Applescript & ASObjC ’Things to watch out for’ list

Much of this list is from suggestions by Shane Stanley, and to a lesser extent, Yvan Koenig. Sorry if I haven’t given credit where credit’s due.

---

1. In ASObjC, the following ***sometimes*** does not work under Sierra (but works in Applescript). If you have code failures, keep this in mind.

```
tell application "System Events"
    set ff to name of files of (path to desktop)
end tell
```

you must use

```
tell application “Finder”
```

---

2. Generally, the code

```
set aPath to ((path to desktop) & "This is a file Name.dat" as text)
```

should be written…

```
set aPath to (path to desktop as text) & "This is a file Name.dat"
```

because the first example uses the ‘**path to** desktop’ as an alias, and it may fail if the default text item delimiters in your code have been altered.

---

3. In Sierra, I’ve found that the use of a *‘property’ variable* ***usually*** now requires the addition of **‘my’** in front of it, even when just reading the value. Before, the use of…

```
property test : "test"
set my test to "not test"
set secondTest to test
```

would work, but now you need to use…

```
set secondTest to my test
```

---

4. If trying to create a Folder Action, this code will do it. Note that it actually attaches the script to an individual folder.…

```
tell application "Finder" to set FAName to name of alias {path:to:targetFolder}
    tell application "System Events"
        if not (folder action FAName exists) then
            make new folder action at end of folder actions with properties {enabled:true, name:FAName, path:"path:to:targetFolder"}
            tell folder action FAName to make new script at end of scripts with properties {name:"myScriptName.scpt"}
        end if
    end tell
end tell
```

However, the above Folder Action addition handler assumes that the script to use already exists. If you need to load in a new script, first use something like the following, where the script to copy is embedded inside the actual Applications ‘Scripts’ folder.

```
set pathToMe to (path to current application as alias) as text
set pathToEmbeddedResourcesScripts to pathToMe & "Contents:Resources:Scripts"
##########################
# Duplicate ftp Downloader script second
##########################
set pathToFolderActionscripts to (path to Folder Action scripts folder from user domain) as text
tell application "Finder"
    set tempPath to pathToEmbeddedResourcesScripts & ":ftp Downloader.scpt"
    duplicate file tempPath to folder pathToFolderActionscripts with replacing
end tell
```

And to actually turn Folder Actions ‘ON’, use…

```
tell application "System Events" to set folder actions enabled to true
```

5. If you want to capture fairly accurate time intervals, use the following, which is good to one ten thousandths of a second.

```
use framework "Foundation"
set x to current application's NSDate's timeIntervalSinceReferenceDate()
set y to current application's NSDate's timeIntervalSinceReferenceDate()
y - x
--> 0.001749992371
```

6. If you’d like to use actual spelled numbers in text, instead of numeric variables, send the variable to this handler…

```
set xAsText to my convertNumberToText(x)

on convertNumberToText(thisNumber)
	try
		set p to 0
		set thisNumber to thisNumber as integer
		set p to 1
		tell current application's NSNumberFormatter to set resultingText to localizedStringFromNumber_numberStyle_(thisNumber, current application's NSNumberFormatterSpellOutStyle)
		set p to 2
		return (resultingText as string)
	on error errmsg
		tell application "System Events" to display dialog "convertNumberToText " & errmsg & " p = " & p giving up after 20
	end try
end convertNumberToText
```

7. Example of how to alter text by the use of text delimiters.

If you'd like to set your default printer by Applescript, use these handlers (by Yvan Koenig)

```
# Use this handler to set the CUPS printer to the "une grande imprimante" #
     large page printer

   try
      set theLargePagePrinter to "une grande imprimante"
      set nameOfPrinter to ""
      set nameOfPrinter to my remplace(theLargePagePrinter, space, "_")
      do shell script ("lpoptions -d " & nameOfPrinter)
   end try

   # replaces every occurrences of d1 by d2 in the text t

   on remplace(t, d1, d2)
      local oTIDs, l
      set {oTIDs, AppleScript's text item delimiters} to {AppleScript's
         text item delimiters, d1}
      set l to text items of t
      set AppleScript's text item delimiters to d2
      set t to l as text
      set AppleScript's text item delimiters to oTIDs
      return t
   end remplace
```

---

8. Fetching 'Mail' mailboxes. Note the use of names *must match with capitals* where used.

```
    tell application "Mail"
       mailboxes of drafts mailbox
    end tell
    tell application "Mail"
       mailboxes of inbox
    end tell
    tell application "Mail"
       mailboxes of mailbox "Year 2016"
    end tell
```

```
tell application "Mail"
    mailboxes of mailbox "Brians stuff"
end tell
tell application "Mail"
    mailboxes of mailbox "Filed personal" of mailbox "Brians stuff"
    set mid to message id of message 1 of mailbox "Filed personal" of mailbox "Brians stuff"
    tell application "System Events" to display dialog mid as text giving up after 20
    # returns B26BDC1400FA7D4A973E2D1FCDB-C7E430345304860C4@*****change02.****.GEELONG
end tell
```

9. Sometimes in an App, you might want to play a certain sound, that is saved as a resource in your Project. Here's how.

```
property theSound : ""

# The next line could be set anywhere, preferably at the start of code.
set my theSound to current application's NSSound's
      soundNamed:"Glass"

tell my theSound
    its setDelegate:me
    its play()
    do shell script ("sleep 1")
end tell
```

In Sierra, and possibly El Capitan, the 'play' command seems to have a bug, in that once called, the played sound stays open, and subsequent calls don't work.

Shane Stanley has suggested a fix, which is in the last handler below . He also added a delay that matches the duration of the sound, otherwise called sounds can 'overlap'.

Note that you can save your own sounds within the actual Application, and they can be virtually any sound format, such as mp3, or aiff. They will be automatically found by name, which does not need to use the file extension. So, just use *'glass by three',* instead of *'glass by three.mp3'*.

If you want to record your own sounds, the FreeWare 'Audacity' is excellent.

```
	on mainLoop()
		repeat
			tell application "System Events" to display dialog "Click to play the Sound 'Glass'" buttons {"Quit me", "Play"} giving up after 60
			set temp to button returned of the result
			if temp = "Play" then my PlayGlassThree()
			if temp = "Quit me" then
				say "quitting"
				tell me to quit
				exit repeat
			end if
		end repeat
	end mainLoop

	on PlayGlassThree()
		set theSoundPlayGlassThree to current application's NSSound's soundNamed:"Glass"
		tell theSoundPlayGlassThree
			its |stop|()
			set theDuration to its duration()
			its play()
		end tell
		current application's NSThread's sleepForTimeInterval:theDuration
	end PlayGlassThree
```

---

10. Sometimes you might want to test for the existence of an item, and do so using a POSIX path. You cannot do so directly from the Finder, but here's how…

POSIX file belongs to the Osax Standard Additions so it's not supposed to be called in a tell application block.

```
set posixPath to "what you want"
tell application "Finder"
	exists (POSIX file posixPath)
end tell
```

violate the rule and so it fails.

```
set posixPath to "what you want"
```

```
tell application "Finder"
        exists (get POSIX file posixPath)
end tell
```

violate the rule too and so it fails.

```
set posixPath to "what you want"
tell application "Finder"
tell me to (POSIX file posixPath)
        exists result
end tell
```

apply the official workaround and so it works.

```
set posixPath to "what you want"
(POSIX file posixPath)
tell application "Finder"
        exists result
end tell
```

apply a logical syntax and so it works.

