retain the fixed length returned by the callout format length property whenever the callout is moved, use the custom length command. Applies only to callouts whose lines consist of more than one segment (types callout three and callout four).

Syntax

automatic length  shape/callout format  Required. An expression that returns a shape or callout format object.

Applying this command sets the auto length property to true.

Example

This example toggles between an automatically scaling first segment and one with a fixed length for the callout line for callout 1 on myDocument.

set myDocument to the active document
if auto length of callout format of callout 1 of myDocument is true then
    custom length callout 1 of myDocument length 50
else
    automatic length callout 1 of myDocument
end if
Command: break forward link

Breaks the forward link for the specified text frame, if such a link exists.

Syntax

`break forward link`  `text frame`  Required. An expression that returns a `text frame` object.

Applying this command to a shape in the middle of a chain of shapes with linked text frames will break the chain, leaving two sets of linked shapes. All of the text, however, will remain in the first series of linked shapes.

Example

This example creates a new document, adds a chain of three linked text boxes to it, and then breaks the link after the second text box.

```
make new document
set myTB to make new text box at active document with properties ¬
    {t e x t  o r i e n t a t i o n : h o r i z o n t a l , ¬
     l e f t  p o s i t i o n : ( i n c h e s  t o  p o i n t s  i n c h e s  1.5) , ¬
     t o p : ( i n c h e s  t o  p o i n t s  i n c h e s  0.5) , ¬
     w i d t h : ( i n c h e s  t o  p o i n t s  i n c h e s  1) , ¬
     h e i g h t : ( i n c h e s  t o  p o i n t s  i n c h e s  0.5) }
set content of text range of text frame of myTB to "This is some text. This ¬
is some more text. This is even more text."
set myTB2 to make new text box at active document with properties ¬
    {t e x t  o r i e n t a t i o n : h o r i z o n t a l , ¬
     l e f t  p o s i t i o n : ( i n c h e s  t o  p o i n t s  i n c h e s  1.5) , ¬
     t o p : ( i n c h e s  t o  p o i n t s  i n c h e s  1.5) , ¬
     w i d t h : ( i n c h e s  t o  p o i n t s  i n c h e s  1) , ¬
     h e i g h t : ( i n c h e s  t o  p o i n t s  i n c h e s  0.5) }
set myTB3 to make new text box at active document with properties ¬
    {t e x t  o r i e n t a t i o n : h o r i z o n t a l , ¬
     l e f t  p o s i t i o n : ( i n c h e s  t o  p o i n t s  i n c h e s  1.5) , ¬
     t o p : ( i n c h e s  t o  p o i n t s  i n c h e s  2.5) , ¬
     w i d t h : ( i n c h e s  t o  p o i n t s  i n c h e s  1) , ¬
     h e i g h t : ( i n c h e s  t o  p o i n t s  i n c h e s  0.5) }
set next textframe of text frame of myTB to text frame of myTB2
set next textframe of text frame of myTB2 to text frame of myTB3
display dialog "Text boxes 1, 2, and 3 are linked."
break forward link text frame of myTB2
```
Command: convert to frame
Converts the specified shape to a frame. Returns a frame object that represents the new frame.

Syntax
convert to frame shape  Required. An expression that returns a shape object.

Shapes that don't support attached text cannot be converted to frames. For pictures, use the convert to inline shape command.

In the current version of Word, frames have been replaced by text boxes.

Example
This example creates a text box using the selected text, and then it converts the text box to a frame.
if selection type of selection is selection normal then
  create textbox selection
  convert to frame text box 1 of active document
end if

Command: convert to inline shape
Converts the specified shape in the drawing layer of a document to an inline shape in the text layer.
You can convert only shapes that represent pictures. This command returns an inline shape object that represents the picture.

Syntax
convert to inline shape shape  Required. An expression that returns a shape object.

Shapes that support attached text cannot be converted to inline shapes. For these shapes, use the convert to frame command.

Example
This example converts each picture in MyDoc.doc to an inline shape.
repeat with s in (get shapes in document "MyDoc.doc")
  if shape type of s is shape type picture then
    convert to inline shape s
  end if
end repeat

Command: convert to shape
Converts an inline shape to a free-floating shape. Returns a shape object that represents the new shape.

Syntax
convert to shape inline shape  Required. An expression that returns an inline shape object.

Example
This example converts the first inline shape in the active document to a floating shape.
convert to shape inline shape 1 of active document
Command: custom drop

Sets the vertical distance (in points) from the edge of the text bounding box to the place where the callout line attaches to the text box. This distance is measured from the top of the text box unless the auto attach property is set to true and the text box is to the left of the origin of the callout line (the place that the callout points to), in which case the drop distance is measured from the bottom of the text box.

Syntax

custom drop callout/callout format  Required. An expression that returns a callout or callout format object.

drop  small real  Required. The drop distance, in points.

If the preset drop command was previously used to set the drop for the specified callout, use the preset drop command to set the DropType property of the callout to drop custom before using the custom drop command so that the custom drop setting takes effect.

Example

This example cancels any preset drop that's been set for callout 1 in myDocument, sets the custom drop distance to 14 points, and specifies that the drop distance always be measured from the top. For the example to work, shape one must be a callout.

set myDocument to active document
preset drop callout 1 of myDocument DropType drop custom
custom drop callout 1 of myDocument drop 14
set auto attach of callout format of callout 1 of myDocument to false

Command: custom length

Specifies that the first segment of the callout line (the segment attached to the text callout box) retain a fixed length whenever the callout is moved. To specify that the first segment of the callout line be scaled automatically whenever the callout is moved, use the automatic length command. Applies only to callouts whose lines consist of more than one segment (types callout three and callout four).

Syntax

custom length  callout/callout format  Required. An expression that returns a callout or callout format object.

length  small real  Required. The length (in points) of the first segment of the callout.

Applying this command sets the auto length property to false and sets the callout format length property to the value specified for the length argument.
Example
This example toggles between an automatically scaling first segment and one with a fixed length for the callout line for callout 1 on myDocument.

```pascal
set myDocument to the active document
if auto length of callout format of callout 1 of myDocument is true then
custom length callout 1 of myDocument length 50
else
automatic length callout 1 of myDocument
end if
```

Command: flip
Flips the specified shape around its horizontal or vertical axis.

Syntax

```pascal
flip shape Required. An expression that returns a shape object.

flip command enumeration Required. Specifies whether the shape is to be flipped horizontally or vertically. Can be either of the following: flip horizontal or flip vertical.
```

Example
This example adds a triangle to myDocument, duplicates the triangle, and then flips the duplicate triangle vertically and makes it red.

```pascal
set myDocument to active document
make new shape at myDocument with properties ¬
{ auto shape type:autoshape right triangle, left position:10, top:10 ¬
  width:50, height:50 }
duplicate shape 1 of myDocument
set fore color of fill format of shape 2 of myDocument to ({255, 0, 0} ¬
as RGB color)
flip shape 2 of myDocument flip command flip vertical
```

Command: get border
Returns the specified border object.

Syntax

```pascal
get border inline shape Required. An expression that returns an inline shape object.

which border enumeration Required. Can be one of the following: border top, border left,
border bottom, border right, border horizontal, border vertical, border diagonal down, or
border diagonal up.
```
Command: *one color gradient*

Sets the specified shape or fill format to a one-color gradient.

**Syntax**

one color gradient  shape/fill format  Required. An expression that returns a shape or fill format object.

  gradient style  enumeration  Required. The gradient style. Can be one of the following: gradient unset, diagonal down gradient, diagonal up gradient, from center gradient, from corner gradient, horizontal gradient, or vertical gradient. (The constant from title gradient is used only in PowerPoint.)

  gradient variant  integer  Required. The gradient variant. Can be a value from 1 to 4, corresponding to the four variants on the Gradient tab in the Fill Effects dialog box. If gradient style is from center gradient, this argument can be either 1 or 2.

  gradient degree  small real  Required. The gradient degree. Can be a value from 0.0 (dark) to 1.0 (light).

**Example**

This example adds a rectangle with a one-color gradient fill to myDocument.

```plaintext
set myDocument to active document
make new shape at myDocument with properties ¬
    { auto shape type: autoshape rectangle, left position: 90, top: 90 ¬
      width: 90, height: 80 }  
set fore color of fill format of shape 1 of myDocument to ({0, 128, 128} as ¬
    RGB color)
one color gradient shape 1 of newDoc gradient style horizontal gradient ¬
    gradient variant 1 gradient degree 1
```

Command: *patterned*

Sets the specified fill to a pattern.

**Syntax**

patterned  shape/fill format  Required. An expression that returns a shape or fill format object.

  pattern  enumeration  Required. The pattern to be used for the specified fill. Can be any of the patterns listed under the pattern property of the fill format class.

Use the back color and fore color properties to set the colors used in the pattern.
Example

This example adds an oval with a patterned fill to myDocument.

```plaintext
set myDocument to active document
make new shape at myDocument with properties ¬
    {auto shape type:autoshape oval, left position:60, top:60, width:80 ¬
        height:40}
set fore color of fill format of shape 1 of myDocument to ({128, 0, 0} as ¬
    RGB color)
set back color of fill format of shape 1 of myDocument to ({0, 0, 255} as ¬
    RGB color)
patterned shape 1 of myDocument pattern dark vertical pattern
```

Command: pick up

Copies the formatting of the specified shape. Use the apply command to apply the copied formatting to another shape.

Syntax

```plaintext
pick up  shape  Required. An expression that returns a shape object.
```

Example

This example copies the formatting of shape 1 on the active document and applies the copied formatting to shape 2 on the same document.

```plaintext
pick up shape 1 of active document
apply shape 2 of active document
```

Command: preset drop

Specifies whether the callout line attaches to the top, bottom, or center of the callout text box or whether it attaches at a point that's a specified distance from the top or bottom of the text box.

Syntax

```plaintext
preset drop  callout/callout format  Required. An expression that returns a callout or callout format object.
```

DropType  Required. The starting position of the callout line relative to the text bounding box. Can be one of the following: drop unset, drop bottom, drop center, drop custom, or drop top.

If you specify drop custom, the values of the drop and auto attach properties and the relative positions of the callout text box and callout line origin (the place that the callout points to) are used to determine where the callout line attaches to the text box.
Example

This example specifies that the callout line attach to the top of the text bounding box for callout 1 on myDocument.

```plaintext
set myDocument to active document
preset drop callout 1 of myDocument DropType drop top
```

This example toggles between two preset drops for callout 1 on myDocument.

```plaintext
set myDocument to active document
if drop type of callout format of callout 1 of myDocument is drop top then
    preset drop callout 1 of myDocument DropType drop bottom
else if drop type of callout format of callout 1 of myDocument is ¬
    drop bottom then
    preset drop callout 1 of myDocument DropType drop top
end if
```

Command: preset gradient

Sets the specified fill to a preset gradient.

Syntax

```plaintext
preset gradient  shape/fill format  Required. An expression that returns a shape or fill format object.

style  enumeration  Required. The gradient style. Can be one of the following: preset gradient
    unset, diagonal down gradient, diagonal up gradient, from center gradient, from corner
gradient, horizontal gradient, or vertical gradient. (The constant from title gradient is used
only in PowerPoint.)

gradient variant  integer  Required. The gradient variant. Can be a value from 1 to 4,
corresponding to the four variants on the Gradient tab in the Fill Effects dialog box. If style is
from center gradient, this argument can be either 1 or 2.

preset gradient type  enumeration  Required. The gradient type. Can be one of the following:

gradient brass gradient calm
    gradient water
    gradient chrome
    gradient chrome2
    gradient daybreak
    gradient desert
    gradient early sunset
    gradient fire
    gradient fog
    gradient gold
    gradient gold2
    gradient horizon
    gradient late sunset
    gradient mahogany gradient
    gradient moss
    gradient nightfall
    gradient ocean
    gradient parchment
    gradient peacock
    gradient rainbow
    gradient rainbow2
    gradient sapphire
    gradient silver
    gradient wheat
    preset gradient unset
```
Example
This example adds a rectangle with a preset gradient fill to myDocument.
Set myDocument to active document
make new shape at myDocument with properties ¬
  {auto shape type:aut}
preset gradient shape 1 of myDocument style horizontal gradient ¬
  gradient variant 1 preset gradient type gradient brass

Command: preset textured
Sets the specified fill to a preset texture.

Syntax
preset textured  shape/fill format  Required. An expression that returns a shape or fill format
  object.

  preset texture  enumeration  Required. The preset texture. Can be one of the following:

  texture blue tissue paper      texture paper bag
  texture bouquet                texture papyrus
  texture brown marble           texture parchment
  texture canvas                 texture pink tissue paper
  texture cork                   texture purple mesh
  texture denim                  texture recycled paper
  texture fish fossil            texture sand
  texture granite                texture stationery
  texture green marble          texture walnut
  texture medium wood            texture water droplets
  texture newsprint              texture white marble
  texture oak                    texture woven mat

Example
This example adds a cylinder with a green-marble textured fill to myDocument.
set myDocument to active document
make new shape at myDocument with properties ¬
  {auto shape type: autoshape can, left position:90, top:90, ¬
  width:40, height:80}
preset textured shape 1 of myDocument preset texture texture green marble

Command: reroute connections
Reroutes connectors so that they take the shortest possible path between the shapes they connect. To do this, the reroute connections command may detach the ends of a connector and reattach them to different connecting sites on the connected shapes.

This command reroutes all connectors attached to the specified shape; if the specified shape is a connector, it's rerouted.

The reroute connections command will not run in Word.
Command: reset

Removes changes that were made to an inline shape.

Syntax

reset inline shape  Required. An expression that returns an inline shape object.

Example

This example inserts a picture as an inline shape, changes the brightness, and then resets the picture to its original brightness.

set aInline to make new inline picture at the beginning of active document with properties {file name:"Macintosh HD:Users:Shared:Bubbles.gif"}
set brightness of picture format of aInline to 0.5
display dialog "Changing brightness back"
reset aInline

Command: reset rotation

Resets the extrusion rotation around the x-axis and the y-axis to 0 (zero) so that the front of the extrusion faces forward. This command doesn't reset the rotation around the z-axis.

Syntax

reset rotation shape/threeD format  Required. An expression that returns a shape or threeD format object.

To set the extrusion rotation around the x-axis and the y-axis to anything other than 0 (zero), use the rotation x and rotation y properties of the threeD format object. To set the extrusion rotation around the z-axis, use the rotation property of the shape object that represents the extruded shape.

Example

This example resets the rotation around the x-axis and the y-axis to 0 (zero) for the extrusion of shape 1 on myDocument.

set myDocument to active document
reset rotation threeD format of shape 1 of myDocument
Command: scale height

Scales the height of the picture by a specified factor. You can indicate whether you want to scale the shape relative to the original size or relative to the current size.

Syntax

scale height  picture  Required. An expression that returns a picture object.

factor  small real  Required. Specifies the ratio between the height of the shape after you resize it and the current or original height. For example, to make a rectangle 50 percent larger, specify 1.5 for this argument.

relative to original size  Boolean  Required. True to scale the shape relative to its original size. False to scale it relative to its current size.

[scale  enumeration]  Optional. The part of the shape that retains its position when the shape is scaled. Can be one of the following: scale from bottom right, scale from middle, or scale from top left. The default value is scale from top left.

Example

This example scales all pictures on myDocument to 175 percent of their original height and width.

set myDocument to active document
repeat with s in (get pictures in myDocument)
    scale height s factor 1.75 with relative to original size
    scale width s factor 1.75 with relative to original size
end repeat

Command: scale width

Scales the width of the picture by a specified factor. You can indicate whether you want to scale the shape relative to the original size or relative to the current size.

Syntax

scale width  picture  Required. An expression that returns a picture object.

factor  small real  Required. Specifies the ratio between the width of the shape after you resize it and the current or original width. For example, to make a rectangle 50 percent larger, specify 1.5 for this argument.

relative to original size  Boolean  Required. True to scale the shape relative to its original size. False to scale it relative to its current size.

[scale  enumeration]  Optional. The part of the shape that retains its position when the shape is scaled. Can be one of the following: scale from bottom right, scale from middle, or scale from top left. The default value is scale from top left.
Drawing Suite

Example
This example scales all pictures on myDocument to 175 percent of their original height and width.
set myDocument to active document
repeat with s in (get pictures in myDocument)
   scale height s factor 1.75 with relative to original size
   scale width s factor 1.75 with relative to original size
end repeat

Command: set extrusion direction
Sets the direction that the extrusion's sweep path takes away from the extruded shape.

Syntax
set extrusion direction  shape/threeD format  Required. An expression that returns a shape or threeD format object.
   preset extrusion direction  enumeration  Required. Specifies the extrusion direction. Can be one of the following: preset extrusion direction unset, extrude bottom, extrude bottom left, extrude bottom right, extrude left, extrude none, extrude right, extrude top, extrude top left, or extrude top right.

This command sets the preset extrusion direction property to the direction specified by the preset extrusion direction argument.

Example
This example specifies that the extrusion for shape 1 on myDocument should extend toward the top of the shape and that the lighting for the extrusion should come from the left.
set myDocument to active document
set visible of threeD format of shape 1 of myDocument to true
set extrusion direction shape 1 of myDocument to preset extrusion direction extrude top
set preset lighting direction of threeD format of shape 1 of myDocument to light from left

Command: set shapes default properties
Applies the formatting of the specified shape to a default shape for that document. New shapes inherit many of their attributes from the default shape.

Syntax
set shapes default properties  shape  Required. An expression that returns a shape object.
Example
This example adds a rectangle to myDocument, formats the rectangle's fill, applies the rectangle's formatting to the default shape, and then adds another (smaller) rectangle to the document. The second rectangle has the same fill as the first one.

set myDocument to make new document
make new shape at the beginning of myDocument with properties ¬
    { auto shape type: autoshape rectangle, left position: 5, top: 5, ¬
      width: 80, height: 60 }
set fore color of fill format of shape 1 of myDocument to ({0, 0, 255} ¬
    as RGB color)
set back color of fill format of shape 1 of myDocument to ({0, 204, 255} ¬
    as RGB color)
patterned shape 1 of myDocument pattern horizontal brick pattern
--sets formatting of default shapes
set shapes default properties shape 1 of myDocument
--makes a new shape with default formatting
make new shape at the end of myDocument with properties ¬
    { auto shape type: autoshape rectangle, left position: 90, top: 90, ¬
      width: 40, height: 30 }

Command: set threeD format
Sets the preset extrusion format. Each preset extrusion format contains a set of preset values for the various properties of the extrusion.

Syntax

set threeD format  shape/threeD format  Required. An expression that returns a threeD format object.

preset threeD format  enumeration  Required. Specifies a preset extrusion format that corresponds to one of the options (numbered from left to right, top to bottom) displayed when you click the 3-D button on the Drawing toolbar. Can be one of the following: preset threeD format unset, format1, format2, format3, format4, format5, format6, format7, format8, format9, format10, format11, format12, format13, format14, format15, format16, format17, format18, format19, or format20. Note that specifying preset threeD format unset for this argument causes an error.

This command sets the preset threeD format property to the format specified by the preset threeD format argument.
Example

This example adds an oval to myDocument and sets its extrusion format to 3D Style 12.

```plaintext
set myDocument to make new document
make new shape at the beginning of myDocument with properties ¬
    {auto shape type: autoshape oval, left position: 30, top: 30, ¬
     width: 50, height: 25}
set visible of threeD format of shape 1 of myDocument to true
set threeD format shape 1 of newDoc preset threeD format format12
```

Command: solid

Sets the specified fill to a uniform color. Use this command to convert a gradient, textured, patterned, or background fill to a solid fill.

Syntax

```plaintext
solid shape/fill format  
```

Required. An expression that returns a shape or fill format object.

Example

This example converts all fills on myDocument to uniform red fills.

```plaintext
set myDocument to active document
repeat with s in (get shapes in myDocument)
    solid s
    set fore color of fill format of s to {{255, 0, 0} as RGB color)
end repeat
```

Command: toggle vertical text

Switches the text flow in the specified WordArt from horizontal to vertical, or vice versa.

Syntax

```plaintext
toggle vertical text word art format  
```

Required. An expression that returns a word art format object.

Using the toggle vertical text command swaps the values of the width and height properties of the shape object that represents the word art object and leaves the left position and top properties unchanged.

The following commands and properties all affect the character orientation and direction of text flow in a shape object that represents WordArt. You may have to experiment to find out how to combine the effects of these properties and commands to get the result you want.

- **flip** command
- **rotation** property of the shape object
- **rotated chars** property of the word art object
- **toggle vertical text** command of the word art object
Example
This example adds WordArt that contains the text "Test" to myDocument and switches from horizontal text flow (the default for the specified word art effect, text effect1) to vertical text flow.
set myDocument to make new document
set newWordArt to make new word art at the beginning of myDocument with properties {preset word art effect: text effect1, word art text: "Test", font name: "Arial Black", font size: 36, bold: false, italic: false, left position: 100, top: 100}
toggle vertical text word art format of newWordArt

Command: two color gradient
Sets the specified fill to a two-color gradient.

Syntax
two color gradient  

shape/fill format  Required. An expression that returns a fill format object.

gradient style  Required. The gradient style. Can be one of the following: gradient unset, diagonal down gradient, diagonal up gradient, from center gradient, from corner gradient, horizontal gradient, or vertical gradient. (The constant from title gradient is used only in PowerPoint.)

gradient variant  Required. The gradient variant. Can be a value from 1 to 4, corresponding to the four variants on the Gradient tab in the Fill Effects dialog box. If style is from center gradient, this argument can be either 1 or 2.

Example
This example adds a rectangle with a two-color gradient fill to myDocument and sets the background and foreground color for the fill.
set myDocument to make new document
make new shape at the beginning of myDocument with properties
    {auto shape type: auto shape rectangle, left position: 0, top: 0, width: 40, height: 80}
set fore color of fill format of shape 1 of myDocument to
    ({128, 0, 0} as RGB color)
set back color of fill format of shape 1 of myDocument to
    ({0, 170, 170} as RGB color)
two color gradient shape 1 of newDoc gradient style horizontal gradient gradient variant 1
**Command: user picture**

Fills the specified shape with one large image. To fill the shape with small tiles of an image, use the **user textured** command.

**Syntax**

```
user picture  shape/fill format  Required. An expression that returns a shape or fill format object.
picture file  Unicode text  Required. The name of the picture file.
```

**Example**

This example adds two rectangles to `myDocument`. The rectangle on the left is filled with one large image of the picture in Tiles.gif; the rectangle on the right is filled with many small tiles of the picture in Tiles.gif.

```
set myDocument to make new document
set myShape1 to make new shape at the beginning of myDocument with properties
  {auto shape type:autoshape rectangle, left position:0, top:0, ¬
   width:200, height:100}
user picture myShape1 picture file "Macintosh HD:Users:Shared:Tiles.gif"
set newShape to make new shape at the beginning of myDocument with properties
  {auto shape type:autoshape rectangle, left position:300, top:0, ¬
   width:200, height:100}
user textured newShape texture file "Macintosh HD:Users:Shared:Tiles.gif"
```

**Command: user textured**

Fills the specified shape with small tiles of an image. To fill the shape with one large image, use the **user picture** command.

**Syntax**

```
user textured  shape/fill format  Required. An expression that returns a shape or fill format object.
texture file  Unicode text  Required. The name of the picture file.
```

**Example**

This example adds two rectangles to `myDocument`. The rectangle on the left is filled with one large image of the picture in Tiles.gif; the rectangle on the right is filled with many small tiles of the picture in Tiles.gif.

```
set myDoc to make new document
set myShape1 to make new shape at the beginning of myDoc with properties
  {auto shape type:autoshape rectangle, left position:0, top:0, ¬
   width:200, height:100}
user picture myShape1 picture file "Macintosh HD:Users:Shared:Tiles.gif"
set newShape to make new shape at the beginning of myDoc with properties
  {auto shape type:autoshape rectangle, left position:300, top:0, ¬
   width:200, height:100}
user textured newShape texture file "Macintosh HD:Users:Shared:Tiles.gif"
```
Command: valid link target

Determines whether the text frame of one shape can be linked to the text frame of another shape. Returns true if target textframe is a valid target. Returns false if target textframe already contains text or is already linked, or if the shape doesn't support attached text.

Syntax

valid link target text frame Required. An expression that returns a text frame object.

  target textframe text frame Required. The target text frame to which you'd like to link the text frame returned by the expression.

Example

This example checks to see whether the text frames for the first and second shapes in the active document can be linked to each other. If so, the example links the two text frames.

set myFrame1 to text frame of shape 1 of active document
set myFrame2 to text frame of shape 2 of active document
if (valid link target myFrame1 target textframe myFrame2 then
    set next textframe of myFrame1 to myFrame2
else
    display dialog "The text frames can't be linked"
end if

Command: z order

Moves the specified shape in front of or behind other shapes in the document (that is, changes the shape's position in the z-order).

