

## Tips for Dosbox and Applescript

by Michael Paine

[http://www4.tpg.com.au/users/aoaug/mac\\_osx.html](http://www4.tpg.com.au/users/aoaug/mac_osx.html)

Updated 28 May 2014

### To create a Applescript that starts an application within Dosbox

1. Start the Applescript editor
2. `do shell script "/path1/DosBox.App/Contents/MacOS/DosBox /path2/oa4.bat"`

where **path1** is the location of the Dosbox app and **path2** is the location of the Dos app that you wish to run in Dosbox. Note that **path2** will become the c: drive within Dosbox.

### To save a script as an application:

1. Choose File > Save.
2. Choose a location in which to save the file.
3. In the Save dialog, choose "application" from the File Format pop-up menu.
4. Enter a name for the script (".app" is the name extension).
5. Select any combination of options. The options are:

- Run Only: Saves the application in a non-editable format. *USE THIS AFTER FIRST SAVING YOUR APPLESCRIPT AS A NORMAL FILE (FOR LATER EDITS)*
  - Startup Screen: Causes the script application to display a dialog upon launch containing the text from the script Description field. (See "Description Field."). *OPTIONAL - NOT MUCH POINT*
  - Stay Open: Causes the application to remain active until you quit it. Choose this option if the script contains an idle handler. *DON'T CHOOSE THIS*
  - Requires Classic: Choose this option if you want the application to run on Mac OS 9 systems. *DON'T CHOOSE THIS*
6. Click Save.

### To change the icon of the Applescript application

1. Find a suitable icon to use. GIFs and other graphics files can be converted to an OS X application icon with Graphic Converter.
2. Right-click on the icon file and select Get Info.
3. Click on the icon in the Get Info window and press Cmd-C
4. Find the Applescript application. Get Info for it.
5. Click on the icon and press Cmd-V. The icon should be replaced with your chosen one. If not then the icon graphic has a problem.

After changing the icon you can drag the application to your dock.

### To announce the launch of Dosbox

Add this line, or similar, to your Applescript

**say** "Starting dosbox" **using** "Kathy"  
([using "Kathy"] is optional – see System Preferences for a list of voices. Note that it is case sensitive)

### To print from Open Access to the OS X printer

Create a folder called Print and add it to the Open Access searching order. When printing choose FILE and specify the file location as the Print folder (eg Save as: PRINT:JUNK ). Create the following Applescript and call it, say, PrintFolderAdditions.scpt.

```
on adding folder items to this_folder after receiving added_items
    delay 20 -- time for job to finish?
    repeat with each_item in added_items
        tell application "Printer Setup Utility" to open each_item
        delay 20 -- time before file is deleted from folder

        tell application "Finder" to delete each_item
        tell application "Printer Setup Utility"
            quit
        end tell

    end repeat
end adding folder items to
```

Run FolderActionsSetup script. Select the Print folder you created previously and assign the PrintFolderAdditions script to it.

Now whenever Open Access (or any other application) sends a text file to the Print folder it is printed and deleted.

UPDATE: MAY 2014

PRINTER SETUP UTILITY changed to ADDPRINTER.APP several years.

Mavericks 10.9.3 has broken the function of ADDPRINTER.APP

```
on adding folder items to this_folder after receiving these_items
    try
        tell application "Finder"
            repeat with i from 1 to number of items in these_items
                try
                    set this_item to item i of these_items
                    set the path_string to this_item as string
                    set the final_path to POSIX path of the path_string
```

```

        do shell script "/usr/bin/lp -o cpi=12 -o page-left=36 "
                                & final_path & ""
    on error error_message
    end try
    delay 30

    tell application "Finder" to delete this_item

        end repeat
    end tell
    on error error_message
    end try
end adding folder items to

```

-o cpi=12 sets characters per inch to 12 (default is 10)

-o page-left=36 sets the left margin to 36 points (about 1/2" - default is zero margin)

To print from another application:

```

on adding folder items to this_folder after receiving myAddedItems
    repeat with myCounter from 1 to (count myAddedItems)
        tell application "<my app>"
            set myDocument to open item myCounter of myAddedItems
            print myDocument using ",preset>" print dialog no
        end tell
    end repeat
end adding folder items to

```

### To run a synchronisation with a Shuffle/iPod then eject the device

```

set the alert_message to "Ready to synchronise? (Ensure Shuffle is
    plugged in and allow up to 1 minute for script to run and quit
    itself)"

```

```

display dialog the alert_message buttons {"Yes", "No"} default button
    2 with icon 1 giving up after 10
set the user_choice to the button returned of the result

```

```

if user_choice is "Yes" then
    tell application "Synk"
        activate
    end tell
    -- get ready to send keystrokes
    tell application "System Events"
        -- make Synk the active application
        tell process "Synk"
            set frontmost to true

```

```

        open "Applications/synk_shuffle.snkd" -- location of Synk
            batch file
        delay 2
    end tell
    -- send the keystrokes
    keystroke "r" using {command down} -- Run Synk batch
    delay 20
    keystroke "q" using {command down} -- Quit Synk

end tell
set the alert_message to "Synchronisation has finished. Do you
    wish to eject the Shuffle?"

display dialog the alert_message buttons {"Yes", "No"} default
    button 2 with icon 1 giving up after 10
set the user_choice to the button returned of the result

if user_choice is "Yes" then
    tell application "Finder"
        eject "<shuffle name>"
    end tell
end if
end if

```

## ***General Applescript stuff***

From "APPLESCRIPT FOR ABSOLUTE STARTERS" by Bert Altenburg  
<http://www.applescriptsourcebook.com/>

## **Numbers and calculations**

```

set pictureWidth to 8
set pictureHeight to 6
set pictureSurfaceArea to pictureWidth * pictureHeight

```

Use the following symbols, officially known as operators, for doing basic mathematical calculations.