---

11. You should not use the *'delay'* command in ASObjC Projects. Read what happens if you use the following…

The delay command **is** broken in Yosemite, at least in applets. Run this script as an applet:

```
set time1 to current date
set progress total steps to 10
repeat with i from 1 to 10
        set progress description to "Processing " & i & " of " & 10
        delay 3
        set progress completed steps to i
end repeat
display dialog ((current date) - time1) as text
```

If you just sit and watch it, it behaves as you'd expect. But as soon as you move your mouse, whoosh -- it finishes post-haste.

So it looks like either the shell or ASObjC is the way to go.

One thing I'd suggest, though, is ***not*** using something like single long sleeps:

```
do shell script "sleep 30"
```

or:

```
current application's NSThread's sleepForTimeInterval:30
```

But rather:

```
repeat 30 times
    do shell script "sleep 1"
end repeat
```

or:

```
repeat 30 times
    current application's NSThread's sleepForTimeInterval:1
end repeat
```

That's because the delay command does some special stuff that avoids spinning cursors, and that doesn't happen with the other methods.

---

12. If making new Folders from your ASObjC Project, then read this…

One way…

```
try
    do shell script "mkdir -p " & quoted form of POSIX path of (my dailyServerFolder as text)
on error errmsg number errnum
    if my runForOz then tell application "System Events" to display dialog "Mail Manager Loop Runner setupPrintCoverPage setting server folder " & errmsg & " error number " & errnum & return & (my dailyServerFolder as text)
end try
```

But…

You're already using ASObjC, so there's no need to use shell scripting for stuff like creating folders; all it does is slow things down. Use something like this instead:

```
set {theResult, theError} to current application's NSFileManager's defaultManager()'s createDirectoryAtPath:(POSIX path of my dailyServerFolder)
```

```
withIntermediateDirectories:true attributes:(missing value) |error|:(reference)
if not (theResult as boolean) then
	set errmsg to theError's localizedDescription() as text
	if my runForOz then tell application "System Events" to display dialog "Mail Manager Loop Runner setupPrintCoverPage setting server folder " & errmsg  & return & my dailyServerFolder
end if
```

However, if you use a separate handler for each nested folder, it is cumbersome. In ASObjC you can use something like this, which creates a nested Folder within a Folder…

```
	use AppleScript version "2.4" -- Yosemite & later
	use framework "Foundation"
	use scripting additions
	--------------------------------------------------------------------------------------------

	set newDirPath to POSIX path of ((path to desktop folder as text) & "test01:test02:")
	its createDirectoryAtPath:newDirPath

------------------------------------------------------------------------------------------
--» HANDLER
------------------------------------------------------------------------------------------
	on createDirectoryAtPath:thePath
		set {theResult, theError} to current application's NSFileManager's defaultManager()'s createDirectoryAtPath:thePath withIntermediateDirectories:true attributes:(missing value) |error|:(reference)
		if not (theResult as boolean) then
			set errorMsg to theError's localizedDescription() as text
			error errorMsg
		end if
	end createDirectoryAtPath:

# Full script can be found at....
# http://lists.apple.com/archives/applescript-users/2016/Jun/msg00088.html
```

13. If you’re having trouble signing an Application not destined for the App Store, try this simple procedure…

Turn off Automatically manage signing.

Set Provisioning Profile to No (you only need that for sandboxing, iCloud or iOS).

Set Team to your certificate name.

Set Signing Certificate to Developer ID.

14. In an emergency, with the sound volume turned off, you might want to temporarily announce a message. Here’s how...

```
property trashEmptyNumber : 100
property trashEmptyTime : current date

tell application "Finder"
    set trashCount to count of items of the trash
end tell

if trashCount as integer ≥ trashEmptyNumber as integer then
    # Calculate a minimum time to pass before actually making the announcement.
    # The higher the set number, the longer to wait.
    if (current date) - (my trashEmptyTime) > (120 * ((trashEmptyNumber as integer) div 100)+600) then
        set vflag to false
        set v to (output volume of (get volume settings)) * 1000 div 1250 / 10 as number
        if v < 6 then
            set vflag to true
            set volume 6
            say ”warning. The trash is overflowing. Please empty it")
        end if
        if vflag then set volume v
        set my trashEmptyTime to (current date)
    end if
end if
```

