

# Illustrator Scripting Guide Errata

## Running The AppleScript Sample Scripts

All the scripts in the Scripting Guide are meant to be executable as written. To run a script from the guide, select and copy the text in Acrobat then paste the script text into your script editor.

Unfortunately, there are some AppleScript scripts in the guide which will not compile as copied. The error is usually caused by lines being wrapped for display without a continuation character ('↵') being added to the end of the line. These compilation errors can be fixed by either adding a continuation character (option-L) to the end of the line or by deleting the return at the end of the line to unwrap the line.

## AppleScript Corrections

Page 40: The AppleScript script has the translate command included as part of the second comment. The script should be

```
tell application "Adobe Illustrator 10"
  -- first, manually select the text art item from the previous exercise or use
  -- AppleScript to make a selection
  set selection to text art item 1 of current document

  -- There are 72 points per inch. To translate an item by 2 inches to the right
  -- and 6 inches to the left, multiply by 72
  translate selection delta x (2 * 72) delta y (6 * 72)
end tell
```

Page 41: The conversions for Q's is wrong. The correct conversion factors are:

1Q = 0.23mm = 0.652 points.

Page 81: In example 6.2, the second line should be:

```
set pathItemList to every path item
```

Page 91: In example 8.1, the fourth to the last line has a "↵" character at the end of the line which will cause a compilation error.

Page 93: In example 9.1, the second line needs a continuation character at the end to compile. The line should be

```
tell application "Finder" to ↵
```

Page 97: The description for example 11.1 should read:

The values in a fixed rectangle can be used in a number of ways in a script.

Page 106: In example 13.1, the fifth line needs a continuation character at the end to compile. The line should be:

```
current document ↵
```

Page 111: In example 16.1, the kind property in the properties list should be gradient type. The line should be:

```
{name:pGradientName, gradient type:radial, entire gradient:entireGradient}
```

Page 212: The second line of the example for making a new symbol has an invalid document reference. The line should be:

```
with properties {source art:group item 1 of document 1}
```

Page 218: Example 48.1 for tags is not a valid script. Setting the URL property of a page item will not create a tag. The correct way to create a tag is

```
tell application "Adobe Illustrator 10"  
  set pathRef to make rectangle in document 1 with properties{name:"rectPath"}  
  make new tag at pathRef with properties {name:"MyTag", value:"a value"}  
  get properties of tags of pathRef  
end tell
```

# Visual Basic Corrections

Page 279: The third and fourth lines of the first script example need to be combined to create an executable script.

```
Dim appRef as New Illustrator.Application
Dim docRef as Illustrator.Document
set docRef = appRef.Open("C:\temp\afFile.ai")
```

Page 304: The script in example 10.2 assumes it is working on an RGB document, and in the last line of the script sets the stroke color of the new path to “Orange”. This swatch is an RGB swatch and is available only in RGB documents. For this reason, running this script on a CMYK document will result in a "There is no such element" run time error.

Page 326: The note for ExportOptionsPhotoshop refers to ExportOptionsPS5. This was the name of the class in the Illustrator 9.0 Scripting. Photoshop export options class has been renamed to ExportOptionsPhotoshop.

The description for example 19.1 doesn't match the comment in the example script. The script will export a Photoshop 6 document.

Page 364: In example 29.1, two lines of code were combined that shouldn't have been.

```
Set frontDocument = appRef.ActiveDocument countOfLayers =
frontDocument.Layers.Count
```

The corrected lines are:

```
Set frontDocument = appRef.ActiveDocument
countOfLayers = frontDocument.Layers.Count
```

Page 382: In example 35.1, if the document contains embedded (rather than linked) raster or placed images, a “There is no file associated with this item” runtime error will be returned:

Page 448: In example 56.1, the contents of a text art item is being set directly

```
tagText = "Tag: (" & tagItem.Name & " , " & tagItem.Value & ")"
```

To set the contents, you need to use the contents property

```
tagText.Contents = "Tag: (" & tagItem.Name & " , " & tagItem.Value & ")"
```

Page 485: In example 68.1, getting the word count should use the Words collection, not the TextLines collection.

```
numWords = numWords + textArtRange.Words.Count
```