Syntax

z order shape Required. An expression that returns a shape object.

  z order command enumeration Required. Specifies where to move the specified shape relative to the other shapes. Can be one of the following: bring shape forward, bring shape in front of text, bring shape to front, send shape backward, send shape behind text, or send shape to back.

Use the z order position property to determine a shape's current position in the z-order.
Class: character

Plural

characters

A list of characters in a selection, range, or document. Each character is an element of a text range object that represents one character.

Use characters to return all the characters in a text range. The following example displays how many characters are selected.

display dialog ((count of characters in selection) as string) & ¬
" characters are selected."

To return a text range object that represents one character, use character index, where index is the index number. The index number represents the position of a character in the text range. The following example formats the first letter in the selection as 24-point bold.

set char1 to character 1 of selection
set bold of char1 to True
set font size of font object of char1 to 24

Properties

<Inheritance> text range

Inherits the properties and elements of the text range class.
Class: grammatical error

Plural

grammatical errors

A list of grammatical errors for the specified document or range. Each item in the grammatical error list is a text range object that represents one grammatical error.

Use grammatical errors to return the list of all grammatical errors in a document or text range. The following example counts the spelling and grammatical errors in the active document and displays the results in a message box.

set pr1 to spelling errors of active document
set sc to count of pr1
set pr2 to grammatical errors of active document
set gc to count of pr2
display dialog "Spelling errors: " & sc & "\r" & "Grammatical errors: " & gc

To return a single grammatical error (represented by a text range object, use grammatical error index, where index is the index number. The following example displays the sentence that contains the first grammatical error in the active document.

set myRange to content of grammatical error 2 of active document
display dialog myRange

Properties

<Inheritance> text range

Inherits the properties and elements of the text range class.

Class: paragraph

Plural

paragraphs

Elements

tab stop

Represents a single paragraph in a selection, range, or document. The paragraphs list includes all the paragraphs in a selection, range, or document.

To return a single paragraph object, use paragraph index, where index is the index number. The following example right aligns the first paragraph in the active document.

set alignment of paragraph 1 of active document to align paragraph right

To add a new, blank paragraph to a document, use the insert paragraph command. When you use the insert paragraph command, the specified range is replaced by the new paragraph.

insert paragraph at text object of paragraph 1 of active document
Text Suite

Properties

add space between east asian and alpha

True if Word is set to automatically add spaces between Japanese and Latin text for the specified paragraphs. Read/write.

add space between east asian and digit

True if Word is set to automatically add spaces between Japanese text and numbers for the specified paragraphs. Read/write.

alignment

Returns or sets the alignment for the specified paragraphs. Read/write.

Can be one of the following:

- align paragraph left
- align paragraph center
- align paragraph right
- align paragraph distribute
- align paragraph justify

auto adjust right indent

True if Word is set to automatically adjust the right indent for the specified paragraphs if you’ve specified a set number of characters per line. Read/write.

base line alignment

Returns or sets the baseline font alignment. Read/write.

Can be one of the following:

- baseline align auto
- baseline align baseline
- baseline align center
- baseline align east asian50
- baseline align top

border options

Returns a border options object that represents all the borders for the specified object. Read-only.
**Text Suite**

*character unit first line indent*

Returns or sets the value (in characters) for a first-line or hanging indent. Use a positive value to set a first-line indent, and use a negative value to set a hanging indent. Read/write.

*character unit left indent*

Returns or sets the left indent value (in characters) for the specified paragraphs. Read/write.

*character unit right indent*

Returns or sets the right indent value (in characters) for the specified paragraphs. Read/write.

*disable line height grid*

**True** if Word aligns characters in the specified paragraphs to the line grid when a set number of lines per page is specified. Read/write.

*drop cap*

Returns a **drop cap** object that represents a dropped capital letter for the specified paragraph. Read-only.

*first line indent*

Returns or sets the value (in points) for a first line or hanging indent. Use a positive value to set a first-line indent, and use a negative value to set a hanging indent. Read/write.

*half width punctuation on top of line*

**True** if Word changes punctuation symbols at the beginning of a line to half-width characters for the specified paragraphs. Read/write.

*hanging punctuation*

**True** if hanging punctuation is enabled for the specified paragraphs. Read/write.

*hyphenation*

**True** if the specified paragraphs are included in automatic hyphenation. **False** if the specified paragraphs are to be excluded from automatic hyphenation. Read/write.

*keep together*

**True** if all lines in the specified paragraphs remain on the same page when Word repaginates the document. Read/write.

*keep with next*

**True** if the specified paragraph remains on the same page as the paragraph that follows it when Word repaginates the document. Read/write.
Text Suite

*line spacing*

Returns or sets the line spacing (in points) for the specified paragraphs. Read/write.

The *line spacing* property can be set after the *line spacing rule* property has been set to *line space at least*, *line space exactly*, or *line space multiple*. If *line space at least* is used, the line spacing can be greater than or equal to, but never less than, the specified *line spacing* value. If *line space exactly* is used, the line spacing never changes from the specified *line spacing* value, even if a larger font is used within the paragraph. If *line spacing multiple* is used, a *line spacing* property value must be specified, in points.

To convert a number of lines to the corresponding value in points, use the *lines to points* command. For example, `lines to points lines 2` returns the value 24.

*line spacing rule*

Returns or sets the line spacing for the specified paragraphs. Read/write.

Can be one of the following:

- *line space1 pt5*
- *line space at least*
- *line space double*
- *line space exactly*
- *line space multiple*
- *line space single*

Use *line space single*, *line space1 pt5*, or *line space double* to set the line spacing to one of these values. To set the line spacing to an exact number of points or to a multiple number of lines, you must also set the *line spacing* property.

*line unit after*

Returns or sets the amount of spacing (in gridlines) after the specified paragraphs. Read/write.

*line unit before*

Returns or sets the amount of spacing (in gridlines) before the specified paragraphs. Read/write.

*no line number*

*True* if line numbers are repressed for the specified paragraphs. Read/write.

Use the *line numbering* property to set line numbers.
Text Suite

**outline level**

Returns or sets the outline level for the specified paragraphs. Read/write.

Can be one of the following:

- **outline level1**
- **outline level2**
- **outline level3**
- **outline level4**
- **outline level5**
- **outline level6**
- **outline level7**
- **outline level8**
- **outline level9**
- **outline level body text**

If a paragraph has a heading style applied to it (Heading 1 through Heading 9), the outline level is the same as the heading style and cannot be changed.

Outline levels are visible only in outline view or the document map pane.

**page break before**

**True** if a page break is forced before the specified paragraphs. Read/write.

**paragraph format**

Returns or sets a [paragraph format](#) object that represents the paragraph settings for the specified range, selection, find or replacement operation, or style. Read/write.

**paragraph left indent**

Returns or sets the left indent value (in points) for the specified paragraphs. Read/write.

**right indent**

Returns or sets the right indent (in points) for the specified paragraphs. Read/write.

**shading**

Returns a shading object that refers to the shading formatting for the specified object. Read-only.

**space after**

Returns or sets the amount of spacing (in points) after the specified paragraph or text column. Read/write.

**space before**

Returns or sets the spacing (in points) before the specified paragraphs. Read/write.

**style**

Returns or sets the style for the specified object. To set this property, specify either the local name of the style, a constant representing one of the built-in styles from the list below, or an object that represents the style. Read/write.
When you return the style for a range that includes more than one style, only the first character or paragraph style is returned.

Can be one of the following:

<table>
<thead>
<tr>
<th>Style Block Quotation</th>
<th>Style List</th>
</tr>
</thead>
<tbody>
<tr>
<td>Style Body Text</td>
<td>Style List2</td>
</tr>
<tr>
<td>Style Body Text 2</td>
<td>Style List3</td>
</tr>
<tr>
<td>Style Body Text3</td>
<td>Style List4</td>
</tr>
<tr>
<td>Style Body Text First Indent</td>
<td>Style List5</td>
</tr>
<tr>
<td>Style Body Text First Indent2</td>
<td>Style List Bullet</td>
</tr>
<tr>
<td>Style Body Text Indent</td>
<td>Style List Bullet2</td>
</tr>
<tr>
<td>Style Body Text Indent2</td>
<td>Style List Bullet3</td>
</tr>
<tr>
<td>Style Body Text Indent3</td>
<td>Style List Bullet4</td>
</tr>
<tr>
<td>Style Caption</td>
<td>Style List Bullet5</td>
</tr>
<tr>
<td>Style Closing</td>
<td>Style List Continue</td>
</tr>
<tr>
<td>Style Comment Reference</td>
<td>Style List Continue2</td>
</tr>
<tr>
<td>Style Comment Text</td>
<td>Style List Continue3</td>
</tr>
<tr>
<td>Style Date</td>
<td>Style List Continue4</td>
</tr>
<tr>
<td>Style Default Paragraph Font</td>
<td>Style List Continue5</td>
</tr>
<tr>
<td>Style Emphasis</td>
<td>Style List Number</td>
</tr>
<tr>
<td>Style Endnote Reference</td>
<td>Style List Number2</td>
</tr>
<tr>
<td>Style Endnote Text</td>
<td>Style List Number3</td>
</tr>
<tr>
<td>Style Envelope Address</td>
<td>Style List Number4</td>
</tr>
<tr>
<td>Style Envelope Return</td>
<td>Style List Number5</td>
</tr>
<tr>
<td>Style Footer</td>
<td>Style Macro Text</td>
</tr>
<tr>
<td>Style Footnote Reference</td>
<td>Style Message Header</td>
</tr>
<tr>
<td>Style Footnote Text</td>
<td>Style Nav Pane</td>
</tr>
<tr>
<td>Style Header</td>
<td>Style Normal</td>
</tr>
<tr>
<td>Style Heading1</td>
<td>Style Normal Indent</td>
</tr>
<tr>
<td>Style Heading2</td>
<td>Style Note Heading</td>
</tr>
<tr>
<td>Style Heading3</td>
<td>Style Page Number</td>
</tr>
<tr>
<td>Style Heading4</td>
<td>Style Plain Text</td>
</tr>
<tr>
<td>Style Heading5</td>
<td>Style Salutation</td>
</tr>
<tr>
<td>Style Heading6</td>
<td>Style Signature</td>
</tr>
<tr>
<td>Style Heading7</td>
<td>Style Strong</td>
</tr>
<tr>
<td>Style Heading8</td>
<td>Style Subtitle</td>
</tr>
<tr>
<td>Style Heading9</td>
<td>Style Table of Authorities</td>
</tr>
<tr>
<td>Style Hyperlink</td>
<td>Style Table of Figures</td>
</tr>
<tr>
<td>Style Hyperlink Followed</td>
<td>Style Title</td>
</tr>
<tr>
<td>Style Index1</td>
<td>Style TOA Heading</td>
</tr>
<tr>
<td>Style Index2</td>
<td>Style TOC1</td>
</tr>
<tr>
<td>Style Index3</td>
<td>Style TOC2</td>
</tr>
<tr>
<td>Style Index4</td>
<td>Style TOC3</td>
</tr>
<tr>
<td>Style Index5</td>
<td>Style TOC4</td>
</tr>
<tr>
<td>Style Index6</td>
<td>Style TOC5</td>
</tr>
<tr>
<td>Style Index7</td>
<td>Style TOC6</td>
</tr>
<tr>
<td>Style Index8</td>
<td>Style TOC7</td>
</tr>
<tr>
<td>Style Index9</td>
<td>Style TOC8</td>
</tr>
<tr>
<td>Style Index Heading</td>
<td>Style TOC9</td>
</tr>
</tbody>
</table>
Text Suite

**text object**

Returns a text range object that represents the portion of a document that's contained in the paragraph object. Read-only.

**widow control**

True if the first and last lines in the specified paragraph remain on the same page as the rest of the paragraph when Word repaginates the document. Read/write.

**word wrap**

True if Word wraps Latin text in the middle of a word in the specified paragraphs. Read/write. This usage may not be available to you, depending on the language support (U.S. English, for example) that you've selected or installed.

**Class: paragraph format**

**Plural**

paragraph formats

**Elements**

tab stop

Represents all the formatting for a paragraph.

To return or set the paragraph format object for a paragraph or paragraphs, use the paragraph format property of the paragraph class. The paragraph format property also returns the paragraph format object for a selection object, text range, Word style, find, or replacement object. The following example centers the third paragraph in the active document.

```
set alignment of paragraph format of paragraph 3 of active document to align paragraph center
```

The following example finds the next double-spaced paragraph after the selection.

```
set theFind to find object of selection
clear formatting theFind
set line spacing rule of paragraph format of theFind to line space double
set content of theFind to ""
set forward of theFind to true
set wrap of theFind to find continue
execute find theFind
```

**Properties**

*add space between east asian and alpha*

True if Word is set to automatically add spaces between Japanese and Latin text for the specified paragraphs. Read/write.

*add space between east asian and digit*

True if Word is set to automatically add spaces between Japanese text and numbers for the specified paragraphs. Read/write.
Text Suite

alignment

Returns or sets the alignment for the specified paragraphs. Read/write.

Can be one of the following:

- align paragraph left
- align paragraph center
- align paragraph right
- align paragraph distribute
- align paragraph justify

auto adjust right indent

True if Word is set to automatically adjust the right indent for the specified paragraphs if you've specified a set number of characters per line. Read/write.

base line alignment

Returns or sets the baseline font alignment. Read/write.

Can be one of the following:

- baseline align auto
- baseline align baseline
- baseline align center
- baseline align east asian50
- baseline align top

border options

Returns a border options object that represents all the borders for the specified object. Read-only.

character unit first line indent

Returns or sets the value (in characters) for a first-line or hanging indent. Use a positive value to set a first-line indent, and use a negative value to set a hanging indent. Read/write.

character unit left indent

Returns or sets the left indent value (in characters) for the specified paragraphs. Read/write.

character unit right indent

Returns or sets the right indent value (in characters) for the specified paragraphs. Read/write.

disable line height grid

True if Word aligns characters in the specified paragraphs to the line grid when a set number of lines per page is specified. Read/write.
first line indent

Returns or sets the value (in points) for a first line or hanging indent. Use a positive value to set a first-line indent, and use a negative value to set a hanging indent. Read/write.

half width punctuation on top of line

True if Word changes punctuation symbols at the beginning of a line to half-width characters for the specified paragraphs. Read/write.

hanging punctuation

True if hanging punctuation is enabled for the specified paragraphs. Read/write.

hyphenation

True if the specified paragraphs are included in automatic hyphenation. False if the specified paragraphs are to be excluded from automatic hyphenation. Read/write.

keep together

True if all lines in the specified paragraphs remain on the same page when Word repaginates the document. Read/write.

keep with next

True if the specified paragraph remains on the same page as the paragraph that follows it when Word repaginates the document. Read/write.

line spacing

Returns or sets the line spacing (in points) for the specified paragraphs. Read/write.

The line spacing property can be set after the line spacing rule property has been set to line space at least, line space exactly, or line space multiple. If line space at least is used, the line spacing can be greater than or equal to, but never less than, the specified line spacing value. If line space exactly is used, the line spacing never changes from the specified line spacing value, even if a larger font is used within the paragraph. If line spacing multiple is used, a line spacing property value must be specified, in points.

To convert a number of lines to the corresponding value in points, use the lines to points command. For example, lines to points lines 2 returns the value 24.

line spacing rule

Returns or sets the line spacing for the specified paragraphs. Read/write.

Can be one of the following:

- line space1 pt5
- line space at least
- line space double
- line space exactly
- line space multiple
- line space single
Text Suite

Use **line space single**, **line space1 pt5**, or **line space double** to set the line spacing to one of these values. To set the line spacing to an exact number of points or to a multiple number of lines, you must also set the **line spacing** property.

**line unit after**

Returns or sets the amount of spacing (in gridlines) after the specified paragraphs. Read/write.

**line unit before**

Returns or sets the amount of spacing (in gridlines) before the specified paragraphs. Read/write.

**no line number**

**True** if line numbers are repressed for the specified paragraphs. Read/write.

Use the **line numbering** property to set line numbers.

**outline level**

Returns or sets the outline level for the specified paragraphs. Read/write.

Can be one of the following:

- **outline level1**
- **outline level2**
- **outline level3**
- **outline level4**
- **outline level5**
- **outline level6**
- **outline level7**
- **outline level8**
- **outline level9**
- **outline level body text**

If a paragraph has a heading style applied to it (Heading 1 through Heading 9), the outline level is the same as the heading style and cannot be changed.

Outline levels are visible only in outline view or in the document map pane.

**page break before**

**True** if a page break is forced before the specified paragraphs. Read/write.

**paragraph format left indent**

Returns or sets the left indent value (in points) for the specified paragraphs. Read/write.

**paragraph format right indent**

Returns or sets the right indent (in points) for the specified paragraphs. Read/write.

**shading**

Returns a **shading** object that refers to the shading formatting for the specified object. Read-only.

**space after**

Returns or sets the amount of spacing (in points) after the specified paragraph or text column. Read/write.
Text Suite

space after auto

True if Word automatically sets the amount of spacing after the specified paragraphs. Read/write.

If space after auto is set to True, the space after property is ignored.

space before

Returns or sets the spacing (in points) before the specified paragraphs. Read/write.

space before auto

True if Word automatically sets the amount of spacing before the specified paragraphs. Can be set to either True or False. Read/write.

When you open an HTML document without cascading style sheets, Word automatically sets the space before auto property to True to render the paragraph spacing exactly as it would appear in a Web browser.

If space before auto is set to True, the space before property is ignored.

style

Returns or sets the style for the specified object. To set this property, specify either the local name of the style, a constant representing one of the built-in styles from the list below, or an object that represents the style. Read/write.

When you return the style for a range that includes more than one style, only the first character or paragraph style is returned.

Can be one of the following:

- style block quotation
- style body text
- style body text 2
- style body text3
- style body text first indent
- style body text first indent2
- style body text indent
- style body text indent2
- style body text indent3
- style caption
- style closing
- style comment reference
- style comment text
- style date
- style default paragraph font
- style emphasis
- style endnote reference
- style endnote text
- style envelope address
- style envelope return
- style footer
- style footnote reference
- style footnote text
- style header
- style heading1
- style list
- style list2
- style list3
- style list4
- style list5
- style list bullet
- style list bullet2
- style list bullet3
- style list bullet4
- style list bullet5
- style list continue2
- style list continue3
- style list continue4
- style list continue5
- style list number
- style list number2
- style list number3
- style list number4
- style list number5
- style macro text
- style message header
- style nav pane
- style normal
- style normal indent
widow control

True if the first and last lines in the specified paragraph remain on the same page as the rest of the paragraph when Word repaginates the document. Read/write.

word wrap

True if Word wraps Latin text in the middle of a word in the specified paragraphs. Read/write. This usage may not be available to you, depending on the language support (U.S. English, for example) that you’ve selected or installed.

Class: section

Plural

sections

Represents a single section in a selection, range, or document. The sections list includes all the sections in a selection, range, or document.

Use section index, where index is the index number, to return a single section object. The following example changes the left and right page margins for the first section in the active document.

set left margin of page setup of section 1 of active document to ¬

    (inches to points inches 0.5)

set right margin of page setup of section 1 of active document to ¬

    (inches to points inches 0.5)

To add a new section to a document, use the insert break command. The following example adds a section break above the first paragraph in the selection.

insert break at text object of selection break type section break continuous
Properties

border options

Returns a border options object that represents all the borders for the specified object. Read-only.

page setup

Returns a page setup object that's associated with the specified section. Read-only.

protected for forms

True if the specified section is protected for forms. When a section is protected for forms, you can select and modify text only in form fields. Read/write.

Note To protect an entire document, use the protect command.

section index

Returns a number that indicates the position of an item in a list. Read-only.

text object

Returns a text range object that represents the portion of a document that's contained in the section object. Read-only.

Class: sentence

Plural

sentences

A list of text range objects that represent all the sentences in a selection, text range, or document. The following example displays the number of sentences selected.

display dialog ((count of sentences of selection) as string) & " sentences - are selected"

Use sentence index, where index is the index number, to return a text range object that represents a sentence. The index number represents the position of a sentence in the sentences list. The following example formats the first sentence in the active document.

set bold of sentence 1 of active document to true

set font size of font object of sentence 1 of active document to 24

Remarks

The count command for this list in a document returns the number of items in the main story only. To count items in other stories, use the get story range command to return the text range of the story you want to count the sentences in.
To add a sentence to a text range object, use the insert command. The following example inserts a sentence after the first paragraph in the active document.

\[
\text{display dialog (}} \left\langle \text{count of sentences of text object of active document} \right\rangle \text{ as string) \& " sentences"
\]

insert text "\r" at text object of paragraph 1 of active document
insert text "The house is blue. " at text object of paragraph 2 of active document

display dialog (\left\langle \text{count of sentences of text object of active document} \right\rangle \text{ as string) \& " sentences"

**Properties**

\textbf{<Inheritance>} text range

Inherits the properties and elements of the text range class.

**Class: shading**

**Plural**

shadings

Contains shading attributes for an object.

Use the shading property to return the shading object. The following example applies fine gray shading to the first paragraph in the active document.

set texture of shading of paragraph 1 of active document to texture10 percent

The following example applies shading with different foreground and background colors to the selection.

set texture of shading of selection to texture20 percent
set foreground pattern color index of shading of selection to blue
set background pattern color index of shading of selection to yellow

The following example applies a vertical line texture to the first row in the first table in the active document.

set texture of shading of row 1 of table 1 of active document to texture vertical
Text Suite

**Properties**

**background pattern color**

Returns or sets the RGB color that's applied to the background of the shading object. Read/write.

**background pattern color index**

Returns or sets the color that's applied to the background of the shading object. Read/write. Can be one of the following:

- auto
- black
- blue
- bright green
- dark blue
- dark red
- dark yellow
- gray25
- gray50
- green
- no highlight
- pink
- red
- teal
- turquoise
- violet
- white
- yellow

**foreground pattern color**

Returns or sets the RGB color that's applied to the foreground of the shading object. This color is applied to the dots and lines in the shading pattern. Read/write.

**foreground pattern color index**

Returns or sets the color that's applied to the foreground of the shading object. This color is applied to the dots and lines in the shading pattern. Read/write. Can be one of the following:

- auto
- black
- blue
- bright green
- by author
- dark blue
- red
- dark yellow
- gray25
- gray50
- green
- no highlight
- pink
- red
- teal
- turquoise
- violet
- white
- yellow
Returns or sets the shading texture for the specified object. Read/write.

Can be one of the following:

- `texture auto`
- `texture solid`
- `texture2 pt5 percent`
- `texture5 percent`
- `texture7 pt5 percent`
- `texture10 percent`
- `texture12 pt5 percent`
- `texture15 percent`
- `texture17 pt5 percent`
- `texture20 percent`
- `texture22 pt5 percent`
- `texture25 percent`
- `texture27 pt5 percent`
- `texture30 percent`
- `texture32 pt5 percent`
- `texture35 percent`
- `texture37 pt5 percent`
- `texture40 percent`
- `texture42 pt5 percent`
- `texture45 percent`
- `texture47 pt5 percent`
- `texture50 percent`
- `texture52 pt5 percent`
- `texture55 percent`
- `texture57 pt5 percent`
- `texture60 percent`
- `texture62 pt5 percent`
- `texture65 percent`

Class: spelling error

Plural

spelling errors

A list of spelling errors for the specified document or range. Each item in the spelling errors list is a text range object that represents one spelling or grammatical error.

The following example counts the spelling and grammatical errors in the selection and displays the results in a message box.

```plaintext
set sp1 to spelling errors of text object of selection
set sc to (count of sp1) as string
set gr1 to grammatical errors of text object of selection
set gc to (count of gr1) as string
display dialog "Spelling errors: " & sc & return & "Grammatical errors:" & gc
```
Text Suite

Use spelling error index, where index is the index number, to return a single spelling error (represented by a text range object). The following example finds the second spelling error in the selection and then selects it.

set myRange to spelling error 2 of text object of selection
select myRange

Remarks

If all the words in the document or range are spelled correctly, using the count command for the spelling errors list returns 0 (zero) and the spelling checked property of the text range returns True.

Properties

<Inheritance> text range

Inherits the properties and elements of the text range class.

Class: text range

Plural

ranges

Elements

| character | bookmark |
| word      | revision |
| sentence  | hyperlink object |
| table     | paragraph |
| footnote  | subdocument |
| endnote   | column |
| Word comment | row |
| cell      | shape |
| section   | readability statistic |
| paragraph | grammatical error |
| field     | spelling error |
| form field | inline shape |
| frame     |           |

Represents a contiguous area in a document. Each text range object is defined by a starting and ending character position. Similar to the way bookmarks are used in a document, text range objects are used to identify specific portions of a document. However, unlike a bookmark, a text range object exists only while the procedure that defined it is running.