15. Here’s a pretty much interruple-proof routine for setting a mail rule.

```
on mailRuleCreator:sender
    try
        tell application "Mail"
            activate
            try
                do shell script ("sleep 0.2")
                tell application "System Events" to tell process "Mail"
                    try
                        click menu item "Message Viewer" of menu 1
                                of menu bar item "Window" of menu
                                bar 1
                    end try
                end tell
                if not (exists mailbox "* items to shift") then make
                      new mailbox with properties {name:"* items to
                      shift"}
                set theAddRuleMailbox to ""
                set inBoxList to name of every account whose enabled
                      is true as list
                set addText to "" as rich text
            end try
        end tell
        repeat
            if (count of inBoxList) = 1 then
                try
                    set theAddRuleMailbox to item 1 of inBoxList
                end try
                exit repeat
            else
                try
                    tell application "Mail"
                        activate
                        set theAddRuleMailbox to choose from list
                                inBoxList with prompt addText &
                                "Please choose an Account to which to
```

```
                            add the Mail rule " & return & return &
                            "\"Move to '* items to shift'\"." & return
                            with title "Mail Manager add Mail rule"
                            OK button name "Choose this Inbox Account"
                            cancel button name "You MUST
                            choose"
                end tell
            on error
                tell application "System Events"
                    try
                        click button "You MUST choose" of window
                                "Mail Manager add Mail rule" of
                                application process "Mail"
                    end try
                end tell
            end try
        end if
        try
            if theAddRuleMailbox ≠ "" then exit repeat
        end try
        set addText to "You MUST pick a mail account!" & return &
              return
    end repeat
on error errmsg number errNum
    tell application "System Events" to display dialog "Odds N
          Sods Setting Mail Rule error " & errmsg giving up after 20

end try
tell application "Mail"
    activate
    try
        if theAddRuleMailbox ≠ "" then
            try
                repeat with y from 1 to count of inBoxList
                    if item y of inBoxList as rich text is theAddRuleMailbox
                            as rich text then exit repeat
                end repeat
                set theMainAddresses to account y's email addresses
                      as list
                set my theMainEmailAddress to item 1 of theMainAddresses
                      as rich text
            on error errmsg number errNum
    tell application "System Events" to display dialog
           "Odds N Sods Setting Mail Address error " &
           errmsg & " number " & errNum giving up
           after 20
end try
try
    delete (every rule where name contains "items
           to shift")
end try
try
    delete (every rule where name contains
           "Autom")
end try
if not (exists rule "Move to '* items to shift'") then
    set x to 0
    repeat
        try
            set x to x + 1
            tell application "System Events" to tell
                    process "Mail"
                # tell application "Mail" to activate
                try
                    keystroke return
                end try
                click menu item "Preferences…" of
                        menu 1 of menu bar item
                        "Mail" of menu bar 1
                activate
                do shell script ("sleep 0.02")
                activate
                click button 8 of toolbar 1 of window
                        1
                #
                # Set descriptive name of rule
                #
                activate
                click button "Add Rule" of group 1 of
                        group 1 of window "Rules"
                # tell application "Mail" to activate
                set value of text field 1 of sheet 1 of
                        window "Rules" to "Move to '*
                        items to shift'"
                #
```

```
# First, select the 'all' menu item
#
activate
click pop up button 1 of sheet 1 of
        window "Rules"
activate
click menu item "all" of menu 1 of
        pop up button 1 of sheet 1 of
        window "Rules"
#
# Set first scroll area, 'Conditions'
#
activate
click pop up button 1 of scroll area 1
        of sheet 1 of window "Rules"
# The darn popups change numbers
        whilst 'Account' is being selec-
        ted
try
    # tell application "Mail" to activate
    activate
    click menu item "Account" of
            menu 1 of pop up button
            1 of scroll area 1 of sheet
            1 of window "Rules"
    do shell script ("sleep  0.02")
end try
# so we have to try both 1 & 2 popups
try
    # tell application "Mail" to activate
    activate
    click menu item "Account" of
            menu 1 of pop up button
            2 of scroll area 1 of sheet
            1 of window "Rules"
    do shell script ("sleep  0.02")
end try
activate
click pop up button 1 of scroll area 1
        of sheet 1 of window "Rules"
do shell script ("sleep  0.02")
activate
```

```
click menu item (theAddRuleMailbox
        as text) of menu 1 of pop up
        button 1 of scroll area 1 of
        sheet 1 of window "Rules"
do shell script ("sleep  0.02")
#
activate
click button "add criterion" of scroll
        area 1 of sheet 1 of window
        "Rules"
#
activate
click pop up button 3 of scroll area 1
        of sheet 1 of window "Rules"
activate
click menu item "Every Message" of
        menu 1 of pop up button 3 of
        scroll area 1 of sheet 1 of
        window "Rules"
do shell script ("sleep 0.02")
#
# Set second scroll area, 'Actions'
#
activate
click pop up button 1 of scroll area 2
        of sheet 1 of window "Rules"
do shell script ("sleep  0.02")
click menu item "Move Message" of
        menu 1 of pop up button 1 of
        scroll area 2 of sheet 1 of
        window "Rules"
do shell script ("sleep  0.02")
activate
click pop up button 2 of scroll area 2
        of sheet 1 of window "Rules"
do shell script ("sleep  0.02")
activate
click menu item "* items to shift" of
        menu 1 of pop up button 2 of
        scroll area 2 of sheet 1 of
        window "Rules"
do shell script ("sleep  0.02")
#
```

```
activate
click button "add action" of scroll area
            2 of sheet 1 of window
            "Rules"
do shell script ("sleep  0.02")
#
activate
click pop up button 3 of scroll area 2
            of sheet 1 of window "Rules"
do shell script ("sleep  0.02")
activate
click menu item "Run Applescript" of
            menu 1 of pop up button 3 of
            scroll area 2 of sheet 1 of
            window "Rules"
do shell script ("sleep  0.02")
activate
click pop up button 4 of scroll area 2
            of sheet 1 of window "Rules"
do shell script ("sleep  0.02")
try
      click menu item "Mail Manager
                  Caller" of menu 1 of pop
                  up button 4 of scroll area
                  2 of sheet 1 of window
                  "Rules"
on error
      try
            click menu item 1 of menu 1
                        of pop up button 4
                        of scroll area 2 of
                        sheet 1 of window
                        "Rules"
      end try
end try
do shell script ("sleep  0.02")
activate
click button "OK" of sheet 1 of win-
            dow "Rules"
do shell script ("sleep  0.02")
activate
try
```

```
                                    click button "Don’t Apply" of sheet
                                             1 of window "Rules"
                                on error
                                    keystroke return
                                end try
                                do shell script ("sleep  0.02")
                                try
                                    activate
                                    click button 1 of window 1
                                end try
                            end tell
                            my writeFile4("shouldWeInstallMailRule",
                                    "false")
                        on error errmsg number errNum
                        end try
                        if (exists rule "Move to '* items to shift'")
                               then exit repeat
                    end repeat
                end if
            end if
        on error errmsg number errNum
            tell application "System Events" to tell process "Mail"
                try
                    do shell script ("sleep 0.1")
                    # activate
                    click button "OK" of sheet 1 of window "Rules"
                end try
                try
                    # activate
                    click button 1 of window 1
                end try
            end tell
            display dialog "mailRuleCreator error " & errmsg
        end try
    end tell
end mailRuleCreator:
```

16. Here’s how to determine the amount of Hard Drive space you have left.

```
repeat
```

```
        set fileManager to current application's NSFileManager's de-
            faultManager()
        set attributes to fileManager's
            attributesOfFileSystemForPath:"/" |error|:(missing value)
        set freeBytes to (attributes's objectForKey:(current applica-
            tion's NSFileSystemFreeSize))'s longLongValue()
         set freeGBytes to (freeBytes div 1000) / 1000000 -->
            “17390.725"
         if freeGBytes < 10 then
            tell application "System Events" to display dialog "
                CRITICAL WARNING!!!" & return & return & return &
                "The amount of free hard drive space is down to " &
                freeGBytes & " GB." & return & return & "You must
                either free up more disk space and Try Again, or Quit
                me!." & return & return buttons {"Quit me", "Try
                again"} default button "Try again" giving up after
                3600
            if button returned of the result = "Quit me" then my
                MM_Quitter()
        else
            exit repeat
        end if
    end repeat
```

