

Create a document from a template

Using the attached template property

In AppleScript, you can make a new document from a template, as you can in Visual Basic for Applications (VBA), using the **Template** argument of the **Add** method. Though using **make new document** and its **with properties** parameter doesn't work, you can set the **attached template** property (which is not read-only) of the new document to any template you want, as shown in the following example:

```
tell application "Microsoft Word"
    set newDoc to make new document
    set attached template of newDoc to
        "Mac HD:Applications:" &
        "Microsoft Office 2004:My Templates:Test Template.dot"
    get name of attached template of newDoc
end tell
```

Note When the dictionary says to provide the "name" of the template, that means the full path. The paths in these examples are broken into shorter bits, but you can enter the whole colon-delimited path of a template. Or, type `POSIX file "" as alias as Unicode text` in Macintosh Script Editor and drag the file between the quotes.

However, there is a bug that occurs when running the code above. The example compiles, runs, and you see the correct name as the result of the last line, which, in this case, is `Test Template.doc`. However, nothing changes in the document: no styles or macros are added.

Workaround for the attached template property

There is a workaround. After you set the attached template, insert it using **insert file**. That gets the text, including headers and footers, styles, and macros. All that it is missing is the page setup—the margins, line spacing, and so on. If the template uses different settings than your Normal template, including larger header space, for example, you can add more code that opens the template for an instant as a **document** so that it can retrieve its page setup settings.

Here is the code for the workaround:

```
set templatePath to "Mac HD:Applications:" &
    "Microsoft Office 2004:My Templates:Test Template.dot"
tell application "Microsoft Word"
    set newDoc to make new document
    set attached template of newDoc to templatePath
    insert file at text object
        of newDoc file name templatePath
end tell
```

Be aware that if you set an attached template to a document, you will get the macro warning if you have that option checked in Security preferences and your template contains macros. Though it appears that you can turn the warning off temporarily using the **automation security** property in **application**, that doesn't work. This limitation is likely an intentional security feature.

Because you need to use the template path twice, set a variable `templatePath` to it.

Workaround that includes the page setup settings

If you want to include the **page setup** settings (the margins, and so on), use the following longer version:

```
set templatePath to "Mac HD:Applications:" &
    "Microsoft Office 2004:My Templates:Test Template.dot"
tell application "Microsoft Word"
    activate
    set newDoc to make new document
    set attached template of newDoc to templatePath

    set theTemplate to attached template of newDoc
    -- a template object now
    set editableTemplate to open as document theTemplate
    -- a document
    set pageSetupProps to properties
        of (get page setup of editableTemplate)
    set lineNumberProps to properties
        of (get line numbering of pageSetupProps)
    close editableTemplate saving no
    tell page setup of newDoc
        set its properties to pageSetupProps
        tell its line numbering
            set its properties to lineNumberProps
        end tell
    end tell

    insert file at text object
        of newDoc file name templatePath
end tell
```

In the longer version, the **open as document** command opens a template as a **document**, so all of the **document** properties, including **page setup**, are available.

On the next line, you use the **properties** property of **page setup**, which returns a record of all of its properties. This is very efficient because it sends just one AppleEvent instead of 30 separate ones, and you don't have to specify each one in the script and set a variable to each. The **properties** property is available for nearly all classes in all Microsoft Office applications.

The **line numbering** property of **page setup** is a reference, so you also get all of the properties for **line numbering**. Otherwise when you try to set another document's **page setup** properties to this **properties** record, it would balk at being set to another document's item.

Then you close the template, having garnered from it all the information you need.

Next is the rather amazing bit. You can set the `newDoc` **page setup** properties to the variable that contains the whole properties record, and the same for the **line numbering** properties. This saves you the trouble of specifying 30 properties by 30 more variables. Also, the 30 properties contain only one property—**class**—that's read-only. It's the same for the line numbering properties.

Then you can insert the template file.

Workaround that eliminates the macro security warning

If you don't want to see or to have to turn off the macro security warning, or if you don't want the template to momentarily open, you can try a different approach. You can substitute the following code for the **make new document with properties** statement:

```
do Visual Basic "Documents.Add Template:=\"Mac HD:\" &
```

```
"Applications:Microsoft Office 2004:My Templates:Test Template.dot\""
```

With this statement, your scripts that convert macros that use Documents.Add Template will work in Microsoft Word 2004 for Mac.