

Applescript Tutorial 4

Is it UNIX or is it not

As was mentioned in the previous tutorial one potential problem is SMILES files often arrive as UNIX files, and there are two different line ending conventions in Mac OS X: Mac-style (lines end with return: "\r" or ASCII character 13) and Unix-style (lines end with line-feed: "\n" or ASCII character 10), so if we try to read a Unix file available here [temp_unix.txt](#) we have a problem. As you can see the entire text has been read in as a single value.

```

set mol_list to {}

set theFile to (choose file with prompt "Select the file:" of type {"TEXT"}) as alias

open for access theFile

set theData to read theFile using delimiter return

close access theFile

tell current application
  choose file with prompt "Select the file:" of type {"TEXT"}
  alias "Macintosh HD:Users:swain:Desktop:AppScript_tut:AppScript_tut_3:temp_unix.txt"
  open for access alias "Macintosh HD:Users:swain:Desktop:AppScript_tut:AppScript_tut_3:temp_unix.txt"
  read 172
  read alias "Macintosh HD:Users:swain:Desktop:AppScript_tut:AppScript_tut_3:temp_unix.txt" using delimiter "
"
  {"c1ccccc1 benzene
Ic1ccccc1 iodobenzene
O=C1CCCCC1 cyclohexanone
NC1CCCCC1 cyclohexamine
CN(C)c1ccnc1 3-dimethylaminopyridine
N1(c2ccccc2)CCNCC1 phenylpiperazine"}
  close access alias "Macintosh HD:Users:swain:Desktop:AppScript_tut:AppScript_tut_3:temp_unix.txt"
end tell

```

We need to alter the previous script to do two things, firstly detect the line-endings to identify whether the file is a UNIX or Mac file type, we then need to use the appropriate delimitator in both the import and write to file. The first part we do by reading in part of the file (100 characters) as shown in the script below, we then see if the result contain a line feed (ASCII character 10) or a return (ASCII character 13).

```

set {lf, return} to {ASCII character 10, ASCII character 13}

set theFile to (choose file with prompt "Select the file:") as alias

set the_result to read theFile for 100
if (the_result contains lf) then
  set delim to lf
  set delim_1 to "Unix File"
else if (the_result contains return) then
  set delim to return
  set delim_1 to "Mac File"
end if
display dialog delim_1

```

We can then replace the delimitator with the variable "delim" for both the read

set theData to read theFile using delimiter delim

and add the correct line-endings to the output

set mol_list to mol_list & delim

The full script now looks like this, it will now read either UNIX or Mac files and then write the output in the corresponding UNIX or Mac format. Some people will no doubt have noticed that the output is test2.smi, this is the correct file extension for SMILES files, unfortunately the ".smi" extension also corresponds to a "self-mounting image".

```

set mol_list to {}
set the_compounds to {}
set all_mols_list to {}
set mol_props_list to {}
set theData to {}

set {lf, return} to {ASCII character 10, ASCII character 13}

set theFile to (choose file with prompt "Select the file:") as alias

set the_file_path to GetParentPath(theFile)

set theSaveFile to the_file_path & "test2.smi"

--display dialog theSaveFile
set the_result to read theFile for 100
if (the_result contains lf) then
    set delim to lf
    set delim_1 to "Unix File"
else if (the_result contains return) then
    set delim to return
    set delim_1 to "Mac File"
end if
display dialog delim_1

open for access theFile
--UNIX file
--set theData to read theFile using delimiter "\n"
set theData to read theFile using delimiter delim
close access theFile

set text item delimiters to tab
repeat with i from 1 to count of items in theData
    set theLine to text items of item i of theData
    copy theLine to the end of mol_list
end repeat
set text item delimiters to ""

set num_compounds to count of items in mol_list

repeat with i from 1 to num_compounds

    set mol_props_list to {}
    set the_compound to item i of mol_list
    set the_SMILES to item 1 of the_compound
    set the_name to item 2 of the_compound
    --display dialog the_SMILES
    --display dialog the_name
    set the clipboard to the_SMILES

    tell application "CS ChemDraw Ultra"

        activate

        if enabled of menu item "Paste" then do menu item "SMILES" of menu "Paste Special" of menu "Edit"

        set the_CD_SMILES to SMILES of selection
        set Elem_Anal to Elemental Analysis of selection
        set Exact_mass to Exact Mass of selection
        set Mol_Form to Molecular Formula of selection
        set Mol_weight to Molecular Weight of selection

        copy the_SMILES to the end of mol_props_list
        copy the_name to the end of mol_props_list
        copy the_CD_SMILES to the end of mol_props_list
        copy Elem_Anal to the end of mol_props_list
        copy Exact_mass to the end of mol_props_list
        copy Mol_Form to the end of mol_props_list
        copy Mol_weight to the end of mol_props_list

        if enabled of menu item "Paste" then do menu item "Clear" of menu "Edit"

```

```

        --display dialog (item 3 of mol_props_list)
    end tell
    copy mol_props_list to the end of all_mols_list
end repeat

repeat with i from 1 to num_compounds
    set mol_list to item i of all_mols_list
    -- convert list to text
    set old_delim to AppleScript's text item delimiters
    set AppleScript's text item delimiters to tab
    set mol_list to mol_list as text
    --set mol_list to mol_list & "\n"  needs UNIX line endings
    set mol_list to mol_list & delim
    set AppleScript's text item delimiters to old_delim
    my write_to_file(mol_list, theSaveFile, true)
end repeat

on GetParentPath(theFile)
    tell application "Finder" to return container of theFile as text
end GetParentPath

on write_to_file(this_data, target_file, append_data)
    try
        set the target_file to the target_file as text
        set the open_target_file to ¬
            open for access file target_file with write permission
        if append_data is false then ¬
            set eof of the open_target_file to 0
        write this_data to the open_target_file starting at eof
        close access the open_target_file
        return true
    on error
        try
            close access file target_file
        end try
        return false
    end try
end write_to_file

```

Errors and Omissions in the file

Sometimes files contain SMILES strings but do not contain the corresponding name (or molecule ID) at the moment the script will fail at the point:

set the_name to item 2 of the_compound

Since there will be no item 2. We can avoid this problem by modifying the script as shown below. First try to extract the name if present then if there is no name construct a name based on the position of the molecule in the file (e.g. the fifth molecule will be called molecule_5). The completed script can be downloaded [here](#).

```

try
    set the_name to item 2 of the_compound
    --If no name set name to molecule and number
end try
if the_name = "" then
    set the_name to "molecule_" & i
end if

```

The completed script can be downloaded [here](#).