17. Here’s a routine for creating a Dock Icon that will ‘stick’ in the Dock, and not add a second icon.

```
on installDockIcon()
    try

        set theInfo to (current application's NSUserDefaults's alloc()'s
            init()'s persistentDomainForName:"com.apple.dock") as
            record
        set theMatches to get theInfo's |persistent-apps|
        set MatchingList to {}
        set x to 0
        repeat with thisRecord in theMatches
            set x to x + 1
            set theTestMatch to get thisRecord's |tile-data|
            set theMMMatch to theTestMatch's |file-label|
            if theMMMatch as text is "Mail Manager" then
```

```
                set end of MatchingList to item x of theMatches
            end if
        end repeat
        set existsMMDockIcon to my installDockItemsTest("Mail Manager")
        if (count of MatchingList) < 1 or not existsMMDockIcon then
            do shell script ("sleep 0.2")
            try
                set item_path to ((path to applications folder) & "Mail
                         Manager:Mail Manager.app" as text)
             end try
            do shell script ("sleep 0.2")
            try
                set item_path to POSIX path of item_path
                do shell script "defaults write com.apple.dock persistent-apps
                        -array-add '<dict><key>tile-data</
                        key><dict><key>file-data</
                        key><dict><key>_CFURLString</
                        key><string>" & item_path & "</
                        string><key>_CFURLStringType</key><integer>0</integer></dict></dict></dict>'"
                do shell script ("killall Dock")
            end try
        end if
    end try
end installDockIcon

on installDockItemsTest(theEntry)
    set x to 1
    try
        tell application "System Events"
            tell process "Dock"
                set t to (title of UI elements of list 1)
                set x to 0
                repeat with theTest in t
                    if theTest as text = theEntry then set x to x + 1
                end repeat
            end tell
        end tell
    end try
    return (x > 0)
end installDockItemsTest
```

18. Bill Cheeseman, Author of UI Browser, uses the following snippet to determine the Accessibility of an Application that has this snippet embedded in it…

```
tell application "System Events" to set GUIScriptingEnabled to UI elements enabled -- read-only in OS X 10.9 Mavericks and newer
```

However, under certain circumstances, too elaborate to list here, this might always return ‘true’.

When this occurs, I use the following, which seems reliable.

Loop this until the user sets it to true.

```
repeat
    tell application "TextEdit"
        activate
        do shell script ("sleep 0.1")
        try
            close every window saving no
        end try
        do shell script ("sleep 0.1")
        tell application "System Events" to tell process "TextEdit"
            try
                do shell script ("sleep 0.2")
                click menu item "Preferences…" of menu 1 of menu bar item "TextEdit" of menu bar 1
                set my assistTrial to true
                exit repeat
            on error errmsg number errnum
                set my assistTrial to false
            end try
        end tell
    end tell
    tell application "System Preferences"
        activate
        tell pane id "com.apple.preference.security" to reveal anchor "Privacy_Accessibility"
        repeat
            try
```

```
                            if not (exists window "Security & Privacy")
                                then exit repeat
                        end try
                        do shell script ("sleep 0.1")
                    end repeat
                    do shell script ("sleep 0.1")
                end tell
            end repeat
```

19. A number formatter.

```
    # Borrowed from Shane STANLEY
    on formatnumber:theNumber
        set theResult to current application's NSNumberFormatter's localizedStringFromNumber:theNumber numberStyle:(current application's NSNumberFormatterDecimalStyle)
        return theResult as text
    end formatnumber:
```

20. A shell script to quit an Application, wait a period (in this case 2 seconds), and re-start it. Useful if you have an Application installation script *within an Application*, that moves the new App from somewhere like the desktop, to the Application Folder, and needs to re-start it so the Finder knows where it is, or so a freshly started App 'knows' it's in Applications.

```
    do shell script ("killall 'Mail Manager'&&sleep 2&&open -a '/Applications/Mail Manager/Mail Manager.app' ")
```

21. Translating Objective-C into ASObjC.

This is one of the most frustrating things I've had to try and grasp. It's not straightforward, and Apple's documentation, as usual, seems to assume you *already know everything that you're looking for,* and fully understand all the technical jargon they throw at you. I've never found anything on Apples sites that explain things in simple, easy to understand language to the beginner, especially in some detail. Written by experts, for experts, I think.

To start off with, try this site…

http://macscripter.net/viewtopic.php?id=30373

22. Using Bar Codes can be a pain, but there's a free Bar Code high density checksum Bar Code available that's called Code 128, and these handlers show how to use it.

There is a maximum number of characters that you should not exceed, or the Bar Code may be un-scannable, depending on the Scanner. Also, the modulus count must not equal 34, or again, un-scannable.

The second handler works with text within the Maximum, (which you must set with **my** maxBarcodeCharacters), or it truncates if the text exceeds the maximum. It also ensures the Bar Code text, before adding the necessary start and end characters, does not end with a 'space' character, which also makes the Bar Code unreadable.

The first handler set a barcode for a date example, ending with 'Z', which indicates it's a Universal date setting. Notice the use of many property variables, some of which you may not require.