Note text range objects are independent of the selection. That is, you can define and manipulate a range without changing the selection. You can also define multiple ranges in a document, while there can be only one selection per pane.

To return a text range object defined by the given starting and ending character positions, use the set range command. The following example returns a text range object that refers to the first 10 characters in the active document.

set myRange to set range text object of active document start 0 end 10
Use the text object property to return a text range object defined by the beginning and end of another object. The text object property applies to many objects (for example, paragraph, bookmark, and cell). The following example returns a text range object that refers to the first paragraph in the active document.

```
set aRange to set range text object of paragraph 1 of active document
```

The following example returns a text range object that refers to the second through fourth paragraphs in the active document.

```
set aRange to set range text object of active document start ¬
           (start of content of text object of paragraph 2 of active document) ¬
           end (end of content of text object of paragraph 4 of active document)
```

For more information about working with text range objects, see working with text range objects.

### Properties

- **bold**
  - True if the font or range is formatted as bold. Read/write.

- **bookmark id**
  - Returns the number of the bookmark that encloses the beginning of the specified selection or range; returns 0 (zero) if there’s no corresponding bookmark. The number corresponds to the position of the bookmark in the document: 1 for the first bookmark, 2 for the second one, and so on. Read-only.

- **border options**
  - Returns a border options object that represents all the borders for the specified object. Read-only.

- **case**
  - Returns or sets the case of the text in the specified range. Read/write.

  Can be one of the following:

  - lower case
  - next case
  - title sentence
  - title word
  - toggle case
  - upper case

- **column options**
  - Returns the column options object associated with the table object. Read-only.

- **content**
  - Returns or sets the text in the specified range or selection. Read/write.
Text Suite

disable character space grid

True if Word ignores the setting for the number of characters per line in the Document Grid. Returns True if the disable character space grid property is set to True for only some of the specified paragraphs. Read/write.

emphasis mark

Returns or sets the emphasis mark for a character or designated character string. Read/write. Can be one of the following:

- emphasis mark none
- emphasis mark comma
- emphasis mark over solid circle
- emphasis mark over white circle
- emphasis mark under solid circle

end of content

Returns or sets the ending character position of a selection, range, or bookmark. Read/write.

Note If this property is set to a value smaller than the start of content property, the start of content property is set to the same value (that is, the start of content and end of content property are equal).

The selection, range, and bookmark objects all have a starting position and an ending position. The ending position is the point farthest away from the beginning of the story.

This property returns the ending character position relative to the beginning of the story. The main document story (main text story) begins with character position 0 (zero). You can change the size of a selection, range, or bookmark by setting this property.

endnote options

Returns an endnote options object that represents all the endnote options in a text range. Read-only.

find object

Returns a find object that contains the criteria for a find operation. Read-only.

Note When this property is used with a text range object, the selection isn't changed unless the select command is applied.

fit text width

Returns or sets the width (in the current measurement units) in which Word fits the text in the current selection or text range. Read/write.

font object

Returns a font object that represents the character formatting of the specified object. Read only.
footnote options

Returns a **footnote options** object that represents all the footnote options in a text range. Read-only.

formatted text

Returns or sets a **text range** object that includes the formatted text in the specified text range or selection. Read/write.

This property returns a **text range** object with the character formatting and text from the specified text range or selection. Paragraph formatting is included in the **text range** object if there's a paragraph mark in the text range or selection.

When you set this property, the text in the range is replaced with formatted text. If you don’t want to replace the existing text, use the **collapse range** command before using this property.

grammar checked

**True** if a grammar check has been run on the specified text range or document. **False** if some of the specified text range or document hasn’t been checked for grammar. Read/write.

To recheck the grammar in a range or document, set the **grammar checked** property to **False**.

highlight color index

Returns or sets the highlight color for the specified range. Read/write.

Can be one of the following:

- no highlight
- black, blue
- bright green
- dark blue
- dark red
- dark yellow
- gray35
- gray50
- green
- pink
- red
- teal
- turquoise
- violet
- white
- yellow

**Note** Setting this property to **no highlight** removes the highlight color (if any) from the specified range.

is end of row mark

**True** if the specified selection or range is collapsed and is located at the end-of-row mark in a table. Read-only.

italic

**True** if the font or range is formatted as italic. Read/write.
text suite

language id

Returns or sets the language for the specified object. Read/write.

Can be one of the following:

- afrikaans
- arabic
- basque
- belgian dutch
- belgian french
- brazilian portuguese
- bulgarian
- byelorussian
- catalan
- croatian
- czech
- danish
- dutch
- english aus
- english canadian
- english new zealand
- english south africa
- english uk
- english us
- estonian
- farsi
- finnish
- french
- french canadian
- german
- greek
- hebrew
- hungarian
- italian
- icelandic
- japanese
- korean

language id east asian

Returns or sets an East Asian language for the specified object. Can be any of the language id constants listed above. Read/write.

Note  This is the recommended way to apply an East Asian language to text in a document created in an East Asian version of Word.

list format

Returns a list format object that represents all the list formatting characteristics of a text range. Read-only.
next story range

Returns a text range object that refers to the next story, as shown in the following table. Read-only.

<table>
<thead>
<tr>
<th>Story type</th>
<th>Item returned</th>
</tr>
</thead>
<tbody>
<tr>
<td>main text story, footnotes story, endnotes story, and comments story</td>
<td>Always returns Nothing</td>
</tr>
<tr>
<td>text frame story</td>
<td>The story of the next set of linked text boxes</td>
</tr>
<tr>
<td>even pages header story, primary header story, even pages footer story,</td>
<td>The next section's story of the same type</td>
</tr>
<tr>
<td>primary footer story, first page header story, first page footer story</td>
<td></td>
</tr>
</tbody>
</table>

no proofing

True if the spelling and grammar checker ignores the specified text. Read/write.

orientation

Returns or sets the orientation of the text in the range or selection when the Text Direction feature is enabled. Read/write.

Can be one of the following:

- text orientation downward
- text orientation horizontal
- text orientation upward
- text orientation vertical east asian
- text orientation horizontal rotated east asian

page setup

Returns a page setup object that's associated with the specified document, text range, section, sections, or selection. Read-only.

paragraph format

Returns or sets a paragraph format object that represents the paragraph settings for the specified range, selection, find or replacement operation, or style. Read/write.

previous bookmark id

Returns the number of the last bookmark that starts before or at the same place as the specified selection or range; returns 0 (zero) if there's no corresponding bookmark. Read-only.

row options

Returns the row options object associated with the table object. Read-only.

shading

Returns a shading object that refers to the shading formatting for the specified object. Read-only.
show Word comments by
Returns or sets the name of the reviewer whose comments are shown in the reviewing pane.
You can choose to show comments either by a single reviewer or by all reviewers. To view the comments of all reviewers, set this property to all reviewers.

show hidden bookmarks
True if hidden bookmarks are included in the bookmarks list. This property also controls whether hidden bookmarks are listed in the Bookmark dialog box (Insert menu). Read/write.
Hidden bookmarks are automatically inserted when cross-references are inserted into the document.

spelling checked
True if spelling has been checked throughout the specified range or document. False if all or some of the range or document hasn't been checked for spelling. Read/write.
To recheck the spelling in a range or document, set the spelling checked property to False.
To see whether the range or document contains spelling errors, use the spelling errors list.

start of content
Returns or sets the starting character position of a selection, text range, or bookmark. Read/write.

Note If this property is set to a value larger than that of the end of content property, the end of content property is set to the same value as that of the start of content property.

Selection, text range, and bookmark objects have starting and ending character positions. The starting position refers to the character position closest to the beginning of the story. This property returns the starting character position relative to the beginning of the story. The main text story (main text story) begins with character position 0 (zero). You can change the size of a selection, range, or bookmark by setting this property.

story length
Returns the number of characters in the story that contains the specified range or selection. Read-only.

story type
Returns the story type for the specified range, selection, or bookmark. Read-only.
Can be one of the following:

- comments story
- endnotes story
- even pages footer story
- even pages header story
- first page footer story
- first page header story
- footnotes story
- main text story
- primary footer story
- primary header story
- text frame story

style
Returns or sets the style for the specified object. To set this property, specify either the local name of the style, a constant representing one of the built-in styles from the list below, or an object that represents the style. Read/write.
When you return the style for a range that includes more than one style, only the first character or paragraph style is returned.

Can be one of the following:

- style block quotation
- style body text
- style body text 2
- style body text3
- style body text first indent
- style body text first indent2
- style body text indent
- style body text indent2
- style body text indent3
- style caption
- style closing
- style comment reference
- style comment text
- style date
- style default paragraph font
- style emphasis
- style endnote reference
- style endnote text
- style envelope address
- style envelope return
- style footer
- style footnote reference
- style footnote text
- style header
- style normal
- style heading1
- style heading2
- style heading3
- style heading4
- style heading5
- style heading6
- style heading7
- style heading8
- style heading9
- style hyperlink
- style hyperlink followed
- style index1
- style index2
- style index3
- style index4
- style index5
- style index6
- style index7
- style index8
- style index9
- style index heading
- style line number
- style list
- style list2
- style list3
- style list4
- style list5
- style list bullet
- style list bullet2
- style list bullet3
- style list bullet4
- style list bullet5
- style list continue
- style list continue2
- style list continue3
- style list continue4
- style list continue5
- style list number
- style list number2
- style list number3
- style list number4
- style list number5
- style macro text
- style nav pane
- style normal indent
- style note heading
- style page number
- style plain text
- style salutation
- style signature
- style strong
- style subtitle
- style table of authorities
- style table of figures
- style TOA heading
- style TOC1
- style TOC2
- style TOC3
- style TOC4
- style TOC5
- style TOC6
- style TOC7
- style TOC8
- style TOC9
Text Suite

*subdocuments expanded*

**True** if the subdocuments in the specified document are expanded. Read/write.

*supplemental language id*

Returns or sets the language for the specified object. Can be any of the *language id* constants listed above. Read/write.

**Note** This is the recommended way to determine the language of a document created in an East Asian version of Word.

*text retrieval mode*

Returns a *text retrieval mode* object that controls how text is retrieved from the specified text range. Read/write.

*underline*

Returns or sets the type of underline applied to the font or range. Read/write.

Can be one of the following:

- **underline none**
- **underline dash**
- **underline dot dash**
- **underline dot dot dash**
- **underline thick**
- **underline dotted**
- **underline double**
- **underline single**
- **underline words**
- **underline wavy**

**Class: word**

**Plural**

*words*

A list of words in a selection, range, or document. Each item in the *words* list is a *text range* object that represents one word.

The following example displays how many words are currently selected.

`display dialog ((count of (words of text object of selection)) as string) & ¬
   " words are selected"`

To return a *text range* object that represents one word, use *word index*, where *index* is the index number. The index number represents the position of the word in the *words* list. The following example formats the first word in the selection as 24-point italic.

`set italic of word 1 of text object of selection to true
set font size of font object of word 1 of text object of selection to 24`

The item in the *words* list includes both the word and the spaces after the word. The following example selects the first word (and its trailing spaces) in the active document.

`select word 1 of text object of active document`
Remarks
The count command with this list in a document returns the number of items in the main story only. To count items in other stories, use the list with the text range object.

The make command isn't available for the Word class. Instead, use the insert command to add text to a text range object. The following example inserts text after the first word in the active document:

insert text "New" at the end of word 1 of text object of active document

Properties
<Inheritance> text range

Inherits the properties and elements of the text range class.

Class: Word style

Plural
Word styles

Represents a single built-in or user-defined style. The Word style class includes style attributes (font, font style, paragraph spacing, and so on) as properties of the Word style class. The Word styles list includes all the styles in the specified document.

To return a single Word style object, use Word style index, where index is the style name, a built-in style, or an index number. You must exactly match the spelling and spacing of the style name, but not necessarily its capitalization. The following example modifies the font name of the user-defined style named "Color" in the active document.

set name of font object of Word style "Color" of active document to "Arial"

The following example sets the built-in Heading 1 style to not be bold.

set bold of font object of Word style style heading1 of active document to false

The style index number represents the position of the style in the alphabetically sorted list of style names. Note that Word styles 1 is the first style in the alphabetic list. The following example displays the base style and style name of the first style in the Word styles list.

display dialog "Base style: " & base style of Word style 1 of active document & return & "Style name: " & name local of Word style 1 of active document

To apply a style to a range, paragraph, or multiple paragraphs, set the style property to a user-defined or built-in style name. The following example applies the Normal style to the first four paragraphs in the active document.

set myRange to set range (text object of active document) start (start of content of text object of paragraph 1 of active document) end (end of content of text object of paragraph 4 of active document)

set style of myRange to style normal
The following example applies the Heading 1 style to the first paragraph in the selection.

```plaintext
set style of text object of paragraph 1 of selection to style heading1
```

The following example creates a character style named "Bolded" and applies it to the selection.

```plaintext
set myStyle to make new Word style at active document with properties ~
   {name local:"Bolded", style type:style type character}
set bold of font object of myStyle to true
set style of text object of selection to "Bolded"
```

### Remarks

To copy styles between documents and templates, use the organizer copy command. To update the styles in the active document to match the style definitions in the attached template, use the update styles command. To open a template as a document so that you can modify the template styles, use the open as document command.

### Properties

**automatically update**

- **True** if the style is automatically redefined based on the selection. **False** if Word prompts for confirmation before redefining the style based on the selection. A style can be redefined when it's applied to a selection that has the same style but different manual formatting. Read/write.

**base style**

Returns or sets an existing style on which you can base the formatting of another style. To set this property, specify either the local name of the base style, a built-in style, or an object that represents the base style. Read/write.

**border options**

Returns a border options object that represents all the borders for the specified object. Read-only.

**built in**

- **True** if the specified object is one of the built-in styles or caption labels in Word. Read-only.

  You can specify built-in styles across all languages by using the built-in style name within a language or by using the style name for the language version of Word.

**description**

Returns the description of the specified style. For example, a typical description for the Heading 2 style might be "Normal + Font: Arial, 12 pt, Bold, Italic, Space Before 12 pt After 3 pt, KeepWithNext, Level 2." Read-only.

**font object**

Returns or sets a font object that represents the character formatting of the specified object. To set this property, specify an expression that returns a font object. Read/write.

**frame**

Returns a frame object that represents the frame formatting for the specified style or find-and-replace operation. Read-only.
**in use**

*True* if the specified style is a built-in style that has been modified or applied in the document or a new style that has been created in the document. Read-only.

**Remarks**

This property doesn't indicate whether the style is currently applied to any text in the document. For instance, if text that’s been formatted with a style is deleted, the *in use* property of the style remains *True*. For built-in styles that have never been used in the document, this property returns *false*.

**language id**

Returns or sets the language for the specified object. Read/write.

Can be one of the following:

- **afrikaans**
- **arabic**
- **basque**
- **belgian dutch**
- **belgian french**
- **brazilian portuguese**
- **bulgarian**
- **byelorussian**
- **catalan**
- **croatian**
- **czech**
- **danish**
- **dutch**
- **english aus**
- **english canadian**
- **english new zealand**
- **english south africa**
- **english uk**
- **english us**
- **estonian**
- **farsi**
- **finnish**
- **french**
- **french canadian**
- **german**
- **greek**
- **hebrew**
- **hungarian**
- **italian**
- **icelandic**
- **japanese**
- **korean**
- **language none**
- **latvian**
- **macedonian**
- **malaysian**
- **mexicanSpanish**
- **language no proofing**
- **norwegian bokmol**
- **norwegian nynorsk**
- **polish**
- **portuguese**
- **romanian**
- **russian**
- **serbian cyrillic**
- **serbian latin**
- **sesotho**
- **simplified chinese**
- **slovak**
- **slovenian**
- **spanish**
- **spanish modern sort**
- **swedish**
- **swiss french**
- **swiss german**
- **swiss italian**
- **traditional chinese**
- **tsonga**
- **tswana**
- **turkish**
- **ukrainian**
- **venda**
- **xhosa**
- **zulu**
Text Suite

language id east asian

Returns or sets an East Asian language for the specified object. Can be any of the language id constants listed above. Read/write.

Note This is the recommended way to apply an East Asian language to text in a document created in an East Asian version of Word.

list level number

Returns the list level for the specified style. Read-only.

list template

Returns a list template object that represents the list formatting for the specified style or list format object. Read-only.

A list template includes all the formatting that defines a particular list. Each of the seven formats (excluding None) found on each of the tabs in the Bullets and Numbering dialog box corresponds to a list template. Documents and templates can also contain collections of list templates.

If the first paragraph in the range for the list format object is not formatted as a list, the list format property returns Nothing.

name local

Returns the name of a built-in style in the language of the user. Setting this property renames a user-defined style or adds an alias to a built-in style. Read/write.

next paragraph style

Returns or sets the style to be applied automatically to a new paragraph inserted after a paragraph formatted with the specified style. To set this property, specify either the local name of the next style, a built-in style, or an object that represents the next style. Read/write.

no proofing

True if the spelling and grammar checker ignores text formatted with this style. Read/write.

paragraph format

Returns or sets a paragraph format object that represents the paragraph settings for the specified range, selection, find or replacement operation, or style. Read/write.

shading

Returns a shading object that refers to the shading formatting for the specified object. Read-only.

style type

Returns the style type. Read-only.

Can be one of the following:

- style type paragraph
- style type table
- style type list
- style type character
## Commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>auto format text range</td>
<td>444</td>
</tr>
<tr>
<td>calculate range</td>
<td>444</td>
</tr>
<tr>
<td>change end of range</td>
<td>445</td>
</tr>
<tr>
<td>change start of range</td>
<td>446</td>
</tr>
<tr>
<td>check grammar</td>
<td>446</td>
</tr>
<tr>
<td>check spelling</td>
<td>447</td>
</tr>
<tr>
<td>check synonyms</td>
<td>447</td>
</tr>
<tr>
<td>close up</td>
<td>448</td>
</tr>
<tr>
<td>collapse range</td>
<td>448</td>
</tr>
<tr>
<td>compute statistics</td>
<td>449</td>
</tr>
<tr>
<td>convert to table</td>
<td>449</td>
</tr>
<tr>
<td>copy as picture</td>
<td>451</td>
</tr>
<tr>
<td>copy object</td>
<td>451</td>
</tr>
<tr>
<td>cut object</td>
<td>451</td>
</tr>
<tr>
<td>expand</td>
<td>452</td>
</tr>
<tr>
<td>get border</td>
<td>452</td>
</tr>
<tr>
<td>get footer</td>
<td>453</td>
</tr>
<tr>
<td>get header</td>
<td>454</td>
</tr>
<tr>
<td>get range information</td>
<td>454</td>
</tr>
<tr>
<td>go to next</td>
<td>458</td>
</tr>
<tr>
<td>go to previous</td>
<td>458</td>
</tr>
<tr>
<td>in range</td>
<td>459</td>
</tr>
<tr>
<td>in story</td>
<td>459</td>
</tr>
<tr>
<td>indent</td>
<td>460</td>
</tr>
<tr>
<td>indent char width</td>
<td>460</td>
</tr>
<tr>
<td>indent first line char width</td>
<td>461</td>
</tr>
<tr>
<td>is equivalent</td>
<td>461</td>
</tr>
<tr>
<td>link to list template</td>
<td>462</td>
</tr>
<tr>
<td>merge subdocuments</td>
<td>462</td>
</tr>
<tr>
<td>modify enclosure</td>
<td>463</td>
</tr>
<tr>
<td>move end of range</td>
<td>463</td>
</tr>
<tr>
<td>move range</td>
<td>464</td>
</tr>
<tr>
<td>move range end until</td>
<td>465</td>
</tr>
<tr>
<td>move range end while</td>
<td>465</td>
</tr>
<tr>
<td>move range start until</td>
<td>466</td>
</tr>
<tr>
<td>move range start while</td>
<td>466</td>
</tr>
<tr>
<td>move range until</td>
<td>467</td>
</tr>
<tr>
<td>move range while</td>
<td>467</td>
</tr>
<tr>
<td>move start of range</td>
<td>468</td>
</tr>
<tr>
<td>navigate</td>
<td>468</td>
</tr>
<tr>
<td>next paragraph</td>
<td>469</td>
</tr>
<tr>
<td>next range</td>
<td>470</td>
</tr>
<tr>
<td>next subdocument</td>
<td>470</td>
</tr>
<tr>
<td>open or close up</td>
<td>470</td>
</tr>
<tr>
<td>open up</td>
<td>471</td>
</tr>
<tr>
<td>outline</td>
<td>471</td>
</tr>
<tr>
<td>outline demote to body</td>
<td>472</td>
</tr>
<tr>
<td>outline promote</td>
<td>472</td>
</tr>
<tr>
<td>paste and format</td>
<td>473</td>
</tr>
<tr>
<td>paste append table</td>
<td>474</td>
</tr>
<tr>
<td>paste as nested table</td>
<td>474</td>
</tr>
<tr>
<td>paste excel table</td>
<td>474</td>
</tr>
<tr>
<td>paste object</td>
<td>475</td>
</tr>
<tr>
<td>paste special</td>
<td>475</td>
</tr>
</tbody>
</table>
Command: auto format text range

Automatically formats a text range. The arguments for this command correspond to the options in the Table AutoFormat dialog box (Table menu).

Syntax

auto format text range  text range  Required. An expression that returns a text range object.

Example

This example automatically formats the selection.

auto format text range text object of selection

Command: calculate range

Calculates a mathematical expression within a range or selection. Returns the result as a small real number.

Syntax

calculate range  text range  Required. An expression that returns a text range.

Example

This example inserts a mathematical expression at the beginning of the active document, calculates the expression, and then appends the results to the range. The result is "1 + 1 = 2".

type text selection text "1 + 1"
set myRange to set range text object of active document start (start of content of text object of active document) end (end of content of text object of active document)

// calculate the result

type text selection text "= " & (calculate range myRange)

This example calculates the selected mathematical expression and displays the result.

display dialog "The answer is... " & calculate range text object of selection
Command: change end of range

Moves or extends the ending character position of a text range to the end of the nearest specified text unit. This command returns a new text range.

Syntax

change end of range  text range  Required. An expression that returns a text range object.

[by  enumeration]  Optional. The unit by which to move the ending character position. Can be one of the following: a character item, a word item, a sentence item, a paragraph item, a line item, a story item, a screen, a section, a column, a row, a window, a cell, a character formatting, a paragraph formatting, a table, or a item unit. The default value is a word item.

[extend type  enumeration]  Optional. Can be either of the following: by moving or by selecting. If by moving is used, both ends of the text range are moved to the end of the specified unit. If by selecting is used, the end of the text range is extended to the end of the specified unit. The default value is by moving.

Remarks

If both the starting and ending positions for the text range are already at the end of the specified unit, this command doesn't move or extend the text range.

Example

This example extends the selection to the end of the paragraph.

change end of range text object of selection by a paragraph item extend type ¬
    by selecting

This example moves myRange to the end of the first word in the selection (after the trailing space).

set myRange to character 1 of text object of selection
change end of range myRange by a word item extend type by moving

This example adds a table, selects the first cell in row two, and then extends the selection to the end of the column.

set myRange to set range text object of active document start 0 end 0
set myTable to make new table at active document with properties ¬
    {text object:myRange, number of rows:5, number of columns:3}
select (get cell from table myTable row 2 column 1)
change end of range text object of selection by a column extend type ¬
    by selecting
Command: change start of range

Moves or extends the start position of the specified text range to the beginning of the nearest specified text unit. This command returns a new text range.

Syntax

change start of range  text range  Required. An expression that returns a text range object.

[by enumeration]  Optional. The unit by which to move the starting character position. Can be one of the following: a character item, a word item, a sentence item, a paragraph item, a line item, a story item, a screen, a section, a column, a row, a window, a cell, a character formatting, a paragraph formatting, a table, or a item unit. The default value is a word item.

[extend type  enumeration]  Optional. Can be either of the following: by moving or by selecting.
If by moving is used, both ends of the text range are moved to the end of the specified unit. If by selecting is used, the start of the text range is extended to the start of the specified unit. The default value is by moving.

Remarks

If the beginning of the specified range or selection is already at the beginning of the specified unit, this command doesn't move or extend the text range.

Example

This example moves myRange to the beginning of the second sentence in the document (myRange is collapsed and positioned at the beginning of the second sentence).
set myRange to sentence 2 of active document
set myRange to change start of range myRange by a sentence item ¬
extend type by moving

Command: check grammar

Begins a grammar check for the specified text range. If the text range contains errors, this command displays the Spelling and Grammar dialog box (Tools menu), with the Check grammar option selected.

