

Bookmarks

Using bookmarks

To get started, depending on your version of Microsoft Word, you can make bookmarks show by doing more or less the following:

1. On the **Word** menu, click **Preferences**, and then click **View**.
2. Under **Show**, select the **Bookmarks** check box.

Below are Visual Basic for Applications (VBA) and AppleScript code samples. The first two are used to insert text at a placeholder bookmark, which is a point in the text that you selected for your bookmark. Compare the following examples:

VBA

```
ActiveDocument.Bookmarks("myBookmark").Range.Text= "Inserted Text"
```

AppleScript

```
tell application "Microsoft Word"
    set content of text object of bookmark
        "myBookmark" of active document to "Inserted Text"
end tell
```

The next examples are used to insert text at an enclosing bookmark, which is some text that you selected as your bookmark. The code also recreates the bookmark, which would otherwise vanish. Compare the following examples:

VBA

```
Dim bmRange As Range

Set bmRange = ActiveDocument.Bookmarks("myBookmark").Range
bmRange.Text = "Inserted Text"
ActiveDocument.Bookmarks.Add _
    Name:="myBookmark", _
    Range:=bmRange
```

AppleScript

```
tell application "Microsoft Word"
    set bmRange to text object of bookmark "myBookmark" of active document
    set content of bmRange to "Inserted Text"
    make new bookmark at active document with properties
        {name:"myBookmark", text object:bmRange}
end tell
```

Dynamic ranges affect scripts for bookmarks

However, when you use the AppleScript code for the enclosing bookmark, the result is a placeholder bookmark in a completely different location in the document. In AppleScript, you cannot keep the same variable for a range after you change it, and ranges are not dynamic. As soon as you change the text of the **text range** of the bookmark, the range assigned to `bmRange` disappears. To verify this, you can run two tests. For the first, do the following to the code block for the enclosing bookmark:

1. Create a new second line inside the `tell` block, just after defining `bmRange` in the first line, by inserting this statement: `properties of bmRange`.
2. End the script after your new second line with this statement: `end tell`.

The result is an enormous record of properties for this text range.

For the second test, do the following to the original code block for the enclosing bookmark:

1. Create a new third line inside the `tell` block, just after setting the content of `bmRange` to "Inserted Text", by inserting this statement: `properties of bmRange`.
2. End the script after your new third line with this statement: `end tell`.

You expect to get the same enormous record of properties with the **content** property changed. Instead, you get no result at all, because `bmRange` does not exist any more. Therefore, you cannot use it to define the **text object** of your new bookmark.

[Make your own bookmark by finding its location](#)

Instead, you do the following:

1. Find the **start of content** for `bmRange` before you alter it, which is identical to the **start of bookmark** for the bookmark.
2. Measure the length of `bmRange`; that is, **count** the string you are inserting.
3. Add the length to the **start of content** result to get the new **end of content** and the new **end of bookmark**.

Note The **start of content** is an insertion point whose number is the same as that of the character it comes after. Remember that the document's first insertion point and **start of content** are 0, not 1. Therefore, to get the new **end of content**, you don't need to subtract 1 after you add the length of the new text.

Use these two numbers for the **start of bookmark** and **end of bookmark** properties to make your new bookmark. These properties are also advantageous because they are designed to have values assigned to them, whereas the **text object** property is read-only.

The following is the complete script for this procedure:

```
tell application "Microsoft Word"
    set bmRange to text object of bookmark "myBookmark" of active document
    set bmStart to bmRange's start of content
    set content of bmRange to "Inserted Text"
    set bmEnd to bmStart + (count "Inserted Text")
```

```
make new bookmark at active document with properties
    {name:"myBookmark", start of bookmark:bmStart,
      end of bookmark:bmEnd}
end tell
```

This works. It had to be adapted from the first VBA translation to accommodate the fact that ranges are not dynamic in AppleScript. The alternative approach utilizes some of the many properties, classes, and commands available.

There's almost always a solution. When you encounter issues in translating your macros from VBA to AppleScript, if you have a sense of why things work as they do, you will undoubtedly find the solutions.