Read http://www.makebarcode.com/specs/code_128.html

A direct download link for a free Barcode 128 type A font. Download Code 128 Barcode Font

```
property maxBarcodeCharacters : 56
property actualDate : current date
property MMWorkingDate : current date
property theSavingDate : current date
property theZeroedDate : current date
property theDays : ""
property theHours : ""
property theMinutes : ""
property theSeconds : ""
property theYear : ""
property storeMonth : ""
property theAddedMonth : ""
property rtfdDateTimeName : ""
property printDateTimeName : ""
property attachmentbarCodeDateTimeName : ""
property barCodeDateTimeName : ""
property dailyName : ""
property actualProcessingStartDate : current date

    on setDateTimeName()
        try
            copy (current date) to my actualDate
            repeat
                set p to 0
                copy (my actualDate) - (time to GMT) to my MMWorkingDate # Set Universal time
                copy (my MMWorkingDate) - days to my theSavingDate
                set the time of my theSavingDate to 0
                copy my MMWorkingDate to my theZeroedDate
                set the time of my theZeroedDate to 0
                set thetempEntireseconds to time of my MMWorkingDate
                set my theDays to day of my MMWorkingDate
                if my theDays < 10 then set my theDays to "0" & my theDays as text
                set my theDays to my theDays as text
                set my theHours to thetempEntireseconds div 3600
                set my theMinutes to (thetempEntireseconds - ((my theHours) * 3600)) div 60
                set my theSeconds to thetempEntireseconds - ((my theHours) * 3600) - ((my theMinutes) * 60)
                if my theHours < 10 then set my theHours to "0" & my theHours as text
                if my theMinutes < 10 then set my theMinutes to "0" & my theMinutes as text
                if my theSeconds < 10 then set my theSeconds to "0" & my theSeconds as text
                set my theYear to year of my MMWorkingDate as text
                set my storeMonth to the month of my MMWorkingDate as text
                set tempmonth to characters 1 through 3 of my storeMonth as text
                copy ((offset of tempmonth in "jan feb mar apr may jun jul aug sep oct nov dec ") + 3) / 4 as integer to my theMonth
                if my theMonth < 10 then set my theMonth to "0" & my theMonth as text
                set my theAddedMonth to my theMonth as text
                set my rtfdDateTimeName to (my theYear & "-" & my theMonth & "-" & my theDays & " " & my theHours & my theMinutes & my theSeconds & "Z.rtfd") as text
                set my printDateTimeName to (my theYear & "-" & my theMonth & "-" & my theDays & " " & my theHours & my theMinutes & my theSeconds & "Z") as text
                set my attachmentbarCodeDateTimeName to (my theYear & my theMonth & my theDays & my theHours & my theMinutes & my theSeconds) as text
                set p to 1
                set theModulusCharacter to (my setUpModulusCount(attachmentbarCodeDateTimeName & "Z" as text)) as integer
                set p to 2
                if theModulusCharacter ≠ 34 then exit repeat
                set p to 3
                set my actualDate to (my actualDate) + 1
            end repeat
            set p to 6
            set my barCodeDateTimeName to (character id 209) & attachmentbarCodeDateTimeName & "Z" & (character id theModulusCharacter) & (character id 211) as text
            set p to 7
            set my dailyName to (my theYear & " " & my theMonth & " " & my theDays) as text
            set my actualProcessingStartDate to my actualDate
        on error errmsg number errnum
            tell application "System Events" to display dialog "setDateTimeName error " & errmsg & " number " & errnum & " p = " & p
        end try
        return
    end setDateTimeName

    on resetTheFileNameToABarcode(theText)
        local tempLength, tempMaxLength, theWords
        try
            set p to 1
            set drawLength to length of theText
            if drawLength > my maxBarcodeCharacters then set drawLength to my maxBarcodeCharacters
            set theWords to items 1 thru drawLength of theText as text
            set x to 0
            repeat 10 times
                set x to x + 1
                set p to 2
                repeat
                    if (character -1 of theWords) as text ≠ " " then exit
                          repeat
                    set theWords to (characters 1 thru -2 of theWords) as
                          text
                end repeat
                set tempLength to (count of theWords)
                set p to 2.1
                set tempText to my setUpTextCount(theWords)
                set p to 4
                set theModulusCharacter to my setUpModulusCount(theWords)
                set p to 5
                if theModulusCharacter ≠ 34 then exit repeat
                set theCount to count of paragraphs of theErrorList
                 set temp to count of theWords
                if temp ≥ ((my maxBarcodeCharacters) - 10) then
                    set theWords to (items 1 thru (temp - x) of theWords
                          & "*") as text
                else
                    set theWords to theWords & "*" as text
                end if
            end repeat
             set theNewFileBarcode to (character id 209) & tempText &
                (character id 211) as text
            return theNewFileBarcode as text
        on error errmsg
            tell current application
                activate
                display dialog "Error in resetTheFileNameToABarcode " &
                      errmsg & " p = " & p as text giving up after 10
            end tell
            set theWords to "Error in Barcode setting"
            set theModulusCharacter to my setUpModulusCount(theWords)
            set p to 4
            set theNewFileBarcode to (character id 209) & tempText &
                (character id theModulusCharacter) & (character id 211)
                as text
            return theNewFileBarcode
        end try
    end resetTheFileNameToABarcode
```

```
on setUpModulusCount(theText)
    local drawTextLength
    try
        set p to 0
        set eachCharacterCount to 104 # the value of the start character for Barcode 128 Set B
        set p to 0.1
        set drawTextLength to (count of theText) as number
        set p to 0.2
        if drawTextLength as real > ((my maxBarcodeCharacters) as real) then
            set p to 0.3
            set drawTextLength to ((my maxBarcodeCharacters)) as real
        end if
        set p to 0.4
        set theText to items 1 thru drawTextLength of theText as text
        set p to 1
        repeat with x from 1 to drawTextLength # We start counting from the first character of the actual text,and multiply it's adjusted value by its position
            set eachCharacter to item x of theText
            set eachCharacterAscii to (id of eachCharacter)
            if eachCharacterAscii < 127 then
                set subtractAsciiOffset to 32
            else
                set subtractAsciiOffset to 105
            end if
            set eachCharacterCount to eachCharacterCount + ((eachCharacterAscii - subtractAsciiOffset) * x)
        end repeat
        set p to 2
        set theModulusNumber to eachCharacterCount mod 103
        set p to 3
        set theAddOnAscii to 32

        set p to 4
        if theModulusNumber > 94 then set theAddOnAscii to 105

        set p to 5
        return (theModulusNumber + theAddOnAscii) as number
    on error errmsg number errnum
```

```
            tell application "System Events" to display dialog "setUpModulus-
                        Count error " & errmsg & " error " & errnum & " p = " &
                        p as text giving up after 20
            return ""
        end try
end setUpModulusCount
```

23. This is a series of handlers that show to create a 'Mail' message, with attachments. It attaches a 'Numbers' chart. Look for the final 'send' line.

If you want to use it, it will need altering to suit yourself, and the addition of some 'properties'. It is not a 'stand-alone' script, just a copy of part of a script I use to send six different types of reports. In my case, they are hourly (any hour, set from a series of tick boxes), a Daily, (sent at the end of the Universal day), a 'Special 24 hour' (last 24 hours data, sent at a time nominated by user), a Weekly, a Monthly, a Yearly.

You'll hopefully get the basics from this example.