Syntax

check grammar  text range  Required. An expression that returns a text range object.

Example

This example begins a spelling and grammar check for all stories in the active document.
check grammar text object of active document
This example begins a spelling and grammar check on section two in MyDocument.doc.
set range2 to set range text object of section 2 of document "MyDocument.doc"
check grammar range2
This example begins a spelling and grammar check on the selection.
check grammar text object of selection
Command: check spelling

Begins a spelling check for the specified text range. If the text range contains errors, this command displays the Spelling and Grammar dialog box (Tools menu), with the Check grammar check box cleared.

Syntax

check spelling text range Required. An expression that returns a text range object.

[custom dictionary dictionary] Optional. Either an expression that returns a dictionary object or the file name of the custom dictionary.

[ignore uppercase Boolean] Optional. True if capitalization is ignored. If this argument is omitted, the current value of the ignore uppercase property is used.

[main dictionary dictionary] Optional. Either an expression that returns a dictionary object or the file name of the main dictionary.

[custom dictionary2–custom dictionary10 dictionary] Optional. Either an expression that returns a dictionary object or the file name of an additional custom dictionary. You can specify as many as nine additional dictionaries.

Example

This example begins a spelling check on all available stories of the active document.

check spelling text object of active document

This example begins a spelling check on section two of MyDocument.doc. The spelling check includes words in all uppercase letters, and it checks words against two custom dictionaries as well as the main dictionary.

set range2 to text object of section 2 of document "MyDocument.doc"

check spelling range2 custom dictionary dictionary "MyWork" ¬
custom dictionary2 dictionary "MyTechnical"

This example begins a spelling check on the selection.

check spelling text object of selection

This example displays the result of a spelling check on the selection.

set pass to check spelling text object of selection
display dialog "Selection has no spelling errors: " & pass

Command: check synonyms

Displays the Thesaurus pane of the Reference Tools in the Toolbox (Thesaurus command, Tools menu), which lists alternative word choices, or synonyms, for the text in the specified range.

Syntax

cHECK synonyms text range Required. An expression that returns a Range object.
Example

This example displays the Thesaurus with synonyms for the selected text.
check synonyms text object of selection

This example displays the Thesaurus dialog box with synonyms for the first word in the active document.
check synonyms word 1 of active document

Command: close up

Removes any spacing before the specified paragraphs.

Syntax

close up paragraph/paragraph format  Required. An expression that returns a paragraph or paragraph format object.

The following two statements are equivalent:
close up paragraph 1 of active document
set space before of paragraph 1 of active document to 0

Example

This example removes any space before the first paragraph in the selection.
close up paragraph 1 of selection

This example changes the Heading 1 style in the active document so that there's no space before Heading 1 paragraphs.
close up paragraph format of Word style "Heading 1" of active document

Command: collapse range

Collapses a text range to the starting or ending position and returns a new text range object. After a range or selection is collapsed, the starting and ending points are equal.

Syntax

collapse range  text range  Required. An expression that returns a text range.

[direction  enumeration]  Optional. The direction in which to collapse the range or selection. Can be either of the following: collapse end or collapse start. The default value is collapse start.

Remarks

If you use collapse end to collapse a range that refers to an entire paragraph, the range is located after the ending paragraph mark (the beginning of the next paragraph). However, you can move the range back one character by using the move end of range command after the range is collapsed, as shown in the following example.

set myRange to text object of paragraph 1 of active document
set myRange to collapse range myRange direction collapse end
set myRange to move end of range myRange by a character item count -1
Text Suite

Example
This example collapses the selection to an insertion point at the beginning of the previous selection.

`collapse range text object of selection direction collapse start`

This example sets `myRange` equal to the contents of the active document, collapses `myRange`, and then inserts a 2x2 table at the end of the document.

`set myRange to text object of active document`
`set myRange to collapse range myRange direction collapse end`
`make new table at active document with properties {number of rows:2, ~
  number of columns:2}`

Command: compute statistics

Returns a statistic based on the contents of the specified text range.

Syntax

`compute statistics`  `text range`  Required. An expression that returns a `text range` object.

`[statistic enumeration]`  Required. The statistic. Can be one of the following: `statistic words`, `statistic lines`, `statistic pages`, `statistic characters`, `statistic paragraphs`, `statistic characters with spaces`, or `statistic east asian characters`.

Command: convert to table

Converts text within a range or selection to a table. Returns the table as a `table` object.

Syntax

`convert to table`  `text range`  Required. An expression that returns a `text range` object.

`[separator enumeration]`  Optional. Specifies the character used to separate text into cells. Can be a character or one of the following: `separate by paragraphs`, `separate by tabs`, `separate by commas`, or `separate by default list separator`. If this argument is omitted, the value of the `default table separator` property is used.

`[number of rows integer]`  Optional. The number of rows in the table. If this argument is omitted, Word sets the number of rows, based on the contents of the range or selection.

`[number of columns integer]`  Optional. The number of columns in the table. If this argument is omitted, Word sets the number of columns, based on the contents of the range or selection.

`[initial column width integer]`  Optional. The initial width (in points) of each column. If this argument is omitted, Word calculates and adjusts the column width so that the table stretches from margin to margin.

`[table format enumeration]`  Optional. Specifies one of the predefined formats listed in the Table AutoFormat dialog box (Table menu). Can be any valid table style.

`[apply borders Boolean]`  Optional. `True` to apply the border properties of the specified format.
[apply shading  Boolean]  Optional. True to apply the shading properties of the specified format.

[apply font  Boolean]  Optional. True to apply the font properties of the specified format.

[apply color  Boolean]  Optional. True to apply the color properties of the specified format.

[apply heading rows  Boolean]  Optional. True to apply the heading-row properties of the specified format.

[apply last row  Boolean]  Optional. True to apply the last-row properties of the specified format.

[apply first column  Boolean]  Optional. True to apply the first-column properties of the specified format.

[apply last column  Boolean]  Optional. True to apply the last-column properties of the specified format.

[auto fit  Boolean]  Optional. True to decrease the width of the table columns as much as possible without changing the way text wraps in the cells.

[auto fit behavior  enumeration]  Optional. Sets the AutoFit rules for how Word sizes a table. Can be one of the following: auto fit fixed, auto fit content, or auto fit window. If default table behavior is word8 table behavior, this argument is ignored.

[default table behavior  enumeration]  Optional. Sets a value that specifies whether Word automatically resizes cells in a table to fit the contents (AutoFit). Can be one of the following: word8 table behavior (AutoFit disabled) or word9 table behavior (AutoFit enabled). The default value is word8 table behavior.

**Example**

This example converts the first three paragraphs in the active document to a table.

```plaintext
set aDoc to the active document
set myRange to set range text object of aDoc start (start of content of text object of paragraph 1 of aDoc) end (end of content of text object of paragraph 3 of aDoc)
convert to table myRange separator separate by paragraphs
```

This example inserts text at the insertion point and then converts the comma-delimited text to a table with formatting.

```plaintext
collapse range text object of selection
type text selection text "one, two, three"
type paragraph selection
type text selection text "one, two, three"
type paragraph selection
set myTable to convert to table (text object of active document) separator ~
  separate by commas table format table format list8
```
**Command: copy as picture**

Copies the specified text range to the Clipboard as a picture.

**Syntax**

`copy as picture` text range  Required. An expression that returns a **text range** object.

**Example**

This example copies the first paragraph in the active document to the Clipboard as a picture, and then it pastes the picture into the next document.

```plaintext
copy as picture text object of paragraph 1 of document 1
if (count of documents ≥ 2) then
  activate object document 2
  paste object text object of active document
end if
```

**Command: copy object**

Copies the specified object to the Clipboard.

**Syntax**

`copy object` text range  Required. An expression that returns a **text range** object.

**Example**

This example copies the contents of the selection into a new document.

```plaintext
if selection type of selection is selection normal then
  copy object text object of selection
  set newDoc to make new document
  paste object text object of newDoc
end if
```

This example copies the first paragraph in the active document and pastes it at the end of the document.

```plaintext
copy object text object of paragraph 1 of active document
collapse range text object of active document direction collapse end
paste object text object of selection
```

**Command: cut object**

Removes the specified object from the document and places it on the Clipboard.

**Syntax**

`cut object` text range  An expression that returns a **text range** object.

The contents of the object are cut to the Clipboard, but the collapsed object remains in the document.
Example
This example cuts the first word in the first paragraph and pastes the word at the end of the paragraph.

cut object word 1 of paragraph 1 of text object of active document
collapse range text object of paragraph 1 of active document direction ¬
collapse end
move end of range text object of paragraph 1 of active document by ¬
a character item count -1
paste object text object of selection

This example cuts the contents of the selection and pastes them into a new document.
if selection type of selection is selection normal then
    cut object text object of selection
    set newDoc to make new document
    paste object text object of newDoc
end if

Command: expand
Expands the specified range or selection. Returns a new text range.

Syntax
expand  text range  Required. An expression that returns a text range object.

[by  enumeration]  Optional. The unit by which to expand the range. Can be one of the following:
    a character item, a word item, a sentence item, a paragraph item, a line item, a story item, a
    screen, a section, a column, a row, a window, a cell, a character formatting, a paragraph
    formatting, a table, or a item unit. The default value is a word item.

Example
This example creates a range that refers to the first word in the active document, and then it
expands the range to reference the first paragraph in the document.
set myRange to word 1 of active document
set myRange to expand myRange by a paragraph item

Command: get border

Returns the specified border object.

Syntax
get border  text range/section/paragraph/font/Word style  Required. The object to get the border from.

[which border  enumeration]  Optional. The border to be retrieved. Can be one of the following:
    border top, border left, border bottom, border right, border horizontal, border vertical,
    border diagonal down, or border diagonal up.
Text Suite

Example
This example applies a border around the first character in the selection. If nothing is selected, the border is applied to the first character after the insertion point.

set visible of (get border character 1 of text object of selection) to true

This example applies a bottom border below all centered paragraphs in the active document.

set allPara to paragraphs in active document
repeat with x in allPara
    if alignment of x is align paragraph center then
        set line style of (get border text object of x which border border bottom) to line style single
        set line width of (get border text object of x which border border bottom) to line width 25 point
    end if
end repeat

This example adds a border around all of the pages in the current section.

set art style of (get border section 1 of selection which border border top) to art basic black dots
set art style of (get border section 1 of selection which border border bottom) to art basic black dots
set art style of (get border section 1 of selection which border border left) to art basic black dots
set art style of (get border section 1 of selection which border border right) to art basic black dots
set art width of (get border section 1 of selection) to 6

Command: get footer

Represents a single footer. The get footer command returns a header footer object.

Syntax
get footer  section  Required. An expression that returns a section object.
    index  enumeration  Required. Specifies the footer to retrieve. Can be one of the following: header footer primary, header footer first page, or header footer even pages.

You can also return a single header footer object by using the header footer object property with a selection object.

Example
The following example changes the text of both the primary header and the primary footer in the first section of the active document.

set content of text object of (get header section 1 of active document index ~ header footer primary) to "Header text"
set content of text object of (get footer section 1 of active document index ~ header footer primary) to "Footer text"
Command: get header

Represents a single header. The get header command returns a header footer object.

Syntax

get header  section  Required. An expression that returns a section object.

index  enumeration  Required. Specifies the header footer object to retrieve. Can be one of the following: header footer primary, header footer first page, or header footer even pages.

You can also return a single header footer object by using the header footer object property with a selection object.

Example

The following example changes the text of both the primary header and the primary footer in the first section of the active document.

set content of text object of (get header section 1 of active document index ¬
header footer primary) to "Header text"
set content of text object of (get footer section 1 of active document index ¬
header footer primary) to "Footer text"

Command: get range information

Returns information about the specified selection or range.

Syntax

generate information  text range  Required. An expression that returns a text range object.

information type  enumeration  Required. The information to return. Can be one of the following:

- active end adjusted page number  Returns the number of the page that contains the active end of the specified text range. If you set a starting page number or make other manual adjustments, returns the adjusted page number (unlike active end page number).

- active end page number  Returns the number of the page that contains the active end of the specified text range, counting from the beginning of the document. Any manual adjustments to page numbering are disregarded (unlike active end adjusted page number).

- active end section number  Returns the number of the section that contains the active end of the specified text range.

- at end of row marker  Returns True if the specified text range is at the end-of-row mark in a table.

- info caps lock  Returns True if Caps Lock is in effect.

- end _ of range column number  Returns the table column number that contains the end of the specified text range.
• **end_of_range_row_number**  Returns the table row number that contains the end of the specified text range.

• **first_character_column_number**  Returns the character position of the first character in the specified text range. If the text range is collapsed, the character number immediately to the right of the range or selection is returned (this is the same as the character column number displayed in the status bar after "Col").

• **first_character_line_number**  Returns the line number of the first character in the selection. If the pagination property of the Word options object is False or the draft property is True, returns -1.

• **frame_is_selected**  Returns True if the text range is an entire frame or text box.

• **header_footer_type**  Returns a value that indicates the type of header or footer that contains the specified text range, as shown in the following table.

<table>
<thead>
<tr>
<th>Value</th>
<th>Type of header or footer</th>
</tr>
</thead>
<tbody>
<tr>
<td>-1</td>
<td>None (the text range isn't in a header or footer)</td>
</tr>
<tr>
<td>0 (zero)</td>
<td>Even page header</td>
</tr>
<tr>
<td>1</td>
<td>Odd page header (or the only header, if there aren't odd and even headers)</td>
</tr>
<tr>
<td>2</td>
<td>Even page footer</td>
</tr>
<tr>
<td>3</td>
<td>Odd page footer (or the only footer, if there aren't odd and even footers)</td>
</tr>
<tr>
<td>4</td>
<td>First page header</td>
</tr>
<tr>
<td>5</td>
<td>First page footer</td>
</tr>
</tbody>
</table>

• **horizontal_position_relative_to_page**  Returns the horizontal position of the specified text range; this is the distance (in twips) from the left edge of the text range to the left edge of the page (20 twips = 1 point, 72 points = 1 inch). If the text range isn't within the screen area, returns -1.

• **horizontal_position_relative_to_page_boundary**  Returns the horizontal position (in twips) of the specified text range relative to the left edge of the nearest text boundary enclosing it (20 twips = 1 point, 72 points = 1 inch). If the text range isn't within the screen area, returns -1.

• **in_clipboard**  Returns True if the specified text range is on the Clipboard.

• **in_commentPane**  Returns True if the specified text range is in a comment pane.

• **in_endnote**  Returns True if the specified text range is in an endnote area in page layout view or in the endnote pane in normal view.

• **in_footnote**  Returns True if the specified text range is in a footnote area in page layout view or in the footnote pane in normal view.
- **in footnote endnote pane**  Returns True if the specified text range is in the footnote or endnote pane in normal view or in a footnote or endnote area in page layout view. For more information, see the descriptions of in footnote and in endnote in the preceding paragraphs.

- **in header footer**  Returns True if the text range is in the header or footer pane or in a header or footer in page layout view.

- **in master document**  Returns True if the text range is in a master document (that is, a document that contains at least one subdocument).

- **in word mail**  Returns a value that indicates the WordMail location of the text range, as shown in the following table.

<table>
<thead>
<tr>
<th>Value</th>
<th>WordMail Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 (zero)</td>
<td>The text range isn't in a WordMail message.</td>
</tr>
<tr>
<td>1</td>
<td>The text range is in a WordMail send note.</td>
</tr>
<tr>
<td>2</td>
<td>The text range is in a WordMail read note.</td>
</tr>
</tbody>
</table>

- **maximum number of columns**  Returns the greatest number of table columns within any row in the text range.

- **maximum number of rows**  Returns the greatest number of table rows within the table in the specified text range.

- **number of pages in document**  Returns the number of pages in the document associated with the text range.

- **info num lock**  Returns True if Num Lock is in effect.

- **over type**  Returns True if overtype mode is in effect. The overtype property of the Word options object can be used to change the state of overtype mode.

- **reference of type**  Returns a value that indicates where the selection is in relation to a footnote, endnote, or comment reference, as shown in the following table.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-1</td>
<td>The text range includes but isn't limited to a footnote, endnote, or comment reference.</td>
</tr>
<tr>
<td>0 (zero)</td>
<td>The text range isn't before a footnote, endnote, or comment reference.</td>
</tr>
<tr>
<td>1</td>
<td>The text range is before a footnote reference.</td>
</tr>
<tr>
<td>2</td>
<td>The text range is before an endnote reference.</td>
</tr>
<tr>
<td>3</td>
<td>The text range is before a comment reference.</td>
</tr>
</tbody>
</table>
• **revision marking**  Returns True if change tracking is in effect.

• **selection mode**  Returns a value that indicates the current selection mode, as shown in the following table.

<table>
<thead>
<tr>
<th>Value</th>
<th>Selection mode</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 (zero)</td>
<td>Normal selection</td>
</tr>
<tr>
<td>1</td>
<td>Extended selection (&quot;EXT&quot; appears on the status bar)</td>
</tr>
<tr>
<td>2</td>
<td>Column selection. (&quot;COL&quot; appears on the status bar)</td>
</tr>
</tbody>
</table>

• **start of range column number**  Returns the table column number that contains the beginning of the text range.

• **start of range row number**  Returns the table row number that contains the beginning of the text range.

• **vertical position relative to page**  Returns the vertical position of the text range; this is the distance (in twips) from the top edge of the selection to the top edge of the page (20 twips = 1 point, 72 points = 1 inch). If the selection isn't visible in the document window, returns -1.

• **vertical position relative to page boundary**  Returns the vertical position (in twips) of the text range relative to the top edge of the nearest text boundary enclosing it (20 twips = 1 point, 72 points = 1 inch). This is useful for determining the position of the insertion point within a frame or table cell. If the selection isn't visible, returns -1.

• **with in table**  Returns True if the selection is in a table.

• **zoom percentage**  Returns the current percentage of magnification as set by the percentage property of the zoom object.

**Example**

This example displays the current page number and the total number of pages in the active document.

display dialog "The selection is on page " & (get range information text object of selection information type active end page number) & " of page " & (get range information text object of selection information type number of pages in document)

If the selection is in a table, this example selects the table.

if (get range information text object of selection information type with in table) = "True" then select table 1 of selection

This example displays a message that indicates the current section number.

collapse range text object of selection direction collapse start

display dialog "The insertion point is in section " & (get range information text object of selection information type active end section number)
Command: go to next

Returns a text range object that refers to the start position of the next item or location specified by the what argument. If you apply this command to the text range of the selection object, the command moves the selection to the specified item (except for the goto a grammatical error, goto a proofreading error, and goto a spelling error constants).

Note When you use this command with the goto a grammatical error, goto a proofreading error, or goto a spelling error constant, the text range object that’s returned includes any grammar error text or spelling error text.

Syntax
go to next  text range  Required. An expression that returns a text range object.

  what  enumeration  Required. The item that the specified text range is to be moved to. Can be one of the following: goto a bookmark item, goto a comment item, goto an endnote item, goto an equation, goto a field item, goto a footnote item, goto a grammatical error, goto a graphic, goto a heading item, goto a line item, goto an object, goto a page item, goto a percent item, goto a proofreading error, goto a section item, goto a spelling error, or goto a table item.

Example
This example adds a bookmark at the top of page 2 in the active document.

set myRange to go to next (word 1 of active document) what goto a page item
make new bookmark at active document with properties {name:"Page2", ¬
text object:myRange}

Command: go to previous

Returns a text range object that refers to the start position of the previous item or location specified by the what argument. If applied to the text range of a selection object, the go to previous command moves the selection to the specified item.

Syntax
go to previous  text range  Required. An expression that returns a text range object.

  what  Required. The item that the specified text range is to be moved to. Can be one of the following: goto a bookmark item, goto a comment item, goto an endnote item, goto an equation, goto a field item, goto a footnote item, goto a grammatical error, goto a graphic, goto a heading item, goto a line item, goto an object, goto a page item, goto a percent item, goto a proofreading error, goto a section item, goto a spelling error, or goto a table item.

Example
This example moves the cursor to the previous field in the active document.

go to previous text object of selection what goto a field item
This example creates a range that references the last footnote reference marker in the active document.

set myRange to go to previous last word of active document what goto ¬
a footnote item
expand myRange by a character item
Command: in range

Returns True if the text range to which the command is applied is contained in the range specified by the text range argument.

Syntax

in range  text range  Required. An expression that returns a text range object.

text range  text range  Required. The text range to which you want to compare the expression.

This command determines whether the text range returned by the first expression is contained in the specified text range by comparing the starting and ending character positions, as well as the story type.

Example

This example determines whether the selection is contained in the first paragraph in the active document.

set status to in range text object of selection text range text object of paragraph 1 of active document

This example sets myRange equal to the first word in the active document. If myRange isn't contained in the selection, myRange is selected.

set myRange to word 1 of active document

if (in range myRange text range text object of selection) is false then
  select myRange

Command: in story

True if the text range to which this command is applied is in the same story as the text range specified by the text range argument.

Note A text range can belong to only one story.

Syntax

in story  text range  Required. An expression that returns a text range object.

text range  text range  Required. The text range object whose story is compared with the story that contains the expression.

Example

This example determines whether the selection is in the same story as the first paragraph in the active document. The message box displays "False" because the selection is in the primary header story and the first paragraph is in the main text story.

set view type of view of active window to page view

set seek view of view of active window to seek current page header

set same to (in story text object of selection text range text object of paragraph 1 of active document)

display dialog same
This example determines whether Range1 and Range2 are in the same story. If they are, bold formatting is applied to Range1.

set range1 to word 1 of selection
set range2 to set range text object of active document start 20 end 100
if (in story range1 text range range2) is true then
    set bold of range1 to true
end if

Command: indent

Indents one or more paragraphs by one level.

Note Using this command is equivalent to clicking the Increase Indent button under Bullets and Numbering on the Formatting Palette.

Syntax

indent paragraph Required. An expression that returns a Paragraph or Paragraphs object.

Example

This example indents all the paragraphs in the active document twice, and then it removes one level of the indent for the first paragraph.

indent paragraphs of active document
indent paragraphs of active document
outdent paragraph 1 of active document

Command: indent char width

Indents one or more paragraphs by a specified number of characters.

Syntax

indent char width paragraph/paragraph format Required. An expression that returns a paragraph or paragraph format object.

count integer Required. The number of characters by which the specified paragraphs are to be indented.

Using this command is equivalent to clicking the Increase Indent button under Bullets and Numbering on the Formatting Palette.

Example

This example indents the first paragraph of the active document by 10 characters.

indent char width paragraph 1 of active document count 10
Command: indent first line char width
Indents the first line of one or more paragraphs by a specified number of characters.

Syntax

indent first line char width paragraph\paragraph format Required. An expression that returns a paragraph or paragraph format object.

count integer Required. The number of characters by which the first line of each specified paragraph is to be indented.

Example
This example indents the first line of the first paragraph in the active document by 10 characters.
indent first line char width paragraph 1 of active document count 10

Command: is equivalent

True if the text range to which this command is applied is equal to the range specified by the text range argument. This command compares the starting and ending character positions, as well as the story type. If all three of these items are the same for both objects, the objects are equal.

Syntax

is equivalent text range Required. An expression that returns a text range object.

text range text range Required. The text range object that's compared with the expression.

Example
This example compares the selection with the second paragraph in the active document. If the selection isn't equal to the second paragraph, the second paragraph is selected.

if (is equivalent text object of selection text range text object of paragraph 2 of active document) is false then
    select text object of paragraph 2 of active document
end if

This example compares Range1 with Range2 to determine whether they're equal. If the two ranges are equal, the content of Range1 is deleted.

set range1 to word 1 of selection
set range2 to word 3 of active document
if (is equivalent range1 text range range2) is true then
    delete range1
end if
Command: link to list template

Links the specified style to a list template so that the style's formatting can be applied to lists.

Syntax

**link to list template**  Word style  Required. An expression that returns a **Word style** object.

**list template**  list template  Required. The list template that the style is to be linked to.

**[list level number integer]**  Optional. An integer corresponding to the list level that the style is to be linked to. If this argument is omitted, then the level of the style is used.

Example

This example changes the formatting of the second outline list template and then links heading styles 1 through 9 to levels 1 through 9. The list template is then applied to the document. Any paragraphs formatted as heading styles will assume the numbering from the list template.

```plaintext
set LT to list template 2 of (get list gallery outline number gallery)
repeat with x from 1 to 9
    set number style of list level x of LT to list number style arabic
    set number position of list level x of LT to inches to points inches ~
        (0.25 * (x-1))
    set text position of list level x of LT to inches to points inches ~
        (0.25 * x)
    set number format of list level x of LT to "%" & x & "."
    link to list template Word style (("Heading " & x) as text)~
        of active document list template LT
end repeat
apply list format template list format of text object of active document ~
    list template LT
```

Command: merge subdocuments

Merges the specified subdocuments of a master document into a single subdocument.