- + for addition
- for subtraction
- / for division
- \* for multiplication

The exponent is written using the exponent symbol ^. Here is a script [4] that calculates the

volume of a cube.

[4]

set cubeEdgeLength to 10.0

set cubeVolume to cubeEdgeLength ^ 3

## Text

set actressRating to "very pretty"

set resultingString to nameOfActress & " is" & actressRating

display dialog resultingString

-----

set theLength to the length of "I am"

## Lists

display dialog stringToBeDisplayed buttons {"So, so", "Don't know ",  
"Yes"}

set myList to {"winter", "summer"}

set item 2 of myList to "spring"

get myList

-----

set myList to {"a", "b", "c", "d", "e", "f", "g", "h"}

set shortList to items 2 through 5 of myList

-----

set theListLength to the length of {"first", "last"}

-----

set itemized to every character of "Hi there."

## Data input (record)

set stringToBeDisplayed to "Julia is a pretty actress."

set tempVar to display dialog stringToBeDisplayed buttons {"So, so",  
"Don't know", "Yes"}

set theButtonPressed to button returned of tempVar

display dialog "You pressed the following button " & theButtonPressed

-----

set temp to display dialog "Who is your favorite actress?" default answer  
""

set textEntered to text returned of temp

-----

set personalData to {age:30}

## Conditionals

```
set ageEntered to 73
set myAge to 30
if ageEntered = myAge then
    display dialog "You are as old as I am."
else
    display dialog "You are not as old as I am." -- if false
end if
```

-----

For numbers

= is (or, is equal to)

> is greater than

< is less than

>= is greater than or equal to

<= is less than or equal to

Negative formulations are possible as well, for example, 'is not greater than'. If you type '/=', it is automatically reformatted at compilation as #, which is short for 'is not'.

For strings

begins with (or, starts with)

ends with

is equal to

comes before

comes after

is in

contains

does not start with

does not contain

is not in

For lists

begins with

ends with

contains

is equal to

is in

-----

```
display dialog "" buttons {"", "", ""} default button 3
set the button_pressed to the button returned of the result
if the button_pressed is "" then
    -- action for 1st button goes here
```

```

else if the button_pressed is "" then
-- action for 2nd button goes here
else
-- action for 3rd button goes here
end if

-----
For records
contains
is equal to -- or =
[15]
set x to {name:"Julia", occupation:"actress"}
if x contains {name:"Julia"} then display dialog "Yes"

```

## Error trapping (Try)

```

try
beep
set x to 1 / 0
say "You will never hear this!"
end try
say "The error does not stop this sentence being spoken"

-----

try
set x to 1 / 0
on error the error_message number the error_number
display dialog "Error: " & the error_number & ". " & the error_message
buttons {"OK"} default button 1
end try

-----

on warning()
display dialog "Don't do that!" buttons {"OK"} default button "OK"
end warning
To use it, your script must call the handler, like this:
warning()

```

## Files and folders

```

tell application "Finder"

```

```

open folder "Macintosh HD:users:ik:Documents"
end tell
-----
tell application "Finder"
open file "Macintosh HD:users:ik:Documents:report.cwk"
end tell
-----
tell application "Finder"
move file "Macintosh HD:Users:ik:Documents:report.cwk" to the trash
end tell

```

## Loops (Repeat)

```

set conditionMet to false
repeat while conditionMet is false
-- if (some test is passed) then execute the following statement
-- set conditionMet to true
end repeat
-----
set conditionMet to false
repeat until conditionMet is true
-- if (some test is passed) then execute the following statement
-- set conditionMet to true
end repeat
-----
repeat with counter from 1 to 5
say "I drank " & counter & " bottles of coke."
end repeat
-----
repeat with counter from 1 to 5 by 2
say "I drank " & counter & " bottles of coke."
end repeat
-----

```

## Sub-routines

```

on circleArea(radius)
    set area to pi * (radius ^ 2)
end circleArea
set areaCalculated to circleArea(3)
-----
on older(a)

```



```
if a > 30 then
    return "older"
end if
return "not older"
end older
set theAge to older(73)
```

```
-----
set aVariableName to (load script "path here")
```

To use a handler of the compiled script, you must use a tell block.

```
tell aVariableName
    handlerName()
end tell
```

## Speech

say "You have mail" using "Cellos" saving to "announce\_mail.aiff"  
will create the sound file in the root directory.

## **Sample Application**

```
-- snap2GC Applescript by Michael Paine Dec 2004
-- For Mac OS X with GraphicConverter
-- http://www4.tpgi.com.au/users/aoaug/mac\_osx.html
--- This initiates the Unix screencapture function to capture an
--- area of the screen selected with the mouse (-i) and
--- saves it to the clipboard (-c) then
--- starts GraphicConverter with the contents of the
--- clipboard (by sending the Command-j keystrokes)
-- For best usage, load into Script Editor and Save As an Application
-- with the Runtime box checked. Drag the application to the Dock for
--- ready access.
--
-- Excute Unix command
do shell script "screencapture -i -c"
-- Ensure GraphicConverter is active (caution: no error trapping here)
tell application "GraphicConverter"
    activate
end tell
-- get ready to send keystrokes
tell application "System Events"
    -- make GC the active application
    tell process "GraphicConverter"
        set frontmost to true
    end tell
    -- send the keystrokes
    keystroke "j" using {command down}
end tell
```