Note that some resources are stored ***within*** the *actual Application file*, such as a Client logo…

```
set logoPath to pathToMe & "Contents:Resources:Report Logo.png" as text


on sendSpecialReport:{}
    try
        set p to 1
        set x to 0
        set theLastTally to 0
        set totalEmails to 0
        set TallyString to {}
        set theSpecialReportHour to (my theHandlerProperty)
        set my theTally to {}
        set totalPrintedJobs to 0
        set the_subject to ""
        try
            tell application "Finder" to set my theTally to (read file ((my
                        mailManagerDesktopFolderPath) & "yesterdays-
                        Tally" as text) using delimiter ",")
        on error
            my saythetext("Yesterdays tally file does not exist.")
        end try
```

```
if (count of my mailAddressesMainReportsList) > 0 and my theTally ≠ {} then
    set p to 2
    if theSpecialReportHour = 0 then
        set theDisplayHour to "12am"
    else
        if theSpecialReportHour = 12 then
            set theDisplayHour to "12pm"
        else
            if theSpecialReportHour < 12 then set theDisplayHour to theSpecialReportHour & "am" as text
            if theSpecialReportHour > 12 then set theDisplayHour to (theSpecialReportHour - 12) & "pm" as text
        end if
    end if
    set my theTally to my theTally as list
    set p to 3
    set theLastMidnight to (my tempDate) - (time to GMT) # Set Universal time
    set cd to current date
    if hours of cd < theSpecialReportHour then set cd to cd - days
    set time of cd to 0
    set p to 4
    set the_subject to "Special Report for " & theDisplayHour & ", for " & year of cd & " " & month of cd & " " & day of cd & ", eMail Tally."
    set p to 4.1
    set my actualDate to theLastMidnight
    set p to 5
    set TimeAdjust to time to GMT
    #
    # Now set up Numbers document
    #
    set my timeOffset to 25
    set p to 5.1
    set my itemDetails to "    Special Tally"
    set p to 5.2
    set my sheetName to ("Special Report" & return & return & "For " & theDisplayHour & ", " & year of cd & " "
           & (month of cd as text) & " " & day of cd & ",
           Special Printed Jobs Tally, local time." as text)
set p to 5.3
set my sendTheChart to true
set p to 5.4
my setUpNewNumbersTable("Special")
#
set p to 7
repeat
    if (theSpecialReportHour) > 23 then
        set theSpecialReportHour to theSpecialReportHour -
                      24
    else
        exit repeat
    end if
end repeat
if (theSpecialReportHour) < 24 then
    set yTime to 0
    repeat with x from (theSpecialReportHour) to 24
        set yTime to yTime + 1
        set theHourlyJobsTally to item (3 + (x * 2)) of my
                      theTally as number
        set totalPrintedJobs to totalPrintedJobs + theHourly-
                      JobsTally
        set theReadingTally to item (4 + (x * 2)) of my the-
                      Tally as number
        set totalEmails to totalEmails + theReadingTally
        set tempX to x # + ((TimeAdjust / hours) div 1) - 1
                      as number
        if tempX < 0 then set tempX to (tempX + 24)
        if tempX = 0 then
            set tempX to "12am" as text
        else
            if tempX = 12 then
                set tempX to "12pm" as text
            else
                if tempX = 24 then
                    set tempX to "12am" as text
                else
                    if tempX > 24 then
                        set tempX to (tempX - 24) &
                                                     "am" as
                                                     text
            else
                if tempX > 12 then
                    set tempX to (tempX - 12) & "pm" as text
                else
                    set tempX to tempX & "am" as text
                end if
            end if
        end if
    end if
end if
set p to 8
if tempX starts with "10" or tempX starts with "11" or tempX starts with "12" then
    set tempX to tempX & " " & tab
else
    set tempX to tempX & "    " & tab
end if
#if (count of tempX) < 4 then set tempX to "  " & tempX as text -- This counts the characters, and adds spaces if below 4
-- This builds string of asterixes for each hour
set GraphString to ""
set p to 9
repeat with y from 1 to theReadingTally
    set GraphString to GraphString & "•"
end repeat
if my displayHourlyJobsFlag and (theHourlyJobsTally - theReadingTally) > 0 then
    repeat with y from 1 to (theHourlyJobsTally - theReadingTally)
        set GraphString to GraphString & "◆"
    end repeat
end if
```

```
if not (my displayHourlyJobsFlag) then
	set theTallyInsert to theReadingTally as string
	if (count of GraphString) < 10 then set theTallyInsert to "  " & theReadingTally as string
else
	set theTallyInsert to theHourlyJobsTally as string
	if (count of GraphString) < 10 then set theTallyInsert to "  " & theHourlyJobsTally as string
end if
-- This add space if number below 10
set p to 10
-- Build Tally string
set end of TallyString to tempX & " :" & theTallyInsert & tab & GraphString & return
set columnThreeSetting to theHourlyJobsTally - theReadingTally
if columnThreeSetting < 0 then set columnThreeSetting to 0
if (my existsNumbersFlag) then
	try
		tell application "Numbers"
			activate
			tell document 1
				tell sheet 1
					tell table 1
						tell column 1 to set value of cell (yTime + 1) to te
```

```
mpX
tell column 2 to set value of cell (yTime + 1) to theReadingTally
if my displayHourlyJobsFlag then
    try
        tell column 3 to set value of cell (yTime
                                                                                        +
                                                                                        1)
                                                                                        to
                                                                                        col
                                                                                        u
                                                                                        m
                                                                                        nT
                                                                                        hr
                                                                                        ee
                                                                                        Se
                                                                                        tti
                                                                                        ng
                                            end try
                                        end if
                                    end tell
                                end tell
                            end tell
                        end tell
                    end try
                end if
            end repeat
        end if
        set p to 11
        # set totalPrintedJobs to (item 53 of my theTally) as integer
        set p to 12
    end if
on error errmsg number errnum
    tell application "System Events" to display dialog "Mailer's send-
             SpecialReport Part one error " & errmsg & " Error num-
             ber is " & errnum & " p = " & p & " x = " & x as text
             giving up after 40
end try

# Part Two
try
    set p to 1
    set x to 0
    set my actualDate to current date
    set my theTally to {}
    tell application "Finder" to set my theTally to (read file ((my
             mailManagerDesktopFolderPath & " RunningTally") as
             text) using delimiter ",")
    set p to 2
```

```
if (count of my mailAddressesMainReportsList) > 0 and my theTally ≠ {} then
    set p to 3
    repeat with x from 1 to theSpecialReportHour - 1
        set yTime to yTime + 1
        set theHourlyJobsTally to item (3 + (x * 2)) of my theTally as string
        set totalPrintedJobs to totalPrintedJobs + theHourlyJobsTally
        set theReadingTally to (item (4 + (x * 2)) of my theTally) as string
        set totalEmails to totalEmails + theReadingTally
        set tempX to x
        if tempX < 0 then set tempX to (tempX + 24)
        if tempX = 0 then
            set tempX to "12am" as text
        else
            if tempX = 12 then
                set tempX to "12pm" as text
            else
                if tempX = 24 then
                    set tempX to "12am" as text
                else
                    if tempX > 24 then
                        set tempX to (tempX - 24) & "am" as text
                    else
                        if tempX > 12 then
                            set tempX to (tempX - 12) & "pm" as text
                        else
                            set tempX to tempX & "am" as text
                        end if
                    end if
                end if
            end if
        end if
        #
        # Now, only fill in ordinary emails if not ChartOnly