Syntax

**merge subdocuments**  document  Required. An expression that returns a master document that contains the subdocuments to be merged.

**[first subdocument subdocument]**  Optional. The first subdocument in a range of subdocuments to be merged.

**[last subdocument subdocument]**  Optional. The last subdocument in a range of subdocuments to be merged.
Example
This example merges the first and second subdocuments in the active document into one
subdocument.
if (count of subdocuments of active document) ≥ 2 then
  set aDoc to active document
  merge subdocuments aDoc first subdocument subdocument 1 of aDoc last ¬
    subdocument subdocument 2 of aDoc
end if

Command: modify enclosure
Adds, modifies, or removes an enclosure around the specified character or characters.

Syntax
modify enclosure  text range  Required. An expression that returns a text range object.
  enclosure style  enumeration  Required. The style of the enclosure. Can be one of the following:
    enclose style none, enclose style small, or enclose style large.
  [enclosure type  enumeration]  Optional. The symbol in which to enclose the specified range.
    Can be one of the following: enclosure circle, enclosure square, enclosure triangle, or
    enclosure diamond. The default is enclosure circle.
  [enclosed text  Unicode text]  Optional. The characters that you want to enclose. If you include
    this argument, Word replaces the specified text range with the enclosed characters. If you
    don’t specify text to enclose, Word encloses all text in the specified range.

Example
This example replaces the current selection with the number 25 enclosed in a diamond.
modify enclosure text object of selection enclosure style enclose style large ¬
  enclosure type enclosure diamond enclosed text "25"

Command: move end of range
Moves the ending character position of a text range object. Use this command to return a new text
range object.

Syntax
move end of range  text range  Required. An expression that returns a text range object.
  [by  enumeration]  Optional. The unit by which the collapsed text range is to be moved. Can be
    one of the following: a character item, a word item, a sentence item, a paragraph item, a line
    item, a story item, a screen, a section, a column, a row, a window, a cell, a character
    formatting, a paragraph formatting, a item unit, or a table. The default value is a character
    item.
  [count  integer]  Optional. The number of units to move. If this number is positive, the ending
    character position is moved forward in the document. If this number is negative, the end is
    moved backward. If the ending position overtakes the starting position, the text range
    collapses and both character positions move together. The default value is 1.
Text Suite

Example
This example sets myRange to be equal to the second word in the active document. The move end of range command is used to move the ending position of myRange (a range object) forward one word. After this script is run, the second and third words in the document are selected.

```lisp
if (count of words in active document) ≥ 3 then
  set myRange to word 2 of active document
  set myRange to move end of range myRange by a word item count 1
  select myRange
end if
```

Command: move range
Collapses the specified text range to its start position or end position and then moves the collapsed object by the specified number of units. Use this command to return a new text range object.

Syntax
move range  text range  Required. An expression that returns a text range object.

[by  enumeration]  Optional. The unit by which the collapsed text range is to be moved. Can be one of the following: a character item, a word item, a sentence item, a paragraph item, a line item, a story item, a screen, a section, a column, a row, a window, a cell, a character formatting, a paragraph formatting, a item unit, or a table. The default value is a character item.

[count  integer]  Optional. The number of units by which the specified text range is to be moved. If count is a positive number, the object is collapsed to its end position and moved forward in the document by the specified number of units. If count is a negative number, the object is collapsed to its start position and moved backward by the specified number of units. The default value is 1. You can also control the collapse direction by using the collapse range command before using the move range command.

  If the text range is in the middle of a unit or isn't collapsed, moving it to the beginning or end of the unit counts as moving it one full unit.

The start position and end position of a collapsed text range are equal.

Applying the move range command to a text range doesn't rearrange text in the document. Instead, it redefines the range to refer to a new location in the document.

If you apply the move range command to any text range other than a text range object variable (for example, move range text object of paragraph 3 of selection), the command has no effect.

Example
This example sets range1 to the first paragraph in the active document and then moves the range forward three paragraphs. After this script is run, the insertion point is positioned at the beginning of the fourth paragraph.

```lisp
set range1 to text object of paragraph 1 of active document
collapse range range1 direction collapse start
set range1 to move range range1 by a paragraph item count 3
```
Text Suite

Command: move range end until
Moves the end position of the specified text range until any of the specified characters are found in the document. If the movement is forward in the document, the text range is expanded. Use this command to return a new text range object.

Syntax
move range end until  text range  Required. An expression that returns a text range object.
characters  Unicode text  Required. One or more characters. This argument is case sensitive.
[count  integer\constant]  Optional. The maximum number of characters by which the specified range or selection is to be moved. Can be a number or either the go forward or go backward constant. If count is a positive number, the text range is moved forward in the document. If it's a negative number, the text range is moved backward. The default value is go forward.

Example
This example extends the selection forward in the document until the letter "a" is found, and then expands the selection to include the letter "a".
set selObj to move range end until (text object of selection) characters "a" ~
count go forward
select selObj
extend selection character "a"

Command: move range end while
Moves the ending character position of a range or selection while any of the specified characters are found in the document. Use this command to return a new text range object.

Syntax
move range end while  text range  Required. An expression that returns a text range object.
characters  Unicode text  Required. One or more characters. This argument is case sensitive.
[count  integer\constant]  Optional. The maximum number of characters by which the range or selection is to be moved. Can be a number or either the go forward or go backward constant. If count is a positive number, the text range is moved forward in the document. If it's a negative number, the range or selection is moved backward. The default value is go forward.

Example
This example moves the end position of the selection forward while the space character is found.
set myRange to move range end while text object of selection characters " " ~
count go forward
This example moves the end position of the selection forward while count is less than or equal to 10 and any letter from 'a' through 'h' is found.
set myRange to move range end while text object of selection ~
characters "abcdefgh" count 10
Text Suite

Command: move range start until

Moves the start position of the specified text range until one of the specified characters is found in
the document. If the movement is backward through the document, the text range is expanded. Use
this command to return a new text range object.

Syntax

move range start until  text range  Required. An expression that returns a text range object.

characters  Unicode text  Required. One or more characters. This argument is case sensitive.

[count  integer\constant]  Optional. The maximum number of characters by which the specified
range or selection is to be moved. Can be a number or either the go forward or go backward
constant. If count is a positive number, the text range is moved forward in the document. If it’s
a negative number, the text range is moved backward. The default value is go forward.

Example

If there’s a dollar sign character ($) in the first paragraph in the selection, this example moves
myRange just before the dollar sign.

set myRange to text object of paragraph 1 of active document
set len to (end of content of myRange) - (start of content of myRange)
collapse range myRange direction collapse start
set myRange to move range start until myRange characters "$" count len

Command: move range start while

Moves the start position of the specified text range while any of the specified characters are found in
the document. Use this command to return a new text range object.

Syntax

move range start while  text range  Required. An expression that returns a text range object.

characters  Unicode text  Required. One or more characters. This argument is case sensitive.

[count  integer\constant]  Optional. The maximum number of characters by which the specified
range or selection is to be moved. Can be a number or either the go forward or go backward
constant. If count is a positive number, the text range is moved forward in the document. If it’s
a negative number, the text range is moved backward. The default value is go forward.

Example

This example moves the start position of the text range of the selection backward through the
document while the space character is found.

set strt to move range start while text object of selection characters " " -
  count go backward

This example moves the start position of the text range backward through the document while
count is less than or equal to 10 and any letter from 'a' through 'h' is found.

set strt to move range start while text object of selection characters -
  "abcdefgh" count -10
Command: move range until

Moves the specified range or selection until one of the specified characters is found in the document. Use this command to return a new text range.

**Syntax**

```
move range until  text range  Required. An expression that returns a text range object.
characters  Unicode text  Required. One or more characters. This argument is case sensitive.
[count  integer\constant] Optional. The maximum number of characters by which the specified text range is to be moved. Can be a number or either the go forward or go backward constant. If count is a positive number, the text range is moved forward in the document. If it's a negative number, the text range is moved backward. The default value is go forward.
```

**Example**

This example moves myRange forward through the next 100 characters in the document until the character "t" is found.

```
set myRange to word 1 of active document
set myRange to move range until myRange characters "t" count 100
```

Command: move range while

Moves the specified text range while any of the specified characters are found in the document. Use this command to return a new text range object.

**Syntax**

```
move range while  text range  Required. An expression that returns a text range object.
characters  Unicode text  Required. One or more characters. This argument is case sensitive.
[count  integer\constant] Optional. The maximum number of characters by which the specified text range is to be moved. Can be a number or either the go forward or go backward constant. If count is a positive number, the text range is moved forward in the document. If it's a negative number, the text range is moved backward. The default value is go forward.
```

**Example**

This example moves aRange while any of the following (uppercase or lowercase) letters are found: "a", "t", or "i".

```
set aRange to character 1 of active document
set aRange to move range while aRange characters "atiATI" count go forward
```
**Command: move start of range**

Moves the start position of the specified text range. Use this command to return a new text range object.

**Syntax**

```plaintext
move start of range  text range  Required. An expression that returns a text range object.

[by  enumeration]  Optional. The unit by which the collapsed text range is to be moved. Can be one of the following: a character item, a word item, a sentence item, a paragraph item, a line item, a story item, a screen, a section, a column, a row, a window, a cell, a character formatting, a paragraph formatting, a item unit, or a table. The default value is a character item.

[count  integer]  Optional. The maximum number of units by which the specified text range is to be moved. If count is a positive number, the start position of the text range is moved forward in the document. If it’s a negative number, the start position is moved backward. If the start position is moved forward to a position beyond the end position, the text range is collapsed and both the start and end positions are moved together. The default value is 1.
```

**Example**

This example moves the start position of the text range one character forward (the text range size is reduced by one character). Note that a space is considered a character.

```plaintext
set myRange to move start of range text object of selection by a character ¬
            item count 1
```

This example sets myRange to be equal to the second word in the active document. The example uses the move start of range command to move the start position of myRange (a text range object) backward one word. After this script is run, the first and second words in the document are selected.

```plaintext
if (count of words in active document) ≥ 2 then
    set myRange to word 2 of active document
    set myRange to move start of range myRange by a word item count ¬
            1
    select myRange
end if
```

**Command: navigate**

Returns a text range object that refers to the start position of the next item or the location specified by the to argument.

**Note** When you use this command with the goto grammatical error, goto proofreading error, or goto spelling error constant, the text range object that’s returned includes any grammar error text or spelling error text.
Text Suite

**Syntax**

navigate  text range  Required. An expression that returns a text range object.

to  enumeration  Required. The item that the specified text range is to be moved to. Can be one of the following: goto a bookmark item, goto a comment item, goto an endnote item, goto an equation, goto a field item, goto a footnote item, goto a grammatical error, goto a graphic, goto a heading item, goto a line item, goto an object, goto a page item, goto a percent item, goto a proofreading error, goto a section item, goto a spelling error, or goto a table item.

[position  enumeration]  Optional. The type of navigation to be performed. Can be one of the following: the first item, the last item, the next item, relative, the previous item, or absolute.
[count  integer]  Optional. The number of items by which to navigate. The default value is one.
[name  Unicode text]  Optional. If the to argument is goto a bookmark item, goto a comment item, goto a field item, or goto an object, this argument specifies a name.

**Example**

This example moves the insertion point to the first cell in the next table.

navigate text object of selection to goto a table item position the next item

This example moves the insertion point just before the fifth endnote reference mark in the active document.

if (count of endnotes of active document) ≥ 5 then
  navigate text object of selection to goto an endnote item ~
  position absolute count 5
end if

This example sets R1 equal to the first footnote reference mark in the active document.

if (count of footnotes of active document) ≥ 1 then
  set R1 to navigate text object of active document to goto a footnote item ~
  position the first item
end if

This example moves the insertion point down four lines.

navigate text object of selection to goto a line item position relative ~
  count 4

**Command: next paragraph**

Returns the next paragraph as a paragraph object.

**Syntax**

next paragraph  paragraph  Required. An expression that returns a paragraph object.

**Example**

This example selects the paragraph following the current selection.

select (next paragraph paragraph 1 of selection)
Command: next range

Returns a new text range object relative to the specified text range.

Note If the original text range is just before the specified unit, the new text range is moved to the following unit. For example, if the text range of the selection is just before a word, the following instruction moves the text range forward to the following word.

set myRange to next range text object of selection by a word item unit 1

Syntax

next range  text range  Required. An expression that returns a text range object.

[by enumeration]  Optional. Can be one of the following: a character item, a word item, a sentence item, a paragraph item, a line item, a story item, a screen, a section, a column, a row, a window, a cell, a character formatting, a paragraph formatting, a table, or a item unit.

[count integer]  Optional. The number of units by which you want to move ahead. The default value is 1.

Example

This example selects the paragraph following the current selection.

select (next range (text object of selection)) by a paragraph item count 1)

Command: next subdocument

Moves the text range to the next subdocument. If there isn’t another subdocument, an error occurs.

Syntax

next subdocument  text range  Required. An expression that returns a text range object.

Example

This example switches the active document to master document view and selects the first subdocument.

if (count of subdocuments in active document) ≥ 1 then
    set view type of view of active window to master view
    home key selection move unit a story extend by moving
    select next subdocument text object of selection
end if

Command: open or close up

If spacing before the specified paragraphs is 0 (zero), this command sets spacing to 12 points. If spacing before the paragraphs is greater than 0 (zero), this command sets spacing to 0 (zero).

Syntax

open or close up  paragraph/paragraph format  Required. An expression that returns a paragraph or paragraph format object.
Example
This example toggles the formatting of the first paragraph in the active document to either add 12 points of space before the paragraph or leave no space before it.

open or close up paragraph 1 of active document

Command: open up

Sets spacing before the specified paragraphs to 12 points.

Syntax

open up  paragraph/paragraph format  Required. An expression that returns a paragraph or paragraph format object.

Remarks

The following two statements are equivalent:

open up paragraph 1 of active document
set space before of paragraph 1 of active document to 12

Example

This example changes the formatting of the second paragraph in the active document to leave 12 points of space before the paragraph.

open up paragraph 2 of active document

Command: outdent

Removes one level of indent for one or more paragraphs.

Note  Using this command is equivalent to clicking the Decrease Indent button under Bullets and Numbering on the Formatting toolbar.

Syntax

outdent  paragraph  Required. An expression that returns a paragraph or list of paragraph objects.

Example

This example indents all the paragraphs in the active document twice, and then it removes one level of the indent for the first paragraph.

indent paragraphs of active document
indent paragraphs of active document
outdent paragraph 1 of active document

Command: outline demote

Applies the next heading level style (Heading 1 through Heading 8) to the specified paragraph or paragraphs. For example, if a paragraph is formatted with the Heading 2 style, this command demotes the paragraph by changing the style to Heading 3.
Syntax

outline demote  paragraph  Required. An expression that returns a paragraph or list of paragraph objects.

Example
This example demotes the selected paragraphs.
outline demote paragraphs of selection
This example demotes the third paragraph in the active document.
outline demote paragraph 3 of active document

Command: outline demote to body
Demotes the specified paragraph or paragraphs to body text by applying the Normal style.

Syntax

outline demote to body  paragraph  Required. An expression that returns a paragraph or list of paragraph objects.

Example
This example demotes the selected paragraphs to body text by applying the Normal style.
outline demote to body paragraphs of selection
This example switches the active window to outline view and demotes the first paragraph in the selection to body text.
set view type of view of active window to outline view
outline demote to body paragraph 1 of selection

Command: outline promote
Applies the previous heading level style (Heading 1 through Heading 8) to the specified paragraph or paragraphs. For example, if a paragraph is formatted with the Heading 2 style, this command promotes the paragraph by changing the style to Heading 1.

Syntax

outline promote  paragraph  Required. An expression that returns a paragraph or list of paragraph objects.

Example
This example promotes the selected paragraphs.
outline promote paragraphs of selection
This example switches the active window to outline view and promotes the first paragraph in the active document.
set view type of view of active window to outline view
outline promote paragraph 1 of active document
**Command: paste and format**

Pastes the contents of the clipboard into the selected cells and formats them as specified.

**Syntax**

*paste and format*  text range  Required. An expression that returns the text range of the cells in which you want to paste the contents of the Clipboard.

*type*  enumeration  Required. Specifies the type of formatting to use when pasting the selected table cells. Can be one of the following:

<table>
<thead>
<tr>
<th>Constant</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>chart</td>
<td>Pastes a Microsoft Excel chart as an embedded OLE object.</td>
</tr>
<tr>
<td>chart linked</td>
<td>Pastes an Excel chart and links it to the original Excel spreadsheet.</td>
</tr>
<tr>
<td>chart picture</td>
<td>Pastes an Excel chart as a picture.</td>
</tr>
<tr>
<td>format original formatting</td>
<td>Preserves original formatting of the pasted material.</td>
</tr>
<tr>
<td>format plain text</td>
<td>Pastes as plain, unformatted text.</td>
</tr>
<tr>
<td>format surrounding formatting</td>
<td>Matches the formatting of the pasted text to the formatting of surrounding text.</td>
</tr>
<tr>
<td>with emphasis</td>
<td></td>
</tr>
<tr>
<td>list combine with existing list</td>
<td>Merges a pasted list with neighboring lists.</td>
</tr>
<tr>
<td>list continue numbering</td>
<td>Continues numbering of a pasted list from the list in the document.</td>
</tr>
<tr>
<td>list restart numbering</td>
<td>Restarts numbering of a pasted list.</td>
</tr>
<tr>
<td>single cell table</td>
<td>Pastes a single cell table as a separate table.</td>
</tr>
<tr>
<td>single cell text</td>
<td>Pastes a single cell as text.</td>
</tr>
<tr>
<td>table append table</td>
<td>Merges pasted cells into an existing table by inserting the pasted rows between the selected rows.</td>
</tr>
<tr>
<td>table insert as rows</td>
<td>Inserts a pasted table as rows between two rows in the target table.</td>
</tr>
<tr>
<td>table original formatting</td>
<td>Pastes an appended table without merging table styles.</td>
</tr>
<tr>
<td>table overwrite cells</td>
<td>Pastes table cells and overwrites existing table cells.</td>
</tr>
</tbody>
</table>

**Example**

This example pastes a selected Excel chart as a picture. This example assumes that the Clipboard contains an Excel chart.

*paste and format (text object of selection) type chart picture*
Command: paste append table

Merges the contents of the Clipboard into an existing table by inserting the pasted rows between the specified rows. No cells are overwritten.

Syntax

paste append table  text range  Required. An expression that returns the text range of the cells that you want to append the contents of the Clipboard to.

You can use the paste append table command only if the Clipboard contains a cell or group of cells and the selected text range is a cell or group of cells in the current document.

Example

This example pastes table cells into row 4 of the first table in the active document. This example assumes that the Clipboard contains a list of table cells.

set r4c1 to (get cell from table table 1 of active document row 4 column 1)
paste append table text object of r4c1

Command: paste as nested table

Pastes a cell or group of cells as a nested table into the specified text range.

Syntax

paste as nested table  text range  Required. An expression that returns a text range object where you want to paste the cells.

You can use the paste as nested table command only if the Clipboard contains a cell or group of cells and the selected text range is a cell or group of cells in the current document.

Example

This example pastes the contents of the Clipboard into the third cell of the first table in the active document.

paste as nested table text object of cell 3 of row 1 of table 1 of active document

Command: paste excel table

Pastes and formats a Microsoft Excel table.

Syntax

paste excel table  text range  Required. An expression that returns a text range object where you want to paste the table.

  linked to Excel  Boolean  Required. True links the pasted table to the original Excel file so that changes made to the Excel file are reflected in Word.

  word formatting  Boolean  Required. True formats the table using the formatting in the Word document. False formats the table according to the original Excel file.

  RTF  Boolean  Required. True pastes the Excel table using Rich Text Format (RTF). False pastes the Excel table as HTML.
Example
This example pastes an Excel table into the active document. The parameters specify that the pasted table is linked to the Excel file, retains the original Excel formatting, and is pasted as RTF. This example assumes that the Clipboard contains an Excel table.

\[
paste\text{ excel table text object of selection linked to Excel true word ~} \\
\text{ formatting false RTF true}
\]

Command: paste object
Inserts the contents of the Clipboard at the specified text range. If you don't want to replace the contents of the text range, use the collapse range command before using this command.

Syntax
\text{paste object text range} \text{ Required. An expression that returns a text range object. The location where you want to paste the object.}

Example
This example copies and pastes the first table in the active document into a new document.

\[
\text{if (count of tables in active document) } \geq \text{ 1 then} \\
\text{ select table 1 of active document} \\
\text{ copy object selection} \\
\text{ set newDoc to make new document} \\
\text{ paste object text object of newDoc}
\]

end if

This example copies the first paragraph in the document and pastes it at the insertion point.

\[
\text{copy object text object of paragraph 1 of active document} \\
\text{ collapse range text object of selection direction collapse start} \\
\text{ paste object text object of selection}
\]

This example copies the selection and pastes it at the end of the document.

\[
\text{if selection type of selection is not selection ip then} \\
\text{ copy object selection} \\
\text{ set range2 to text object of active document} \\
\text{ set range2 to collapse range range2 direction collapse end} \\
\text{ paste object range2}
\]

end if

Command: paste special
Inserts the contents of the Clipboard. Unlike with the paste command, the paste special command allows you to control the format of the pasted information and (optionally) establish a link to the source file (for example, a Microsoft Excel worksheet).

Note If you don't want to replace the contents of the specified text range, use the collapse range command before you use this command. When you use this command, the text range doesn't expand to include the contents of the Clipboard.
Text Suite

Syntax

`paste special`  text range  Required. An expression that returns a `text range` object representing the location where you want to paste the object.

- [link] Boolean  Optional. True to create a link to the source file of the Clipboard contents. The default value is false.
- [placement] enumeration  Optional. Can be either of the following: float over text or in line. The default value is in line.
- [display as icon] Boolean  Optional. True to display the link as an icon. The default value is false.
- [data type] enumeration  Optional. A format for the Clipboard contents when they’re inserted into the document. Can be one of the following: `paste oleobject`, `paste rtf`, `paste text`, `paste metafile picture`, `paste bitmap`, `paste device independent bitmap`, `paste hyperlink`, `paste shape`, `paste enhanced metafile`, `paste styled text`, or `paste html`. The default format varies, depending on the contents of the Clipboard.
- [icon label] Unicode text  Optional. If `display as icon` is true, this argument is the text that appears below the icon.

Example

This example inserts the Clipboard contents at the insertion point as unformatted text.

```plaintext
set myRange to collapse range text object of selection direction collapse start
paste special myRange data type paste text
```

This example copies the selected text and pastes it into a new document as a hyperlink. The source document must first be saved for this example to work.

```plaintext
if selection type of selection is selection normal then
    copy object selection
    set newDoc to make new document
    paste special text object of newDoc data type paste hyperlink with link
end if
```

**Command: previous paragraph**

Returns a `paragraph` object relative to the specified paragraph.

**Syntax**

`previous paragraph`  paragraph  Required. An expression that returns a `paragraph` object.

**Example**

This example selects the paragraph that precedes the selection in the active document.

```plaintext
set prevPara to previous paragraph paragraph 1 of selection
select text object of prevPara
```
Command: previous range

Returns a text range object relative to the specified selection or range.

Note  If the text range is just after the specified unit, the text range is moved to the previous unit. For example, if the selection is just after a word (before the trailing space), the following instruction moves the selection backwards to the previous word.

```plaintext
set newSel to (previous range text object of selection by a word item count 1)
select newSel
```

Syntax

`previous range`  text range  Required. An expression that returns a text range object.

[by  enumeration]  Optional. Can be one of the following: a character item, a word item, a sentence item, a paragraph item, a section item, a story item, a screen, a section, a column, a row, a window, a cell, a character formatting, a paragraph formatting, a table, or a item unit.
The default value is a character item.

[count  integer]  Optional. The number of units by which you want to move back. The default value is 1.

Example

This example selects the paragraph that precedes the selection in the active document.

```plaintext
set newSel to previous range text object of selection by a paragraph item count 1
select newSel
```

This example applies bold formatting to the first word in the active document.

```plaintext
set newSel to previous range word 2 of active document by a word item count 1
set bold of newSel to true
```

Command: previous subdocument

Moves the range or selection to the previous subdocument. If there isn't another subdocument, an error occurs.

Syntax

`previous subdocument`  text range  Required. An expression that returns a text range object.

