

## Create a document

The code to make a new, blank document is different in AppleScript than it is in Visual Basic for Applications (VBA). Compare the following examples:

### VBA

```
Application.Documents.Add
```

### AppleScript

```
make new document
```

In AppleScript, you generally specify most of a document's properties at creation time, using a **with properties** parameter. Any elements, if needed, are added afterwards.

[Create a new document in Word](#)

However, that is not the case in AppleScript with Microsoft Word.

You cannot specify properties when creating a document. You have to create it first, and then you can alter it. The process is the same in VBA: You create a document, using the **Add** method, and then you can alter it. So VBA scripters will be familiar with this process.

In VBA, the only arguments that you can specify with `Documents.Add` are **Template**, **NewTemplate**, **DocumentType**, and **Visible**. The 50+ properties for **Document** have to be specified later. It is the same in AppleScript.

For example, in VBA, you can set the font and a bit of content with the following code:

```
Set NewDoc = Documents.Add
With NewDoc.Content
    .Font.Name = "Arial"
    .Text = "Here is some Text."
End With
```

Meanwhile, an AppleScripter would expect to do the same thing with the following code:

```
tell application "Microsoft Word"
    set newDoc to make new document with properties
        {text object: {font object:{name:"Arial"},
        content: "Here is some text."}}
end tell
```

**Note** Even though **text object** is listed in the dictionary as read-only, such properties can usually be set at inception.

However, if you use the code above, all you get is a blank new document. Instead, you have to create the **document** first, and then set the document's **text object** properties afterwards, as shown in the following example:

```
tell application "Microsoft Word"
    set newDoc to make new document
    tell newDoc's text object
        set name of font object to "Arial"
        set content to "Here is some text."
    end tell
end tell
```

Can you use the four arguments that you can set with VBA's `Documents.Add` statement in AppleScript?

It is unlikely that you will use `.Visible = False` in a macro; nonetheless, even if you wanted to, you can't use it in AppleScript to make an invisible document. However, you can set the **collapsed** property of the document's window 1 to *true*, which minimizes it after you create it.

In VBA, if you set the **NewTemplate** argument to *True*, you can preset a new document to be a template. However, you can't do this in AppleScript. That said, you can save a new document as a template later. When you use the **save as** command, set its **file format** parameter to *template*.

If you need to convert a VBA macro that uses `NewTemplate:=True`, ignore that statement in AppleScript and use `save as file format format template` when you save.

Similarly, the only value for **DocumentType** that works on the Macintosh, other than the default value, *wdNewBlankDocument*, is *wdNewWebPage*. Again, if you convert a VBA macro that uses the statement `DocumentType:= wdNewWebPage`, ignore it, and use `file format format HTML` when saving.

You can also set **view** to *online view* after you create a document, as shown in the following example:

```
set view type of view of window 1 of newDoc to online view
```