        #
        set p to 5
set GraphString to ""
if tempX starts with "10" or tempX starts with "11" or tempX starts with "12" then
    set tempX to tempX & " " & tab
else
    set tempX to tempX & "    " & tab
end if
#if (count of tempX) < 4 then set tempX to "  " & tempX as text -- This counts the characters, and adds spaces if below 4
set thestring to ""
# set theLastTally to theLastTally + theReadingTally
if not (my displayHourlyJobsFlag) then
    set theTallyInsert to theReadingTally as string
    if (count of GraphString) < 10 then set theTallyInsert to "  " & theReadingTally as string
else
    set theTallyInsert to theHourlyJobsTally as string
    if (count of GraphString) < 10 then set theTallyInsert to "  " & theHourlyJobsTally as string
end if
-- This add space if number below 10
set p to 10
-- Build Tally string
if theReadingTally > 0 then
    repeat with y from 1 to (theReadingTally + 1) div 2
        set GraphString to GraphString & "•"
    end repeat
end if
if my displayHourlyJobsFlag and (theHourlyJobsTally - theReadingTally) > 0 then
    repeat with y from 1 to ((theHourlyJobsTally - theReadingTally) + 1) div 2
        set GraphString to GraphString & "◆"
    end repeat
end if
set end of TallyString to tempX & " :" & theTallyInsert & tab & GraphString & return
set p to 6
set columnThreeSetting to theHourlyJobsTally - theReadingTally
if columnThreeSetting < 0 then set columnThreeSetting to 0
set p to 7
if (my existsNumbersFlag) then
    try
        tell application "Numbers"
            activate
            tell document 1
                tell sheet 1
                    tell table 1
                        tell column 1 to set value of cell (yTime + 1) to tempX
                        tell column 2 to set value of cell (yTime + 1) to theReadingTally
```

```
                            if my displayHourlyJobsFlag then
                                try
                                    tell column 3 to set value of cell (yTime + 1) to columnThreeSetting
                                end try
                            end if
                        end tell
                    end tell
                end tell
            end tell
        end try
    end if
end repeat

set p to 13
set the_content to (my SetTheHeading() & ¬
    "Special 24 hour Report for " & theDisplayHour & ", for "
                    & weekday of cd & ", " & month of cd & "
                    " & day of cd & "." & return & return &
```

```
                         "Total Mail items processed : " & item 1 of
                         my theTally as string) & return & return &
                         ¬
            "For the recorded 24 hour period, we computer processed..."
                         & return & return & ¬
            "Total processed eMails for the period : " & totalEmails &
                         " (represented by '•')." & return & return
                         & ¬
            "Total Printed Jobs for the period is " & totalPrintedJobs
        set p to 14
        if my displayHourlyJobsFlag then set the_content to the_content
                   & " (represented by '◆')"
        set p to 15
        if (my displayHourlyJobsFlag) then
            set p to 16
            set the_content to the_content & return & return & the
                         TallyString & return & return & "In the
                         Chart, blue indicates processed emails," &
                         return & "green indicates extra Printed
                         Jobs over emails processed." & return &
                         return
        else
            set p to 17
            set the_content to the_content & "." & return & return &
                         the TallyString & return & return
        end if
        set p to 18
        set the_content to the_content & "Have a good day."
        set p to 19
        set textcolor to {80 * 256, 4 * 256, 99 * 256 - 1}
        set p to 20
        # tell application "System Events" to display dialog the_subject
                   & return & the_content & return & return &
                   mailAddressesHourlyReportsList as text
        set my sendTheChart to true
        my eMailit(the_subject, the_content, mailAddressesHourlyReportsList,
                   textcolor)
    end if
on error errmsg number errnum
    tell application "System Events" to display dialog "Mailer's sendSpecialReport
             Part Two error " & errmsg & " Error number
             is " & errnum & " p = " & p & " x = " & x as text
             giving up after 40
    end try
end sendSpecialReport:

on setUpNewNumbersTable(tempHeader)
    try
        set p to 1
        tell application "Numbers"
            set p to 3
            activate
            set p to 4
            set my hourCutOff to my hourCutOff as text
            set p to 5
            set myNewDoc to my makeNewDoc(("Mail Manager Email
                       Tally" & ".numbers" as text), (path to desktop))
                       --< go and create blank document
            set p to 6
            set x to 0
            repeat
                set x to x + 1
                do shell script ("sleep 1")
                if (exists document 1) then exit repeat
                if x = 10 then return
            end repeat
        end tell
        set p to 7
        if my timeOffset < 3 then set my timeOffset to 3
        set p to 8
        set theColumnCount to 2
        set p to 9
        if tempHeader is "Week Day" or my displayHourlyJobsFlag then
                 set theColumnCount to 3
        set p to 10
        tell application "Numbers"
            activate
            set p to 11
            if not (exists document 1) then
                set p to 12
                tell application "System Events" to display dialog
                             "Hourly Reports, Numbers document.
                             There is no document open." buttons
                             {"Cancel", "OK"} default button 2 giving
                             up after 40
                return
```

```
end if
set p to 13
set my theSheetName to my sheetName
activate
set p to 14
tell document 1
    set p to 15
    tell sheet 1
        set p to 16
        set this_table to make new table with properties {name:my theSheetName, column count:theColumnCount, row count:((my timeOffset))}
        set p to 17
        tell table (my theSheetName)
            -- set any global cell properties
            set p to 18
            set the height of every row to 18
            set p to 19
            set the width of columns 1 thru 1 to 110
            set p to 21
            set the vertical alignment of every row to center
            set p to 22
            set the alignment of every row to right
            set p to 23
            tell column 1
                -- set specific properties for the title column
                set the vertical alignment to center
                set the alignment to right
                -- insert labels
                set value of cell (1) to tempHeader
            end tell
            set p to 24
            set value of cell 1 of column "B" to "Emails"
            set p to 25
            if (count of columns) = 3 then
                set value of cell 1 of column "C" to "Extra Jobs"
                set the width of columns 2 thru 3 to 70
            end if
            set p to 26
```

```
                        set alignment of every cell of row 1 to center
                    end tell
                end tell
            end tell
        end tell
    on error errmsg number errnum
        if my RunForOz then tell application "System Events" to display
                  dialog "Mailer's setUpNewNumbersTable error " & errmsg
                  & " Error number is " & errnum & " p = " & p as
                  text giving up after 40
    end try
end setUpNewNumbersTable

on makeNewDoc(n, d)
    try
        set p to 1
        if not my existsNumbersFlag then return
        set p to 2
        tell application "System Events"
            set SSD to get delay interval of screen saver preferences
            if (exists process "ScreenSaverEngine") then set delay interval
                       of screen saver preferences to SSD
        end tell
        do shell script ("sleep 1")
        set p to 3
        try
            tell application "Finder" to move file ((path to desktop as
                       text) & "Hourly Tally" & ".numbers") to trash
        end try
        do shell script ("sleep 1")
        set p to 4
        tell application "Numbers"
            activate
            try
                close every document saving no
            end try
            do shell script ("sleep 1")
            set p to 5
            tell application "System Events" to tell process "Numbers"
                try
                    # do shell script ("sleep 2")
                    if exists window "Choose a Template" then
                        set p to 6
    repeat while exists window "Choose a Template"