Example

This example switches the active document to master document view and selects the previous subdocument.

```plaintext
if (count of subdocument in active document) ≥ 1 then
  set view type of view of active window to master view
  end key selection move unit a story extend by moving
  select (previous subdocument text object of selection)
end if
```
Command: relocate

In outline view, moves the paragraphs within the specified text range after the next visible paragraph or before the previous visible paragraph. Body text moves with a heading only if the body text is collapsed in outline view or if it's part of the range.

Syntax

relocate  text range  Required. An expression that returns a text range object.

direction  enumeration  Required. The direction of the move. Can be either of the following: relocate up or relocate down.

Example

This example moves the third, fourth, and fifth paragraphs in the active document below the next (sixth) paragraph.

set theStart to start of content of text object of paragraph 3 of the active document

set theEnd to end of content of text object of paragraph 5 of the active document

set myRange to set range text object of the active document start theStart end theEnd

set view type of view of the active window to outline view

relocate myRange direction relocate down

This example moves the first paragraph in the selection above the previous paragraph.

set view type of view of the active window to outline view

relocate text object of paragraph 1 of selection direction relocate up

Command: reset

Removes manual paragraph formatting (formatting not applied using a style). For example, if you manually right align a paragraph and the underlying style has a different alignment, the reset command changes the alignment to match the formatting of the underlying style.

Syntax

reset  paragraph/paragraph format  Required. An expression that returns a paragraph or paragraph format object.

Example

This example removes manual paragraph formatting from the second paragraph in the active document.

reset paragraph 2 of the active document
**Command: set range**

Sets the starting and ending character positions for the text range.

*Note*  Character position values start at the beginning of the story, with the first value being 0 (zero). All characters are counted, including nonprinting characters. Hidden characters are counted even if they’re not displayed.

**Syntax**

```plaintext
set range  text range  Required. An expression that returns a text range object.

start  integer  Required. The starting character position of the text range.

end  integer  Required. The ending character position of the text range.
```

**Remarks**

The `set range` command redefines the starting and ending positions of an existing `range` object. This command differs from the `create range` command, which is used to create a text range, given a starting and ending position.

**Example**

This example selects the first 10 characters in the document.

```plaintext
select (set range text object of active document start 0 end 10)
```

This example uses `set range` to redefine `myRange` to refer to the first three paragraphs in the active document.

```plaintext
set myRange to create range active document start (start of content of text object of paragraph 1 of active document) end (end of content of text object of paragraph 1 of active document)

set myRange to set range myRange start (start of content of text object of paragraph 1 of active document) end (end of content of text object of paragraph 3 of active document)
```

**Command: sort ascending**

Sorts paragraphs in ascending alphanumeric order. The first paragraph is considered a header record and isn't included in the sort. Use the `sort` command to include the header record in a sort.

*Note*  This command offers a simplified form of sorting intended for data merge data sources that contain columns of data. For most sorting tasks, use the `sort` command.

**Syntax**

```plaintext
sort ascending  text range  Required. An expression that returns a text range object.
```

**Example**

This example sorts the paragraphs in the active document in ascending order.

```plaintext
sort ascending text object of active document
```
Text Suite

Command: sort

Sorts the paragraphs in the specified range or selection.

Syntax

sort  text range  Required. An expression that returns a text range object.

[exclude header  Boolean]  Optional. True to exclude the first row or paragraph from the sort operation. The default value is False.

[field number  integer]  Optional. The fields to sort by. Word sorts by field number, then by field number two, and then by field number three.

[sort field type  enumeration]  Optional. The respective sort types for field number, field number two, and field number three. Can be one of the following: sort field alphanumeric, sort field date, sort field syllable, sort field japan jis, sort field stroke, or sort field korea ks. The default value is sort field alphanumeric.

[sort order  enumeration]  Optional. The sorting order to use when sorting field number, field number two, and field number three. Can be one of the following: sort order ascending or sort order descending. The default value is sort order ascending.

[field number two  integer]  Optional. The fields to sort by. Word sorts by field number, then by field number two, and then by field number three.

[sort field type two  enumeration]  Optional. The respective sort types for field number, field number two, and field number three. Can be one of the following: sort field alphanumeric, sort field date, sort field syllable, sort field japan jis, sort field stroke, or sort field korea ks. The default value is sort field alphanumeric.

[sort order two  enumeration]  Optional. The sorting order to use when sorting field number, field number two, and field number three. Can be one of the following: sort order ascending or sort order descending. The default value is sort order ascending.

[field number three  integer]  Optional. The fields to sort by. Word sorts by field number, then by field number two, and then by field number three.

[sort field type three  enumeration]  Optional. The respective sort types for field number, field number two, and field number three. Can be one of the following: sort field alphanumeric, sort field date, sort field syllable, sort field japan jis, sort field stroke, or sort field korea ks. The default value is sort field alphanumeric.

[sort order three  enumeration]  Optional. The sorting order to use when sorting field number, field number two, and field number three. Can be one of the following: sort order ascending or sort order descending. The default value is sort order ascending.

[sort column  Boolean]  Optional. True to sort only the column specified by the table object.

[separator  enumeration]  Optional. The type of field separator. Can be one of the following: sort separate by commas, sort separate by default table separator, or sort separate by tabs. The default value is sort separate by commas.

[case sensitive  Boolean]  Optional. True to sort with case sensitivity. The default value is False.
[language id  enumeration]  Optional. Specifies the sorting language.

Can be one of the following:

afrikaans  language none
arabic  latvian
basque  macedonian
belgian dutch  malay
belgian french  malaysian
brazilian portuguese  mexican Spanish
bulgarian  language no proofing
byelorussian  norwegian bokmål
catalan  norwegian nynorsk
croatian  polish
czech  portuguese
danish  romanian
dutch  russian
dutch  serbian cyrillic
english aus  serbian latin
english canadian  sesotho
english new zealand  simplified Chinese
english south africa  slovak
english uk  slovenian
english us  spanish
estonian  spanish modern sort
farsi  swedish
finnish  swiss French
french  swiss German
french canadian  swiss italian
german  traditional Chinese
greek  tswana
hebrew  turkish
hungarian  ukrainian
icelandic  venda
japanese  xhosa
korean  zulu

Example
This example inserts three lines of text into a new document and then sorts the lines in ascending alphanumeric order.

set newDoc to make new document
set content of text object of newDoc to "pear" & return & "zucchini" & return ¬
& "apple" & return
sort text object of newDoc sort order sort order ascending
Command: sort descending

Sorts paragraphs in descending alphanumeric order. The first paragraph is considered a header record and isn't included in the sort. Use the sort command to include the header record in a sort.

Note  This command offers a simplified form of sorting intended for data merge data sources that contain columns of data. For most sorting tasks, use the sort command.

Syntax

sort descending  text range  Required. An expression that returns a text range object.

Example

This example sorts the paragraphs in the active document in ascending order.

sort descending text object of active document

Command: space 1

Single-spaces the specified paragraphs. The exact spacing is determined by the font size of the largest characters in each paragraph.

Syntax

space 1  paragraph/paragraph format  Required. An expression that returns a paragraph or paragraph format object.

Remarks

The following two statements are equivalent:

space 1 paragraph 1 of active document

set line spacing rule of paragraph 1 of active document to line space single

Example

This example changes the first paragraph in the active document to single spacing.

space 1 paragraph 1 of active document

Command: space 15

Formats the specified paragraphs with 1.5-line spacing. The exact spacing is determined by adding 6 points to the font size of the largest character in each paragraph.

Syntax

space 15  paragraph/paragraph format  Required. An expression that returns a paragraph or paragraph format object.

Remarks

The following two statements are equivalent:

space 15 paragraph 1 of active document

set line spacing rule of paragraph 1 of active document to line space 1 pt 5

Example

This example changes the first paragraph in the active document to 1.5-line spacing.

space 15 paragraph 1 of active document
**Command: space 2**

Double-spaces the specified paragraphs. The exact spacing is determined by adding 12 points to the font size of the largest character in each paragraph.

**Syntax**

```
space 2  paragraph/paragraph format  Required.  An expression that returns a paragraph or paragraph format object.
```

**Remarks**

The following two statements are equivalent:

```
space 2 paragraph 1 of active document
set line spacing rule of paragraph 1 of active document to line space double
```

**Example**

This example changes the first paragraph in the selection to double spacing.

```
space 2 paragraph 1 of selection
```

**Command: tab hanging indent**

Sets a hanging indent to a specified number of tab stops. Can be used to remove tab stops from a hanging indent if the value of `count` is a negative number.

**Syntax**

```
tab hanging indent  paragraph/paragraph format  Required.  An expression that returns a paragraph or paragraph format object.

count  integer  Required.  The number of tab stops to indent (if positive) or the number of tab stops to remove from the indent (if negative).
```

**Example**

This example sets a hanging indent to the second tab stop for the first paragraph in the active document.

```
tab hanging indent paragraph 1 of active document count 2
```

This example moves the hanging indent back one tab stop for the first paragraph in the active document.

```
tab hanging indent paragraph 1 of active document count -1
```

**Command: tab indent**

Sets the left indent for the specified paragraphs to a specified number of tab stops. Can also be used to remove the indent if the value of `count` is a negative number.

**Syntax**

```
tab indent  paragraph/paragraph format  Required.  An expression that returns a paragraph or paragraph format object.

count  integer  Required.  The number of tab stops to indent (if positive) or the number of tab stops to remove from the indent (if negative).
```
Example
This example indents the first paragraph in the active document to the second tab stop.

```
  tab indent paragraph 1 of the active document count 2
```

This example moves the indent of the first paragraph in the active document back one tab stop.

```
  tab indent paragraph 1 of the active document count -1
```

Command: **text range spelling suggestions**

Returns a record that represents the spelling error type and the list of words suggested as spelling replacements for the first word in the specified range. The spelling error type can be one of the following type class types: spelling correct, spelling not in dictionary, or spelling capitalization.

Syntax

```
text range spelling suggestions  text range  Required. An expression that returns a text range object.

  [custom dictionary  dictionary]  Optional. Either an expression that returns a dictionary object or the file name of the custom dictionary.

  [ignore uppercase  Boolean]  Optional. True to ignore words in all uppercase letters. If this argument is omitted, the current value of the ignore uppercase property is used.

  [main dictionary  dictionary]  Optional. Either an expression that returns a dictionary object or the file name of the main dictionary. If you don't specify a main dictionary, Word uses the main dictionary that corresponds to the language formatting of the first word in the text range.

  [suggestion mode  enumeration]  Optional. Specifies the way Word makes spelling suggestions. Can be one of the following constants. The default value is spelling word type spell word.
```

<table>
<thead>
<tr>
<th>Constant</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>spelling word type spell word</td>
<td>Word suggests correct spellings for the word or the first word in the specified range.</td>
</tr>
<tr>
<td>spelling word type wildcard</td>
<td>Word suggests replacements that match the search criteria for a word that contains the question mark (?) or asterisk (*) wildcard character. This constant is not supported in the current spell checker.</td>
</tr>
<tr>
<td>spelling word type anagram</td>
<td>Word suggests anagrams for the word. Word doesn't suggest anagrams from a custom dictionary. This constant is not supported in the current spell checker.</td>
</tr>
</tbody>
</table>

Note  This parameter may be ignored, depending on the dictionary file currently in use.

```
  [custom dictionary2-custom dictionary10  dictionary]  Optional. Either an expression that returns a dictionary object or the file name of an additional custom dictionary. You can specify as many as nine additional dictionaries.
```
Example

This example looks for alternate spelling suggestions for the first word in the selection. If there are suggestions, the example runs a spelling check on the selection.

set spellSug to list of (text range spelling suggestions text object of ¬
             selection)
if (count of spellSug) = 0 then
    display dialog "No suggestions."
else
    check spelling text object of selection
end if
Proofing Suite

Classes

autocorrect..................................................................................................................486
autocorrect entry ........................................................................................................486
dictionary ....................................................................................................................488
first letter exception .................................................................................................491
language ......................................................................................................................491
other corrections exception .......................................................................................495
readability statistic .....................................................................................................495
synonym info .................................................................................................................496
two initial caps exception .........................................................................................498

autocorrect

Plural

autocorrections

Elements

autocorrect entry

first letter exception

two initial caps exception

other corrections exception

Represents the AutoCorrect functionality in Word.

Use the autocorrect object property of the application class to return the autocorrect object. The following example enables the AutoCorrect options and creates an AutoCorrect entry.

set ac to autocorrect object
set two initial caps auto add of ac to true
set correct days of ac to true
set newAuto to make new autocorrect entry with properties ¬
{name:"usually", autocorrect value:"usually"} at ac

The autocorrect entry element represents the AutoCorrect entries in the AutoCorrect dialog box (Tools menu).
Proofing Suite

Properties

correct days

True if Word automatically capitalizes the first letter of days of the week. Read/write.

correct initial caps

True if Word automatically makes the second letter lowercase if the first two letters of a word are typed in uppercase. For example, "WOrd" is corrected to "Word." Read/write.

correct sentence caps

True if Word automatically capitalizes the first letter in each sentence. Read/write.

first letter auto add

True if Word automatically adds abbreviations to the list of AutoCorrect First Letter exceptions. Word adds an abbreviation to this list if you delete and then retype the letter that Word capitalized immediately after the period following the abbreviation. Read/write.

other corrections auto add

True if Word automatically adds words to the list of AutoCorrect exceptions on the Other Corrections tab in the AutoCorrect Exceptions dialog box (AutoCorrect command, Tools menu). Word adds a word to this list if you delete and then retype a word that you didn't want Word to correct. Read/write.

replace text

True if Word automatically replaces specified text with entries from the AutoCorrect list. Read/write.

replace text from spelling checker

True if Word automatically replaces misspelled text with suggestions from the spelling checker as the user types. Word only replaces words that contain a single misspelling and for which the spelling dictionary only lists one alternative. Read/write.

two initial caps auto add

True if Word automatically adds words to the list of AutoCorrect Initial Caps exceptions. A word is added to this list if you delete and then retype the uppercase letter (following the initial uppercase letter) that Word changed to lowercase. Read/write.
**autocorrect entry**

**Plural autocorrect entries**

Represents a single AutoCorrect entry. The **autocorrect entry** class includes the entries in the **AutoCorrect** dialog box (Tools menu).

Use `auto correct entry index` of `autocorrect object`, where `index` is the AutoCorrect entry name or index number, to return a single **autocorrect entry** object. You must exactly match the spelling (but not necessarily the capitalization) of the name as it's shown under Replace in the **AutoCorrect** dialog box. The following example sets the value of the AutoCorrect entry named "teh."

```plaintext
set autocorrect value of autocorrect entry "teh" of autocorrect object to "the"
```

To insert an AutoCorrect entry at the specified range, use the `apply correction` command. The following example adds an AutoCorrect entry and then inserts it in place of the selection.

```plaintext
make new autocorrect entry at autocorrect object with properties ¬
{name:"helpl", autocorrect value:"hello"}
apply correction autocorrect entry "helpl" of autocorrect object to range ¬
text object of selection object of active document
```

Use the standard `make` command to add an AutoCorrect entry to the list of available entries.

The following example adds a plain-text AutoCorrect entry for the misspelling of the word "their."

```plaintext
make new autocorrect entry at autocorrect object with properties ¬
{name:"thier", autocorrect value:"their"}
```

The following example creates an AutoCorrect entry named "PMO" based on the text and formatting of the selection.

```plaintext
make new autocorrect entry at autocorrect object with properties ¬
{text range:text object of selection object of active document, ¬
name:"PMO"}
```

**Properties**

- **autocorrect value**
  Returns or sets the value of the **autocorrect entry**. Read/write.

- **entry index**
  Returns the index number that indicates the position of an item in its container element list. Read-only.

- **name**
  Returns or sets the name of the **autocorrect entry**. Read/write.

- **rich text**
  **True** if formatting is stored with the **autocorrect entry** replacement text. Read-only.
Proofing Suite

**dictionary**

**Plural**

**dictionaries**

Represents a dictionary. Dictionary objects are returned by properties of the `language` class; these include the `active spelling dictionary`, `active grammar dictionary`, `active thesaurus dictionary`, and `active hyphenation dictionary` properties.

Use `dictionary index`, where `index` is an index number or the string name for the dictionary, to return a single `dictionary` object that represents a custom dictionary. The following example returns the first dictionary in Word.

```
dictionary 1
```

The following example returns the dictionary named "MyDictionary."

```
dictionary "MyDictionary"
```

Use the `language specific` property to determine whether the specified custom dictionary can have a specific language assigned to it with the `language ID` property. If the dictionary is language specific, it will verify only text that is formatted for the specified language.

For each language for which proofing tools are installed, you can use the `active grammar dictionary`, `active hyphenation dictionary`, `active spelling dictionary`, and `active thesaurus dictionary` properties of the `language` class to return the corresponding `dictionary` objects. The following example returns the full path for the active spelling dictionary used in the U.S. English version of Word.

```
set mySpell to the active spelling dictionary of (retrieve language english us)
display dialog (path of mySpell) & (path separator of application ¬
                      "Microsoft Word") & (name of mySpell)
```

The `read only` property returns `true` for built-in proofing dictionaries and `false` for custom spelling dictionaries.

**Properties**

**dictionary type**

Returns the dictionary type. Read-only.

Can be one of the following:

- `grammar`
- `hyphenation`
- `spelling`
- `spelling complete`
- `spelling custom`
- `spelling legal`
- `spelling medical`
- `hangul hanja conversion`
- `custom`
- `thesaurus`
Proofing Suite

language id

Returns or sets the language for the specified object. Read/write.

Can be one of the following:

<table>
<thead>
<tr>
<th>language</th>
<th>language</th>
<th>language</th>
</tr>
</thead>
<tbody>
<tr>
<td>afrikaans</td>
<td>french</td>
<td>russian</td>
</tr>
<tr>
<td>arabic</td>
<td>french canadian</td>
<td>serbian cyrillic</td>
</tr>
<tr>
<td>basque</td>
<td>german</td>
<td>serbian latin</td>
</tr>
<tr>
<td>belgian dutch</td>
<td>greek</td>
<td>sesotho</td>
</tr>
<tr>
<td>belgian french</td>
<td>hebrew</td>
<td>simplified chinese</td>
</tr>
<tr>
<td>brazilian portuguese</td>
<td>hungarian</td>
<td>slovak</td>
</tr>
<tr>
<td>bulgarian</td>
<td>italian</td>
<td>slovenian</td>
</tr>
<tr>
<td>byelorussian</td>
<td>icelandic</td>
<td>spanish</td>
</tr>
<tr>
<td>catalan</td>
<td>japanese</td>
<td>spanish modern sort</td>
</tr>
<tr>
<td>croatian</td>
<td>korean</td>
<td>swedish</td>
</tr>
<tr>
<td>czech</td>
<td>language none</td>
<td>swiss french</td>
</tr>
<tr>
<td>danish</td>
<td>latvian</td>
<td>swiss german</td>
</tr>
<tr>
<td>dutch</td>
<td>macedonian</td>
<td>swiss italian</td>
</tr>
<tr>
<td>english aus</td>
<td>malaysian</td>
<td>traditional chinese</td>
</tr>
<tr>
<td>english canadian</td>
<td>mexicanSpanish</td>
<td>tswana</td>
</tr>
<tr>
<td>english new zealand</td>
<td>language no proofing</td>
<td>tswana</td>
</tr>
<tr>
<td>english south africa</td>
<td>norwegian bokmol</td>
<td>turkish</td>
</tr>
<tr>
<td>english uk</td>
<td>norwegian nynorsk</td>
<td>ukrainian</td>
</tr>
<tr>
<td>english us</td>
<td>polish</td>
<td>venda</td>
</tr>
<tr>
<td>estonian</td>
<td>portuguese</td>
<td>xhosa</td>
</tr>
<tr>
<td>farsi</td>
<td>romanian</td>
<td>zulu</td>
</tr>
</tbody>
</table>

language specific

True if the custom dictionary is to be used only with text formatted for a specific language. Read/write.

name

Returns the dictionary's file name. Read-only.

path

Returns the path to the specified dictionary object. Read-only.

Note The path does not include a trailing character; for example, Macintosh HD:Users:Shared. To add the character that separates folders and drive letters, use the path separator property of the application class. Use the name property to return the file name without the path.

read only

True if the specified dictionary cannot be changed. Read-only.
first letter exception

Plural
first letter exceptions

Represents an abbreviation excluded from automatic correction.

Note The first character following a period is automatically capitalized when the correct sentence caps property of the autocorrect class is set to true. The character you type following a first letter exception item is not capitalized.

Use first letter exception index, where index is the abbreviation or the index number, to return a first letter exception object. The following example deletes the abbreviation "appt." from the First Letter list in the AutoCorrect Exceptions dialog box.

delete first letter exception "appt." of autocorrect object

The following example displays the name of the first item in the First Letter list in the AutoCorrect Exceptions dialog box.

display dialog (get name of first letter exception 1 of autocorrect object)

To add an abbreviation to the list of first-letter exceptions, use the standard make command. The following example adds the abbreviation "addr." to this list.

make new first letter exception at autocorrect object ¬
   with properties {name:"appt."}

Properties
entry index
   Returns the index number that indicates the position of an item in its container element list.
   Read-only.

name
   Returns the word that is excepted from AutoCorrect actions. Read-only.

language

Plural
languages

Represents a language used for proofing or formatting in Word.

To return a single language object, use the retrieve language command.

The name property returns the name of a language, whereas the name local property returns the name of a language in the language of the user. The following example returns the string "Italiano" for name and "Italian" for name local when it's run in the U.S. English version of Word.

display dialog (get name of (retrieve language italian))
display dialog (get name local of (retrieve language italian))
Proofing Suite

For each language for which proofing tools are installed, you can use the *active grammar dictionary*, *active hyphenation dictionary*, *active spelling dictionary*, and *active thesaurus dictionary* properties of the *language* class to return the corresponding *dictionary* objects. The following example returns the full path for the active spelling dictionary used in the U.S. English version of Word.

```applescript
set mySpell to the active spelling dictionary of ¬
   (retrieve language english us)
display dialog (path of mySpell) & (path separator of application ¬
   "Microsoft Word") & (name of mySpell)
```

**Setting the writing style**

The writing style is the set of rules used by the grammar checker. The *writing style list* property returns a list of strings that represent the available writing styles for the specified language. The following example returns the list of writing styles for U.S. English.

```applescript
set langUS to retrieve language english us
set WrStyles to get writing style list of langUS
Use the *default writing style* property to set the default writing style you want Word to use.
set langUS to retrieve language english us
set default writing style of langUS to "Casual"
```

You can override the default writing style by using the *set active writing style* command. This command is applied to a specified document for text marked in a specified language. The following example sets the writing style to be used for checking U.S. English, French, and German for the active document.

```applescript
set active writing style active document language ID english us ¬
   writing style "Technical"
set active writing style active document language ID french ¬
   writing style "Commercial"
set active writing style active document language ID german ¬
   writing style "Grammatik"
```

**Remarks**

You must have the proofing tools installed for each language you intend to check.

If you mark text as *no proofing*, Word skips the marked text when running a spelling or grammar check. To mark text for a specified language or for no proofing, use the *Set Language* command (*Tools* menu, *Language* submenu).
Proofing Suite

Properties

**active grammar dictionary**

Returns a dictionary object that represents the active grammar dictionary for the specified language. Read-only.

If no grammar dictionary is installed for the specified language, this property returns nothing.

**active hyphenation dictionary**

Returns a dictionary object that represents the active hyphenation dictionary for the specified language. Read-only.

If no hyphenation dictionary is installed for the specified language, this property returns nothing.

**active spelling dictionary**

Returns a dictionary object that represents the active spelling dictionary for the specified language. Read-only.

If no spelling dictionary is installed for the specified language, this property returns nothing.

**active thesaurus dictionary**

Returns a dictionary object that represents the active thesaurus dictionary for the specified language. Read-only.

If no thesaurus dictionary is installed for the specified language, this property returns nothing.

**default writing style**

Returns or sets the default writing style used by the grammar checker for the specified language. The name of the writing style is the localized name for the specified language. Read/write.

This property controls the global setting for the writing style. When setting this property, you must use the exact name found in the Writing style box in the Spelling & Grammar pane of the Preferences dialog box (Word menu).

The set active writing style command sets the writing style for each language in a document. Setting the active writing style overrides the default writing style setting.
Proofing Suite

**language id**

Returns or sets the language for the specified object. Read/write.

