

Word AppleScript fundamentals


The Microsoft Word AppleScript topics consist of typical and popular subroutines (macros) in VBA, and their equivalents in AppleScript.

[Use the activate command when testing scripts](#)

If you are testing scripts using Macintosh Script Editor, you may want to begin each script with an **activate** command on its own line, and as the first line inside the `tell application "Microsoft Word"` block. That command brings Word to the front so that you can see what is happening. The **activate** command is not included in the sample scripts here, but you may want to use it.

You do not need to include **activate** when you are running saved scripts from the Script menu while Word is in the front. There may also be occasions when you are running a script from elsewhere and you don't want Word to come to the front.

[Prerequisite reading](#)

Before starting to script in Word, it is a good idea to read the "Using the Word Dictionary" section (especially the "Working with text range objects" topic) in the [Microsoft Word 2004 AppleScript Reference](#)  (<http://download.microsoft.com/download/1/3/E/13E8AE25-78F7-41A8-B252-F09C465CE29C/Word2004AppleScriptRef.pdf>).

Although the reference guide is largely aimed at AppleScripters who are unfamiliar with the object model for Word, it is full of useful knowledge for all scripters.

[Classes and properties with the same name](#)

Although it may seem confusing, application developers avoid using the same name for a class and a property for good reason.

The AppleScript class that corresponds to VBA's **Range** object is **text range**, which is found in the Text Suite. The AppleScript property that corresponds to VBA's **Range** property, which occurs in countless classes, is almost always **text object**, which returns an object of **text range** class.

Similarly, in AppleScript, the **font** class is the "type" of the **font object** property in other classes.

[Changes to text ranges](#)

Text ranges that you define are not dynamic. You can manipulate them. However, if you change a text range's start or end position, or its content, the variable that you use to reference it will no longer return the range. You have to reset the variable to the result returned or get ahold of it some other way.

This construction might cause some changes when you convert your macros from VBA to AppleScript, but there is virtually always some way to work around this difference. For more information about ranges that you change, see [Ranges are not dynamic](#) and [Bookmarks](#).