Can be one of the following:

- afrikaans
- arabaic
- basque
- belgian dutch
- belgian french
- brazilian portuguese
- bulgarian
- byelorussian
- catalan
- croatian
- czech
- danish
- dutch
- english aus
- english canadian
- english new zealand
- english south africa
- english uk
- english us
- estonian
- farsi
- finnish
- french
- french canadian
- german
- greek
- hebrew
- hungarian
- italian
- icelandic
- japanese
- korean
- language none
- latvian
- macedonian
- malaysian
- mexican spanish
- language no proofing
- norwegian bokmål
- norwegian nynorsk
- polish
- portuguese
- romanian
- russian
- serbian cyrillic
- serbian latin
- sesotho
- simplified chinese
- slovak
- slovenian
- spanish
- spanish modern sort
- swedish
- swiss french
- swiss german
- swiss italian
- traditional chinese
- tswana
- tsonga
- turkish
- ukrainian
- venda
- xhosa
- zulu

**name**

Returns the name of the proofing tools language. Read-only.

**name local**

Returns the name of a proofing tool language in the language of the user. Read-only.

**spelling dictionary type**

Returns or sets the proofing tool type. Read/write.

Can be one of the following:

- grammar
- hangul hanja conversion
- custom
- hyphenation
- spelling
- spelling complete
- spelling custom
- spelling legal
- spelling medical
- thesaurus

You can use this property to change the active spelling dictionary to one of the available add-on dictionaries that work with Word. For example, there are legal, medical, and complete spelling dictionaries you can use instead of the standard dictionary.

**writing style list**

Returns a string array that contains the names of all writing styles available for the specified language. Read-only.
other corrections exception

Plural
other corrections exceptions

Represents a single AutoCorrect exception. The other corrections exception class includes all words that Word won't correct automatically. This list corresponds to the list of AutoCorrect exceptions on the Other Corrections tab in the AutoCorrect Exceptions dialog box (AutoCorrect command, Tools menu).

To return an other corrections exception object, use other corrections exception index, where index is the AutoCorrect exception name or the index number. The following example deletes "WTop" from the list of AutoCorrect exceptions.

delete other corrections exception "WTop" of autocorrect object

The index number represents the position of the AutoCorrect exception in the list of AutoCorrect exceptions on the Other Corrections tab in the AutoCorrect Exceptions dialog box. The following example displays the name of the first item in the list.

display dialog (get name of other corrections exception 1 of autocorrect object)

If the value of the other corrections auto add property of the autocorrect class is true, words are automatically added to the list of AutoCorrect exceptions. Use the standard make command to add an item to the other corrections exception list. The following example adds "TipTop" to the list of AutoCorrect exceptions.

make new other corrections exception of autocorrect object with properties ¬
   { name: "TipTop" }

Properties
entry index
   Returns the index number that indicates the position of an item in its container element list. Read-only.

name
   Returns the word that is excepted from AutoCorrect actions. Read-only.

readability statistic

Plural
readability statistics

Represents one of the readability statistics for a document or range.

Use readability statistic index, where index is the index number, to return a single readability statistic object. The statistics are ordered as follows: Words, Characters, Paragraphs, Sentences, Sentences per Paragraph, Words per Sentence, Characters per Word, Passive Sentences, Flesch Reading Ease, and Flesch-Kincaid Grade Level. The following example returns the character count for the active document.

display dialog (get readability value of readability statistic 2 of ¬
   text object of active document)
**Proofing Suite**

**Properties**

*name*

Returns the name of the readability statistic. Read-only.

*readability value*

Returns the value of the grammar statistic. Read-only.

**synonym info**

**Plural**

**synonym info**

Plural synonym info objects represent the information about synonyms, antonyms, related words, or related expressions for the specified range or a given string.

To return a synonym info object, use the get synonym info object command. The synonym info object can be returned either from a word or a phrase. You specify the lookup word or phrase and a proofing language ID. The following example returns a synonym info object from the current selection.

```plaintext
get synonym info object item to check content of selection ¬
language ID english us
```

The found property returns true if any information is found in the thesaurus for the specified word or phrase. Note, however, that this property returns true not only if synonyms are found but also if related words, related expressions, or antonyms are found.

Many of the properties of the synonym info object return a list that contains an array of strings. When working with these properties, you can assign the returned list to a variable and then index the variable to see the elements in the list. In the following example, sList is assigned to the synonym list for the first meaning of the word in the current selection. The Repeat with...In function steps through the list, and then each element is displayed in a message box.

```plaintext
set synInfo to get synonym info object item to check content of selection ¬
language id english us
set sList to get synonym list from synInfo meaning index 1
repeat with i in sList
    display dialog i
end repeat
```
You can check the value of the `meaning count` property to prevent potential errors in your code. The following example returns a list of synonyms for the second meaning for the word or phrase in the selection and displays these synonyms in a message box.

```lisp
set synInfo to get synonym info object item to check content of selection
    language id english us
if meaning count of synInfo ≥ 2 then
    set synList to get synonym list from synInfo meaning index 2
    repeat with i in synList
        display dialog i
    end repeat
else
    display dialog "There is no second meaning for the selection."
end if
```

**Properties**

- **antonyms**
  Returns a list of antonyms for the word or phrase. The list is returned as an array of strings. Read-only.

- **context**
  Returns the word or phrase that was looked up in the thesaurus. Read-only.

  The thesaurus will sometimes look up a shortened version of the string or range used to return the `synonym info` object. The `word` property allows you to see the exact string that was used.

- **found**
  True if the thesaurus finds synonyms, antonyms, related words, or related expressions for the word or phrase. Read-only.

- **meaning count**
  Returns the number of entries in the list of meanings found in the thesaurus for the word or phrase. Returns 0 (zero) if no meanings were found. Read-only.

  Each meaning represents a unique list of synonyms for the word or phrase. The lists of related words, related expressions, and antonyms aren’t counted as entries in the list of meanings.

- **meanings**
  Returns the list of meanings for the word or phrase. The list is returned as an array of strings. Read-only.

  The lists of related words, related expressions, and antonyms aren’t counted as entries in the list of meanings.
Proofing Suite

part of speech

Returns a list of the parts of speech corresponding to the meanings found for the word or phrase looked up in the thesaurus. The list is returned as an array of integers. Read-only.

The part of speech is returned as a list consisting of the following: adjective, adverb, noun, and verb. The array elements are ordered to correspond to the elements returned by the meanings property.

related expressions

Returns a list of expressions related to the specified word or phrase. The list is returned as an array of strings. Read-only.

Typically, there are very few related expressions found in the thesaurus.

related words

Returns a list of words related to the specified word or phrase. The list is returned as an array of strings. Read-only.

two initial caps exception

Plural

two initial caps exceptions

Represents a single initial-capital AutoCorrect exception. The two initial caps exception class includes all the items listed in the Don't correct box on the INitial CAps tab in the AutoCorrect Exceptions dialog box.

To return a two initial caps exception object, use two initial caps exception index, where index is the initial capital exception name or the index number. The following example deletes the initial-capital exception named "KMenu."

delete two initial caps exception "KMenu" of autocorrect object

The index number represents the position of the initial-capital exception in the two initial caps exception list. The last exception added to this list is index number 1. The following example displays the name of the first item in the two initial caps exception list.

display dialog (get name of two initial caps exception 1 of autocorrect object)

If the two initial caps auto add property of the autocorrect class is true, words are automatically added to the list of initial-capital exceptions. Use the standard make command to add an item to the two initial caps exception list. The following example adds "Industry" to the list of initial-capital exceptions.

make new two initial caps exception at autocorrect object with properties ¬
    {name: "Industry"}

Properties

entry index

Returns the index number that indicates the position of an item in its container element list. Read-only.

name

Returns the word that is excepted from AutoCorrect actions. Read-only.
**Commands**

apply correction............................................................................................................499  
get synonym list for ...........................................................................................................499  
get synonym list from........................................................................................................500

**Command: apply correction**

Replaces a range with the value of the specified AutoCorrect entry.

**Syntax**

apply correction  autocorrect entry  Required. An expression that returns an autocorrect entry object.
  to range  text range  Required. The textrange object that's replaced by the AutoCorrect entry.

**Example**

This example adds an AutoCorrect replacement entry, then applies the "sr" AutoCorrect entry to the selected text.

make new autocorrect entry at autocorrect object with properties ¬  
  {name:"sr", autocorrect value:"Stuart Railson"}

apply correction autocorrect entry "sr" of autocorrect object to range ¬  
  text object of selection of active document

This example applies the "sr" AutoCorrect entry to the first word in the active document.

apply correction autocorrect entry "sr" of autocorrect object to range ¬  
  word 1 of active document

**Command: get synonym list for**

Returns a list of synonyms for a specified meaning of a word or phrase. The list is returned as an array of strings. Read-only.

**Syntax**

get synonym list for  synonym info  Required. An expression that returns a synonym info object.
  item to check  Unicode text  Required. The meaning as a string.

**Example**

This example returns a list of synonyms for the word "big," using the meaning 'considerable' in U.S. English.

set synInfo to get synonym info object item to check "big" ¬  
  language id english us

set sList to get synonym list for synInfo item to check "considerable"

repeat with i in sList
  display dialog i
end repeat
Proofing Suite

**Command: get synonym list from**

Returns a list of synonyms for a specified meaning of a word or phrase. The list is returned as an array of strings. Read-only.

**Syntax**

get synonym list from synonym info  Required. An expression that returns a synonym info object.

  meaning index  integer  Required. The meaning as an index number in the list of possible meanings.

**Example**

This example returns a list of synonyms for the second meaning of the selected word or phrase and displays these synonyms in a message box. If there's no second meaning or if there are no synonyms, this is stated in a message box.

```
set synInfo to get synonym info object item to check content of selection ¬
    language id english us
if meaning count of synInfo ≥ 2 then
  set synList to get synonym list from synInfo meaning index 2
  repeat with i in synList
    display dialog i
  end repeat
else
  display dialog "There is no second meaning for this word or phrase."
end if
```
Table Suite

Classes

cell ..........................................................501
column ..................................................504
column options ...........................................505
row ..........................................................506
row options .............................................508
table ..........................................................511

Class: cell

Plural
cells

Elements
table

Represents a single table cell.
To return a cell object, use cell column number of row row number, where column number is the column number and row number is the row number of the cell.
The following example applies shading to the second cell in the first row.
set myCell to cell 2 of row 1 of table 1 of active document
set texture of shading of myCell to texture20 percent
The following example applies shading to the first cell in the first row.
set texture of shading of cell 1 of row 1 of table 1 of active document ¬
to texture20 percent
To insert new cells, use the insert cells command.
To add a row or column of cells, use the insert rows or the insert columns command.

Properties

border options

Returns a border options object that represents all the borders for the specified object. Read-only.
bottom padding

Returns or sets the amount of space (in points) to add below the contents of a single cell or all the cells in a table. Read/write.

The setting of the bottom padding property for a single cell overrides the setting of the bottom padding property for the entire table.

column

Returns a column object that represents the table column containing the specified cell. Read-only.

column index

Returns the number of the table column that contains the specified cell. Read-only.

fit text

True if Word visually reduces the size of text typed into a cell so that it fits within the column width. Read/write.

If the fit text property is set to true, the font size of the text is not changed, but the visual width of the characters is adjusted to fit all of the typed text into the cell.

height

Returns or sets the height of the specified cell or cells in a table. If the height rule property of the specified row is row height auto, height returns 9.999999E+6; setting the height property sets the height rule property to row height at least. Read/write.

height rule

Returns or sets the rule for determining the height of the specified cells or rows. Read/write.

Can be one of the following:

- row height at least
- row height auto
- row height exactly

left padding

Returns or sets the amount of space (in points) to add to the left of the contents of a single cell or all the cells in a table. Read/write.

The setting of the left padding property for a single cell overrides the setting of the left padding property for the entire table.

nesting level

Returns the nesting level of the specified cells, columns, rows, or tables. Read-only.

The outermost table has a nesting level of 1. The nesting level of each successively nested table is one higher than the previous table.

next cell

Returns the next cell object. Read-only.
preferred width

Returns or sets the preferred width (in points or as a percentage of the window width) for the specified cells, columns, or table. Read/write.

If the preferred width type property is set to preferred width points, the preferred width property returns or sets the width in points. If the preferred width type property is set to preferred width percent, the preferred width property returns or sets the width as a percentage of the window width.

preferred width type

Returns or sets the measurement unit to use for the preferred width of the specified cells, columns, or table. Read/write.

Can be one of the following:

- preferred width auto
- preferred width percent
- preferred width points

previous cell

Returns the previous cell object. Read-only.

right padding

Returns or sets the amount of space (in points) to add to the right of the contents of a single cell or all the cells in a table. Read/write.

The setting of the right padding property for a single cell overrides the setting of the right padding property for the entire table.

row

Returns a row object that represents the row containing the specified cell. Read-only.

row index

Returns the number of the row that contains the specified cell. Read-only.

shading

Returns a shading object that refers to the shading formatting for the specified object. Read-only.

text object

Returns a text range object that represents the text in the specified text frame. Read-only.

top padding

Returns or sets the amount of space (in points) to add above the contents of a single cell or all the cells in a table. Read/write.

The setting of the top padding property for a single cell overrides the setting of the top padding property for the entire table.
vertical alignment

Returns or sets the vertical alignment of text on each page in a document or section, or in one or more cells of a table. Read/write.

Can be one of the following:

- cell align vertical bottom
- cell align vertical center
- cell align vertical top

width

Returns or sets the width (in points) of the specified object. Read/write.

word wrap

True if Microsoft Word wraps text to multiple lines and lengthens the cell so that the cell width remains the same. Read/write.

Class: column

Plural: columns

Represents a single table column.

Use column index, where index is the index number, to return a single column object. The index number represents the position of the column in a table (counting from left to right).

The following example selects column one in the first table in the active document.

select (column 1 of table 1 of active document)

The following example deletes the text in cell one, inserts new text, and then sorts the entire column.

delete content of text object of (cell 1 of row 1 of table 1 of active document)

set content of text object of (cell 1 of row 1 of table 1 of active document) to "Sales"

sort ascending (column 1 of table 1 of active document)

To add a column to a table, use the insert columns command.

Properties

border options

Returns a border options object that represents all the borders for the specified object. Read-only

column index

Returns the number of the table column that contains the specified cell. Read-only.

is first

True if the specified column or row is the first one in the table. Read-only.
Table Suite

**is last**

True if the specified column or row is the last one in the table. Read-only.

**nesting level**

Returns the nesting level of the specified cells, columns, rows, or tables. Read-only.

The outermost table has a nesting level of 1. The nesting level of each successively nested table is one higher than the previous table.

**next column**

Returns the next column in the table. Read-only.

**preferred width**

Returns or sets the preferred width (in points or as a percentage of the window width) for the specified cells, columns, or table. Read/write.

If the preferred width type property is set to preferred width points, the preferred width property returns or sets the width in points. If the preferred width type property is set to preferred width percent, the preferred width property returns or sets the width as a percentage of the window width.

**preferred width type**

Returns or sets the measurement unit to use for the preferred width of the specified cells, columns, or table. Read/write.

Can be one of the following:

- preferred width auto
- preferred width percent
- preferred width points

**previous column**

Returns the previous column. Read-only.

**shading**

Returns a shading object that refers to the shading formatting for the specified object. Read-only.

**width**

Returns or sets the width (in points) of the specified object. Read/write.

**Class: column options**

Represents options that can be set for columns.

To return a column options object, use the column options property of the table class.

This example adjusts the width of the columns in the first table in the active document so that they're equal.

distribute width column options of table 1 of active document
Properties

border options

Returns a border options object that represents all the borders for the specified object. Read-only.

default width

Returns or sets the default width (in points) of columns in the specified table. Read/write.

preferred width

Returns or sets the preferred width (in points or as a percentage of the window width) for the specified cells, columns, or table. Read/write.

If the preferred width type property is set to preferred width points, the preferred width property returns or sets the width in points. If the preferred width type property is set to preferred width percent, the preferred width property returns or sets the width as a percentage of the window width.

preferred width type

Returns or sets the measurement unit to use for the preferred width of the specified cells, columns, or table. Read/write.

Can be one of the following:

- preferred width auto
- preferred width percent
- preferred width points

shading

Returns a shading object that refers to the shading formatting for the specified object. Read-only.

Class: row

Plural

rows

Elements

cell

Represents a row in a table.

Use row index, where index is the index number, to return a single row object. The index number represents the position of the row in the selection, range, or table. The following example deletes the first row in the first table in the active document.

delete row 1 of table 1 of active document

To add a row to a table, use the insert rows command.
Table Suite

**Properties**

*alignment*

Returns or sets the alignment for the specified rows. Read/write.

Can be one of the following:

- **align row left**
- **align row center**
- **align row right**

*allow break across pages*

**True** if the text in a table row or rows is allowed to split across a page break. Read/write.

*border options*

Returns a **border options** object that represents all of the borders for the specified object. Read-only.

*heading format*

**True** if the specified row or rows are formatted as a table heading. Rows formatted as table headings are repeated when a table spans more than one page. Read/write.

*height*

Returns or sets the height of the specified row or rows in a table. If the `rule height` property of the specified row is **row height auto**, `height` returns 9.999999E+6; setting the `height` property sets `rule height` to **row height at least**. Read/write.

*is first*

**True** if the specified column or row is the first one in the table. Read-only.

*is last*

**True** if the specified column or row is the last one in the table. Read-only.

*nesting level*

Returns the nesting level of the specified cells, columns, rows, or tables. Read-only.

The outermost table has a nesting level of 1. The nesting level of each successively nested table is one higher than the previous table.

*next row*

Returns the next row object. Read-only.

*previous row*

Returns the previous row object. Read-only.

*row index*

Returns the number of the row that contains the specified cell. Read-only.

*row left indent*

Returns or sets the distance (in points) between the left page margin and the left edge of the text in the specified rows. Read/write.
Table Suite

rule height
Returns or sets the rule for determining the height of the specified cells or rows. Read/write.
Can be one of the following:
- row height at least
- row height auto
- row height exactly

shading
Returns a shading object that refers to the shading formatting for the specified object. Read-only.

space between columns
Returns or sets the distance (in points) between text in adjacent columns of the specified row or rows. Read/write.

text object
Returns a text range object that represents the text in the specified row object. Read-only.

Class: row options
Represents options that can be set for rows.
To return a row options object, use the row options property of the table class.
This example adjusts the height of the rows in the first table in the active document so that they’re equal.

distribute row height row options of table 1 of active document

Properties
alignment
Returns or sets the alignment for the specified rows. Read/write.
Can be one of the following:
- align row left
- align row center
- align row right

allow break across pages
True if Microsoft Word allows the specified table to break across pages. Read/write.

allow overlap
Returns undefined if the specified rows include both overlapping rows and nonoverlapping rows. Can be set to either True or False. Read/write. Setting allow overlap to True also sets wrap around text to True, and setting wrap around text to False also sets allow overlap to False.
border options

Returns a border options object that represents all the borders for the specified object. Read-only.

distance bottom

Returns or sets the distance (in points) between the document text and the bottom edge of the text-free area surrounding the specified shape. The size and shape of the specified shape, together with the values of the wrap type and wrap side properties of the wrap format object, determine the size and shape of this text-free area. Read/write.

distance left

Returns or sets the distance (in points) between the document text and the left edge of the text-free area surrounding the specified shape. The size and shape of the specified shape, together with the values of the wrap type and wrap side properties of the wrap format object, determine the size and shape of this text-free area. Read/write.

distance right

Returns or sets the distance (in points) between the document text and the right edge of the text-free area surrounding the specified shape. The size and shape of the specified shape, together with the values of the wrap type and wrap side properties of the wrap format object, determine the size and shape of this text-free area. Read/write.

distance top

Returns or sets the distance (in points) between the document text and the top edge of the text-free area surrounding the specified shape. The size and shape of the specified shape, together with the values of the wrap type and wrap side properties of the wrap format object, determine the size and shape of this text-free area. Read/write.

heading format

True if the specified row or rows are formatted as a table heading. Rows formatted as table headings are repeated when a table spans more than one page. Read/write.

height

Returns or sets the height of the specified row or rows in a table. If the rule height property of the specified row is row height auto, height returns 9.999999E+6; setting the height property sets rule height to row height at least. Read/write.

horizontal position

Returns or sets the horizontal distance between the edge of the frame and the item specified by the relative horizontal position property. Read/write.

Can be a number that indicates a measurement in points, or can be one of the following:

- left
- right
- center
- inside
- outside
relative horizontal position

Specifies what the horizontal position of a frame or shape is relative to. Read/write.

Can be one of the following:

- relative horizontal position column
- relative horizontal position margin
- relative horizontal position page

relative vertical position

Returns or sets the distance (in points) between the left page margin and the left edge of the text in the specified rows. Read/write.

rule height

Returns or sets the rule for determining the height of the specified cells or rows. Read/write.

Can be one of the following:

- row height at least
- row height auto
- row height exactly

shading

Returns a shading object that refers to the shading formatting for the specified object. Read-only.

space between columns

Returns or sets the distance (in points) between text in adjacent columns of the specified row or rows. Read/write.

vertical position

Returns or sets the vertical distance between the edge of the frame and the item specified by the relative vertical position property. Read/write.

Can be a number that indicates a measurement in points, or can be one of the following:

- top
- bottom
- center

wrap around text

Returns or sets whether text should wrap around the specified rows. Returns true if only some of the specified rows have wrapping enabled. Read/write.

Setting the wrap around text property to false also sets the allow overlap property to false. Setting the allow overlap property to true also sets the wrap around text property to true.
**Class:** table

**Plural**

tables

**Elements**

column

row

table

Represents a single table.

Use `table index`, where `index` is the index number, to return a single table object. The index number represents the position of the table in the selection, text range, or document. The following example converts the first table in the active document to text.

```plaintext
set table1 to table 1 of active document
convert to text table1 separator separate by tabs
```

**Properties**

**allow auto fit**

True if Microsoft Word automatically resizes cells in a table to fit their contents. Read/write.

**allow page breaks**

True if Microsoft Word allows the specified table to break across pages. Read/write.

**auto format type**

Returns the type of automatic formatting that's been applied to the specified table. Read-only.

Can be one of the following:

- table format 3D effects1
- table format 3D effects2
- table format 3D effects3
- table format classic1
- table format classic2
- table format classic3
- table format classic4
- table format colorful1
- table format colorful2
- table format colorful3
- table format columns1
- table format columns2
- table format columns3
- table format columns4
- table format columns5
- table format columns6
- table format columns7
- table format columns8
- table format grid1
- table format grid2
- table format grid3
- table format grid4
- table format grid5
- table format grid6
- table format grid7
- table format grid8
- table format grid9
- table format grid10
- table format grid11
- table format grid12
- table format grid13
- table format grid14
- table format grid15
- table format grid16
- table format grid17
- table format grid18
- table format grid19
- table format grid20
- table format grid21
- table format grid22
- table format grid23
- table format grid24
- table format grid25
- table format grid26
- table format grid27
- table format grid28
- table format grid29
- table format grid30
- table format grid31
- table format grid32
- table format grid33
- table format grid34
- table format grid35
- table format grid36
- table format grid37
- table format grid38
- table format grid39
- table format grid40
- table format grid41
- table format grid42
- table format grid43
- table format grid44
- table format grid45
- table format grid46
- table format grid47
- table format grid48
- table format grid49
- table format grid50
- table format grid51
- table format grid52
- table format grid53
- table format grid54
- table format grid55
- table format grid56
- table format grid57
- table format grid58
- table format grid59
- table format grid60
- table format grid61
- table format grid62
- table format grid63
- table format grid64
- table format grid65
- table format grid66
- table format grid67
- table format grid68
- table format grid69
- table format grid70
- table format grid71
- table format grid72
- table format grid73
- table format grid74
- table format grid75
- table format grid76
- table format grid77
- table format grid78
- table format grid79
- table format grid80
- table format grid81
- table format grid82
- table format grid83
- table format grid84
- table format grid85
- table format grid86
- table format grid87
- table format grid88
- table format grid89
- table format grid90
- table format grid91
- table format grid92
- table format grid93
- table format grid94
- table format grid95
- table format grid96
- table format grid97
- table format grid98
- table format grid99
- table format grid100

**Note** To apply automatic formatting to a table, use the `auto format table` command.

**border options**

Returns a border options object that represents all of the borders for the specified object. Read-only.
Table Suite

**bottom padding**

Returns or sets the amount of space (in points) to add below the contents of a single cell or all the cells in a table. Read/write.

The setting of the *bottom padding* property for a single cell overrides the setting of the *bottom padding* property for the entire table.

**column options**

Returns the [column options](#) object associated with the table object. Read-only.

**left padding**

Returns or sets the amount of space (in points) to add to the left of the contents of a single cell or all the cells in a table. Read/write.

The setting of the *left padding* property for a single cell overrides the setting of the *left padding* property for the entire table.

**nesting level**

Returns the nesting level of the specified cells, columns, rows, or tables. Read-only.

The outermost table has a nesting level of 1. The nesting level of each successively nested table is one higher than the previous table.

**number of columns**

Returns the number of columns in the associated table object. Read-only.

**number of rows**

Returns the number of rows in the associated table object. Read-only.

**preferred width**

Returns or sets the preferred width (in points or as a percentage of the window width) for the specified cells, columns, or table. Read/write.

If the *preferred width type* property is set to *preferred width points*, the *preferred width* property returns or sets the width in points. If the *preferred width type* property is set to *preferred width percent*, the *preferred width* property returns or sets the width as a percentage of the window width.

**preferred width type**

Returns or sets the measurement unit to use for the preferred width of the specified cells, columns, or table. Read/write.

Can be one of the following:

- preferred width auto
- preferred width percent
- preferred width points
Table Suite

right padding

Returns or sets the amount of space (in points) to add to the right of the contents of a single cell or all the cells in a table. Read/write.

The setting of the right padding property for a single cell overrides the setting of the right padding property for the entire table.

row options

Returns the row options object associated with the table object. Read-only.

shading

Returns a shading object that refers to the shading formatting for the specified object. Read-only.

spacing

Returns or sets the spacing (in points) between the columns in the table. Read/write.

text object

Returns a text range object that represents the text in the specified text frame. Read-only.

top padding

Returns or sets the amount of space (in points) to add above the contents of a single cell or all the cells in a table. Read/write.

The setting of the top padding property for a single cell overrides the setting of the top padding property for the entire table.

uniform

True if all the rows in a table have the same number of columns. Read-only.
Commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>auto fit</td>
<td>514</td>
</tr>
<tr>
<td>auto fit behavior</td>
<td>515</td>
</tr>
<tr>
<td>auto format table</td>
<td>515</td>
</tr>
<tr>
<td>auto sum</td>
<td>516</td>
</tr>
<tr>
<td>convert row to text</td>
<td>517</td>
</tr>
<tr>
<td>convert to text</td>
<td>517</td>
</tr>
<tr>
<td>distribute row height</td>
<td>518</td>
</tr>
<tr>
<td>distribute width</td>
<td>518</td>
</tr>
<tr>
<td>formula</td>
<td>518</td>
</tr>
<tr>
<td>get border</td>
<td>519</td>
</tr>
<tr>
<td>get cell from table</td>
<td>520</td>
</tr>
<tr>
<td>merge cell</td>
<td>520</td>
</tr>
<tr>
<td>set left indent</td>
<td>521</td>
</tr>
<tr>
<td>set table item height</td>
<td>521</td>
</tr>
<tr>
<td>set table item width</td>
<td>522</td>
</tr>
<tr>
<td>sort ascending</td>
<td>523</td>
</tr>
<tr>
<td>sort descending</td>
<td>523</td>
</tr>
<tr>
<td>split cell</td>
<td>524</td>
</tr>
<tr>
<td>split table</td>
<td>524</td>
</tr>
<tr>
<td>table sort</td>
<td>524</td>
</tr>
<tr>
<td>update auto format</td>
<td>526</td>
</tr>
</tbody>
</table>

**Command: auto fit**

Changes the width of a table column to accommodate the width of the text without changing the way text wraps in the cells.

**Syntax**

`auto fit column/column options`  
Required. An expression that returns a `column` or `column options` object.

If the table is already as wide as the distance between the left and right margins, this command has no effect.

**Example**

This example creates a 3x3 table in a new document and then changes the width of the first column to accommodate the width of the text.

set newDoc to make new document

set myTable to make new table at the beginning of newDoc with properties ¬

{number of rows:3, number of columns:3}

set content of text object of cell 1 of row 1 of myTable to "First cell"

auto fit column 1 of myTable
This example creates a 3x3 table in a new document and then changes the width of all the columns to accommodate the width of the text.

```plaintext
set newDoc to make new document
set myTable to make new table at the beginning of newDoc with properties ¬
    {number of rows:3, number of columns:3}
set content of text object of cell 1 of row 1 of myTable to "First cell"
set content of text object of cell 2 of row 1 of myTable to "This is cell (1,2)"
set content of text object of cell 3 of row 1 of myTable to "(1,3)"
auto fit every column of myTable
```

**Command: auto fit behavior**

Determines how Word resizes a table when the AutoFit feature is used. Word can resize the table based on the content of the table cells or the width of the document window. You can also use this command to turn off AutoFit so that the table size is fixed, regardless of cell contents or window width.

**Syntax**

```plaintext
auto fit behavior  table   Required. An expression that returns a table object.
     behavior  enumeration   Required. How Word resizes the specified table when the AutoFit
                      feature is used. Can be one of the following: auto fit content, auto fit fixed, or auto fit
                      window.
```

Setting the `behavior` to `auto fit content` or `auto fit window` sets the `allow auto fit` property to `true` if it's currently `false`. Likewise, setting the `behavior` to `auto fit fixed` sets the `allow auto fit` property to `false` if it's currently `true`.

**Example**

This example sets the AutoFit behavior for the first table in the active document to automatically resize based on the width of the document window.

```plaintext
set myTable to table 1 of active document
auto fit behavior myTable behavior auto fit window
```

**Command: auto format table**

Applies a predefined look to a table. The arguments for this command correspond to the options in the Table AutoFormat dialog box (Table menu).

**Syntax**

```plaintext
auto format table  table   Required. An expression that returns a table object.
                      [table format  enumeration]   Optional. The predefined table format. Can be any one of the auto
                      format table type properties constants. The default value is table format simple1.
                      [apply borders  Boolean]   Optional. True to apply the border properties of the specified format.
                      The default value is true.
                      [apply shading  Boolean]   Optional. True to apply the shading properties of the specified format.
                      The default value is true.
```
Table Suite

[apply font] Boolean  Optional. True to apply the font properties of the specified format. The default value is true.

[apply color] Boolean  Optional. True to apply the color properties of the specified format. The default value is false.

[apply heading rows] Boolean  Optional. True to apply the heading-row properties of the specified format. The default value is true.

[apply last row] Boolean  Optional. True to apply the last-row properties of the specified format. The default value is false.

[apply first column] Boolean  Optional. True to apply the first-column properties of the specified format. The default value is true.

[apply last column] Boolean  Optional. True to apply the last-column properties of the specified format. The default value is false.

[auto fit] Boolean  Optional. True to decrease the width of the table columns as much as possible without changing the way text wraps in the cells. The default value is true.

Example
This example creates a 5x5 table in a new document and applies all the properties of the Colorful 2 format to the table.

set newDoc to make new document
set myTable to make new table at the beginning of newDoc with properties ¬
{number of rows:5, number of columns:5}
auto format table myTable table format table format colorful2 with apply color

Command: auto sum
Inserts an = (Formula) field that calculates and displays the sum of the values in table cells above or to the left of the cell specified in the expression. For information about how Word determines which values to add, see the formula command.

Syntax
auto sum  cell  Required. An expression that returns a cell object.

Example
This example creates a 3x3 table in a new document and sums the numbers in the first column.

set newDoc to make new document
set myTable to make new table at the beginning of newDoc with properties ¬
{number of rows:3, number of columns:3}
set R3C3 to cell 3 of row 3 of myTable
set content of text object of cell 3 of row 1 of myTable to "10"
set content of text object of cell 3 of row 2 of myTable to "15"
auto sum R3C3
Command: convert row to text

Converts a row to text and returns a text range object that represents the delimited text.

Syntax

`convert to text  row/row options`  Required. An expression that returns a row or row options object.

- `[separator  enumeration]`  Optional. The character that delimits the converted columns (paragraph marks delimit the converted rows). Can be one of the following: separate by commas, separate by default list separator, separate by paragraphs, or separate by tabs. The default value is separate by tabs.

- `[nested tables  Boolean]`  Optional. True if nested tables are converted to text. This argument is ignored if separator is not separate by paragraphs. The default value is true.

Example

In the following example, the first row in the active document is converted to text and then formatted as Heading 1.

```plaintext
set myTable to table 1 of active document
set aRange to convert row to text (row 1 of myTable) separator separate by paragraphs
set style of aRange to style heading1
```

Command: convert to text

Converts a table to text and returns a text range object that represents the delimited text.

Syntax

`convert to text  table`  Required. An expression that returns a table object.

- `[separator  enumeration]`  Optional. The character that delimits the converted columns (paragraph marks delimit the converted rows). Can be one of the following: separate by commas, separate by default list separator, separate by paragraphs, or separate by tabs. The default value is separate by tabs.

- `[nested tables  Boolean]`  Optional. True if nested tables are converted to text. This argument is ignored if separator is not separate by paragraphs. The default value is true.

When you apply the convert to text command to a table object, the object is deleted. To maintain a reference to the converted contents of the table, you must assign the text range object returned by the convert to text command to a new object variable.

Example

In the following example, the first table in the active document is converted to text and then formatted as a bulleted list.

```plaintext
set myTable to table 1 of active document
set aRange to convert to text myTable separator separate by paragraphs
apply list format template (list format of aRange) list template
    (list template 2 of (get list gallery bullet gallery))
```
Command: distribute row height
Adjusts the height of the specified rows or cells so that they're equal.

Syntax

\texttt{distribute row height} \ row options \ Required. An expression that returns a \texttt{row options} object.

Example
This example adjusts the height of the rows in the first table in the active document so that they're equal.
\texttt{distribute row height row options of table 1 of active document}

Command: distribute width
Adjusts the width of the specified columns or cells so that they're equal.

Syntax

\texttt{distribute width} \ column options \ Required. An expression that returns a \texttt{column options} object.

Example
This example adjusts the width of the columns in the first table in the active document so that they're equal.
\texttt{distribute width column options of table 1 of active document}

Command: formula
Inserts an = (Formula) field that contains the specified formula into a table cell.

Syntax

\texttt{formula} \ cell \ Required. An expression that returns a \texttt{cell} object.

\texttt{[formula string] \ Unicode text} \ Optional. The mathematical formula you want the = (Formula) field to evaluate. Spreadsheet-type references to table cells are valid. For example, \texttt{"=SUM(A4:C4)"} specifies the first three values in the fourth row.

\texttt{[number format string] \ Unicode text} \ Optional. A format for the result of the = (Formula) field.

Formula string is optional as long as there is at least one cell that contains a value above or to the left of the cell that contains the insertion point. If the cells above the insertion point contain values, the inserted field is \texttt{=} \texttt{SUM(ABOVE)}; if the cells to the left of the insertion point contain values, the inserted field is \texttt{=} \texttt{SUM(LEFT)}. If both the cells above the insertion point and the cells to the left of the insertion point contain values, Word uses the following rules to determine which \texttt{SUM} function to insert:

- If the cell immediately above the insertion point contains a value, Word inserts \texttt{=} \texttt{SUM(ABOVE)}.
- If the cell immediately above the insertion point doesn't contain a value and the cell immediately to the left of it does, Word inserts \texttt{=} \texttt{SUM(LEFT)}.
- If neither adjoining cell contains a value, Word inserts \texttt{=} \texttt{SUM(ABOVE)}.
- If you don't specify \texttt{formula string} and all the cells above and to the left of the insertion point are empty, the result of the field is an error.
Example
This example creates a 3x3 table at the beginning of the active document and then averages the numbers in the last column.

set myTable to make new table at active document with properties ¬
{number of rows: 3, number of columns: 3}
set content of text object of cell 3 of row 1 of myTable to "100"
set content of text object of cell 3 of row 2 of myTable to "50"
formula cell 3 of row 3 of myTable formula string "=Average(Above)"

Command: get border

Returns the specified border object.

Syntax
get border  table/row/column/cell/row options/column options  Required. An expression that returns one of the specified objects.

which border  enumeration  Required.

Can be one of the following:

- border top
- border left
- border bottom
- border right
- border horizontal
- border vertical
- border diagonal down
- border diagonal up

Example
This example applies a double line border to the bottom of the first table in the active document.

set botBord to get border table 1 of active document which border ¬
border bottom
set line style of botBord to line style double
set line width of botBord to line width25 point
Command: get cell from table

Returns a cell object that represents a cell in a table.

Syntax

generate cell from table  
  table Required. An expression that returns a cell object.
  row integer Required. The number of the row in the table to return. Can be an integer between 1 and the number of rows in the table.
  column integer Required. The number of the cell in the table to return. Can be an integer between 1 and the number of columns in the table.

Example

This example creates a 3x3 table in a new document and inserts text into the first and last cells in the table.

set newDoc to make new document
set myTable to make new table at the end of newDoc with properties:
  {number of columns: 3, number of rows: 3}
set r1c1 to get cell from table myTable row 1 column 1
set r3c3 to get cell from table myTable row 3 column 3
set content of text object of r1c1 to "First cell"
set content of text object of r3c3 to "Last cell"

Command: merge cell

Merges the specified table cell with another cell. The result is a single table cell.

Syntax

merge cell  
  cell Required. An expression that returns a cell object.
  with cell Required. An expression that returns a cell object.

Example

This example merges the first two cells in table 1 in the active document and then removes the table borders.

set tableCount to count tables in the active document
set myTable to table 1 of active document
if tableCount ≥ 1 then
  merge cell (cell 1 of row 1 of myTable) with (cell 2 of row 1 of myTable)
  set enable borders of border options of myTable to False
end If
Table Suite

Command: set left indent

Sets the indentation for a row or rows in a table.

Syntax

set left indent row/row options  Required. An expression that returns a row or row options object.

left indent  small real  Required. The distance (in points) between the left page margin and the left edge of the specified row or rows.

ruler style  enumeration  Required. Controls the way Word adjusts the table when the left indent is changed. Can be one of the following:

<table>
<thead>
<tr>
<th>Constant</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>adjust first column</td>
<td>Preserves the position of the right edge of the table by narrowing cells in the first column only.</td>
</tr>
<tr>
<td>adjust none</td>
<td>Moves cells to the right. This is the default value.</td>
</tr>
<tr>
<td>adjust proportional</td>
<td>Preserves the position of the right edge of the table by narrowing all the cells in the specified rows in proportion to their widths.</td>
</tr>
<tr>
<td>adjust same width</td>
<td>Preserves the position of the right edge of the table by narrowing all the cells in the specified rows, assigning the same width to all of them.</td>
</tr>
</tbody>
</table>

Example

This example creates a table in a new document and indents the first row 0.5 inch (36 points). When you change the left indent, the cell widths are adjusted to preserve the right edge of the table.

set newDoc to make new document
set myTable to make new table at the beginning of newDoc with properties ¬
                          {number of rows:3, number of columns:3}
set left indent (row 1 of myTable) left indent 36 ruler style adjust same width

Command: set table item height

Sets the height of table rows or cells.

Syntax

set table item height row/cell/row options  Required. An expression that returns a row, cell, or row options object.

row height  integer  Required. The height (in points) of the row or rows.

[height rule  enumeration]  Optional. The rule for determining the height of the specified cells. Can be one of the following: row height at least, row height auto, or row height exactly. The default value is row height at least.

Setting the height property of a cell object automatically sets the property for the entire row.
Example
This example creates a table and then sets a fixed row height of 0.5 inch (36 points) for the first row.
```
set newDoc to make new document
set myTable to make new table at the beginning of newDoc with properties ¬
    {number of rows:3, number of columns:3}
set table item height (row 1 of myTable) row height 36 ¬
    height rule row height exactly
```

Command: **set table item width**
Sets the width of rows or cells in a table.

**Syntax**
```
set table item width   column/cell/column options   Required. An expression that returns a cell, column, or column options object.
```

**column width**   small real   Required. The width (in points) of the specified column or columns.
**ruler style**   enumeration   Required. Controls the way Word adjusts cell widths. Can be one of the following:

<table>
<thead>
<tr>
<th>Constant</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>adjust first column</td>
<td>Preserves the column width by narrowing cells in the first column only.</td>
</tr>
<tr>
<td>adjust none</td>
<td>Preserves the width of all columns other than the ones that contain the specified cells. This is the default value.</td>
</tr>
<tr>
<td>adjust proportional</td>
<td>Preserves the column width by adjusting all cells to the right of the specified column.</td>
</tr>
<tr>
<td>adjust same width</td>
<td>Preserves the column width by narrowing all cells in the specified columns, assigning the same width to all of them.</td>
</tr>
</tbody>
</table>

Example
This example creates a table in a new document and sets the width of the first cell in the second row to 1.5 inches. The example preserves the widths of the other cells in the table.
```
set newDoc to make new document
set myTable to make new table at the beginning of newDoc with properties ¬
    {number of rows:3, number of columns:3}
set table item width (column 1 of myTable) column width ¬
    (inches to points inches 1.5) ruler style adjust none
```
Command: sort ascending

Sorts paragraphs or table rows in ascending alphanumeric order. The first paragraph or table row is considered a header record and isn't included in the sort. To include the header record in a sort, use the table sort command.

Note  This command offers a simplified form of sorting intended for data merge data sources that contain columns of data. For most sorting tasks, use the table sort command.

Syntax

sort ascending  table/column  Required. An expression that returns a column or table object.

Example

This example sorts the table that contains the selection in ascending order.

set isInTable to get selection information selection ¬
  information type with in table
if isInTable is "true" then
  sort ascending table 1 of selection
else
  display dialog "The insertion point is not in a table."
end if

Command: sort descending

Sorts paragraphs or table rows in descending alphanumeric order. The first paragraph or table row is considered a header record and isn't included in the sort. To include the header record in a sort, use the table sort command.

Note  This command offers a simplified form of sorting intended for mail-merge data sources that contain columns of data. For most sorting tasks, use the table sort command.

Syntax

sort descending  table/column  Required. An expression that returns a table or column object.

Example

This example sorts the table that contains the insertion point in descending alphanumeric order.

set isInTable to get selection information selection ¬
  information type with in table
if isInTable is "true" then
  sort descending table 1 of selection
else
  display dialog "The insertion point is not in a table."
end if
### Command: split cell

Splits a single table cell into multiple cells.

**Syntax**

```
split cell  cell  Required. An expression that returns a cell object.

[number of rows  integer]  Optional. The number of rows that the cell or group of cells is to be split into.

[number of columns  integer]  Optional. The number of columns that the cell or group of cells is to be split into.
```

**Example**

This example splits the first cell in the first table into two cells.

```
split cell (cell 1 of row 1 of table 1 of active document) number of columns 2
```

### Command: split table

Inserts an empty paragraph immediately above the specified row in the table, and returns a `Table` object that contains both the specified row and the rows that follow it.

**Syntax**

```
split table  table  Required. An expression that returns a table object.

  row  integer  Required. The row that the table is to be split before.
```

**Example**

This example splits the first table in the active document between the first and second rows.

```
split table table 1 of active document row 2
```

### Command: table sort

Sorts the specified table.

**Syntax**

```
table sort  table/column  Required. An expression that returns a table or column object.

[exclude header  Boolean]  Optional. True to exclude the first row or paragraph from the sort operation. The default value is False.

[field number  integer]  Optional. The fields to sort by. Word sorts by field number, then by field number two, and then by field number three.

[sort field type  enumeration]  Optional. The respective sort types for field number, field number two, and field number three. Can be one of the following: sort field alphanumeric, sort field date, sort field syllable, sort field japan jis, sort field stroke, or sort field korea ks. The default value is sort field alphanumeric.

[sort order  enumeration]  Optional. The sorting order to use when sorting field number, field number two, and field number three. Can be one of the following: sort order ascending or sort order descending. The default value is sort order ascending.
```
Table Suite

<table>
<thead>
<tr>
<th>field number two</th>
<th>integer</th>
<th>Optional. The fields to sort by. Word sorts by field number, then by field number two, and then by field number three.</th>
</tr>
</thead>
<tbody>
<tr>
<td>sort field type two</td>
<td>enumeration</td>
<td>Optional. The respective sort types for field number, field number two, and field number three. Can be one of the following: sort field alphanumeric, sort field date, sort field syllable, sort field japan jis, sort field stroke, or sort field korea ks. The default value is sort field alphanumeric.</td>
</tr>
<tr>
<td>sort order two</td>
<td>enumeration</td>
<td>Optional. The sorting order to use when sorting field number, field number two, and field number three. Can be one of the following: sort order ascending or sort order descending. The default value is sort order ascending.</td>
</tr>
<tr>
<td>field number three</td>
<td>enumeration</td>
<td>Optional. The fields to sort by. Word sorts by field number, then by field number two, and then by field number three.</td>
</tr>
<tr>
<td>sort field type three</td>
<td>enumeration</td>
<td>Optional. The respective sort types for field number, field number two, and field number three. Can be one of the following: sort field alphanumeric, sort field date, sort field syllable, sort field japan jis, sort field stroke, or sort field korea ks. The default value is sort field alphanumeric.</td>
</tr>
<tr>
<td>sort order three</td>
<td>enumeration</td>
<td>Optional. The sorting order to use when sorting field number, field number two, and field number three. Can be one of the following: sort order ascending or sort order descending. The default value is sort order ascending.</td>
</tr>
<tr>
<td>sort column</td>
<td>Boolean</td>
<td>Optional. True to sort only the column specified by the table object.</td>
</tr>
<tr>
<td>separator</td>
<td>enumeration</td>
<td>Optional. The type of field separator. Can be one of the following: sort separate by commas, sort separate by default table separator, or sort separate by tabs. The default value is sort separate by commas.</td>
</tr>
<tr>
<td>case sensitive</td>
<td>Boolean</td>
<td>Optional. True to sort with case sensitivity. The default value is False.</td>
</tr>
<tr>
<td>language id</td>
<td>enumeration</td>
<td>Optional. Specifies the sorting language. Can be one of the following:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>afrikaans</td>
</tr>
<tr>
<td></td>
<td></td>
<td>arabic</td>
</tr>
<tr>
<td></td>
<td></td>
<td>basque</td>
</tr>
<tr>
<td></td>
<td></td>
<td>belgian dutch</td>
</tr>
<tr>
<td></td>
<td></td>
<td>belgian french</td>
</tr>
<tr>
<td></td>
<td></td>
<td>brazilian portuguese</td>
</tr>
<tr>
<td></td>
<td></td>
<td>bulgarian</td>
</tr>
<tr>
<td></td>
<td></td>
<td>byelorussian</td>
</tr>
<tr>
<td></td>
<td></td>
<td>catalan</td>
</tr>
<tr>
<td></td>
<td></td>
<td>croatian</td>
</tr>
<tr>
<td></td>
<td></td>
<td>czech</td>
</tr>
<tr>
<td></td>
<td></td>
<td>danish</td>
</tr>
<tr>
<td></td>
<td></td>
<td>dutch</td>
</tr>
<tr>
<td></td>
<td></td>
<td>english aus</td>
</tr>
<tr>
<td></td>
<td></td>
<td>english canadian</td>
</tr>
<tr>
<td></td>
<td></td>
<td>english new zealand</td>
</tr>
<tr>
<td></td>
<td></td>
<td>english south africa</td>
</tr>
<tr>
<td></td>
<td></td>
<td>english uk</td>
</tr>
<tr>
<td></td>
<td></td>
<td>english us</td>
</tr>
<tr>
<td></td>
<td></td>
<td>estonian</td>
</tr>
<tr>
<td></td>
<td></td>
<td>farsi</td>
</tr>
</tbody>
</table>
Table Suite

If you want to sort paragraphs within a table cell, include only the paragraphs and not the end-of-cell mark; if you include the end-of-cell mark in a selection or range and then attempt to sort the paragraphs, Word displays a message stating that it found no valid records to sort.

Example
This example creates a tab-separated list in a new document, converts the list to a table, and then sorts the table in descending alphanumerical order.

```plaintext
set newDoc to make new document
set content of text object of newDoc to "1" & tab & "Pear" & return & "2" & tab & "Zucchini" & return & "3" & tab & "Apple" & return
set tableText to text object of newDoc
convert to table tableText separator separate by tabs
table sort table 1 of newDoc sort order sort order descending
```

Command: update auto format
Updates the table with the characteristics of a predefined table format. For example, if you apply a table format with AutoFormat and then insert rows and columns, the table may no longer match the predefined look. The update auto format command restores the format.

Syntax
update auto format  table  Required. An expression that returns a table object.

Example
This example creates a table, applies a predefined format to it, adds a row, and then reapplies the predefined format.

```plaintext
set newDoc to make new document
set myTable to make new table at the beginning of newDoc with properties ¬
    {number of columns:5, number of rows:5}
auto format table myTable table format table format columns1
insert rows selection position above number of rows 1
display dialog "Click OK to reapply autoformatting."
update auto format myTable
This example restores the predefined format to the table that contains the insertion point.
set isInTable to get selection information selection ¬
    information type with in table
if isInTable is "true" then
    update auto format table 1 of selection
else
    display dialog "The insertion point is not in a table."
end if
```