        click button 2 of window 1
        keystroke return
        do shell script ("sleep 0.2")
    end repeat
else
    set p to 7
    try
        click button "New Document" of window 1
    end try
    set p to 8
    repeat
        set tempCount to count of windows
        if tempCount ≠ 0 then
            try
                close every document saving no
            on error
                exit repeat
            end try
            try
                close every window saving no
            on error
                exit repeat
            end try
        else
            exit repeat
        end if
    end repeat
    set p to 9
    keystroke "n" using command down
    set p to 10
    repeat until exists window "Choose a Template"
        do shell script ("sleep 0.1")
        if x ≥ 40 then exit repeat
    end repeat
    repeat while exists window "Choose a Template"
        click button 2 of window 1
        keystroke return
        do shell script ("sleep 0.2")
    end repeat
```

```
                end if
            on error
                try
                    set p to 11
                    click menu item "New" of menu 1 of menu bar
                                    item "File" of menu bar 1
                end try
            end try
        end tell
        set x to 0
        set p to 12
        repeat until exists window 1
            do shell script ("sleep 0.1")
            set x to x + 1
            if x ≥ 20 then exit repeat
        end repeat
        do shell script ("sleep 0.2")
        set temp to {}
        try
            set p to 13
            set temp to first item of ((name of windows) as list)
            delete every table of every sheet of every document
                        of window temp
        end try
        set p to 14
        tell application "System Events" to tell process "Numbers"
            try
                click menu item "Hide Inspector" of menu 1 of
                              menu bar item "View" of menu bar
                              1
            end try
            try
                click menu item "Hide Inspectors" of menu 1 of
                              menu bar item "View" of menu bar
                              1
            end try
        end tell
        return temp
    end tell
    if my RunForOz then say "closing numbers" without waiting until
              completion
on error errmsg number errnum
```

```
        if my RunForOz then tell application "System Events" to display dialog "Mailer's makeNewDoc error " & errmsg & " Error number is " & errnum & " p = " & p as text giving up after 40
    end try
end makeNewDoc

on eMailit(the_subject, the_content, textcolor)
    set p to 0
    try
        set theSendCompressedFile to 0
        set fileManager to current application's NSFileManager's defaultManager()
        set attributes to fileManager's attributesOfFileSystemForPath:"/" |error|:(missing value)
        set freeBytes to (attributes's objectForKey:(current application's NSFileSystemFreeSize))'s longLongValue()
        set freeBytes2 to my formatnumber:freeBytes
        set my freeGBytes to (freeBytes div 1000) / 1000000 --> "277390725"
        # set my freeGBytesDraw to (my formatnumber:(freeBytes div 1000) / 1000000) --> "277,390.725"
    on error errmsg
        tell application "System Events" to display dialog "eMailit 1 " & errmsg giving up after 20
    end try
    if the_subject contains "Daily Report" then
        try
            set p to 100.0
            if weekday of (current date) is in "Thursday" then
                set p to 100.1
                set vn to my readFile2("Store The Version IP.dat")
                set p to 100.2
                set y to ""
                repeat with x from (count of vn) to 1 by -1
                    set p to 100.3
                    if character x of vn = "." then exit repeat
                    set y to character x of vn & y as text
                end repeat
                set p to 100.4
                set versionNumber to y as number
            end if
        on error errmsg
        if my RunForOz then display dialog "Mail add-on bcc " & errmsg & return & "p  = " & p giving up after 30
    end try
end if
try
    if (count of mailAddressesMainReportsList) = 0 then
        my saythetext("There are no Email recipients set")
        if ("Hourly" is in my itemDetails) then
            my saythetext(" for hourly reports.")
        else
            if ("Special" is in my itemDetails) then
                my saythetext(" for special reports.")
            else
                my saythetext(" for main reports.")
            end if
            return
        end if
    end if
    repeat with x from 1 to 8
        # set the_content to the_content & return
    end repeat
end try
# Set up Numbers Chart to be saved to desktop
set p to 7
try
    if my sendTheChart then
        repeat 4 times
            my setUpTallyNumbersChartCreator()
            if (my existsNumbersFlag) then
                tell application "Numbers"
                    try
                        if exists chart 1 of sheet 1 of document 1 then
                            exit repeat
                        end if
                    end try
                end tell
            end if
        end repeat
    end if
end try
# Sets up & copies Chart
try
```

```
set theCCRecipients to my mailAddressesMainReportsList
repeat
    set p to 1
    set theSendCompressedFile to theSendCompressedFile + 1
    if theSendCompressedFile ≥ 3 then exit repeat
    tell application "Finder"
        try
            set pathToMe to path to current application as text
            set logoPath to pathToMe &
                        "Contents:Resources:Report
                        Logo.png" as text
            set PosixLogoPath to POSIX path of logoPath
        on error
            set logoPath to ""
        end try
        set p to 2
        if (count of logoPath) < 4 or not (exists file logoPath)
                    then
            try
                set pathToMe to (path to applications folder) &
                            "Mail Manager:Mail Manager.app:" as text
                set logoPath to pathToMe & "Contents:Resources:Report Logo.png" as
                            text
                set PosixLogoPath to POSIX path of logoPath
            end try
        end if
    end tell
    set p to 3
    if the_subject contains "Hourly" or the_subject contains
                "Special" then
        if theSendCompressedFile = 2 then exit repeat
        set theCCRecipients to {}
        repeat with eachAddress in mailAddressesMainReportsList
            set end of theCCRecipients to item 2 of eachAddress as text
        end repeat
    else
        set newListTrue to {}
        set newListFalse to {}
        set newListTotal to {}
    if the_subject contains "Annual Report" or the_subject contains "Monthly Report" or the_subject contains "Weekly Report" or the_subject contains "Daily Report" then
        repeat with eachAddress in theCCRecipients
            set end of newListTotal to item 2 of eachAddress as text
            # item 1 contains 'send compressed file' flag
            if (item 1 of eachAddress as text) is "true" then
                set end of newListTrue to item 2 of eachAddress as text
            else
                set end of newListFalse to item 2 of eachAddress as text
            end if
        end repeat
        if theSendCompressedFile = 1 then copy newListFalse to theCCRecipients
        if newListFalse = {} then set theSendCompressedFile to 2
        if theSendCompressedFile = 2 then
            copy newListTrue to theCCRecipients
            if newListTrue = {} then exit repeat
        end if
        # This will happen on first pass (theSendCompressedFile=1), if NewListTrue = {}
        if newListTrue = {} then
            copy newListTotal to theCCRecipients
        end if
    end if
end if
try
    tell application "Mail"
        activate
        set p to 4
        set newMessage to make new outgoing message with properties {visible:true, subject:the_subject, content:the_content, sender:my theMainEmailAddress}
        set p to 5
        tell content of newMessage
```

```
try
    set y to count of the_content
    set color of characters 1 thru y to textcolor
end try
try
    set x to offset of my theBusinessName in the_content
    set font of characters x thru (x + (count of my theBusinessName) - 1) to "Helvetica Bold"
    set color of characters x thru (x + (count of my theBusinessName) - 1) to {56342, 2442, 607}
end try
try
    set x to offset of ("Hourly " & my hourCutOff & " Report" as rich text) in the_content
    if x ≠ 0 then
        set xx to count of ("Hourly " & (my hourCutOff) & " Report" as rich text)
        try
            set font of characters x thru (x + xx - 1) to "Cochin"
        end try
        set size of characters x thru (x + xx - 1) to 20
        set color of characters x thru (x + xx - 1) to {43 * 256, 0, 256 * 256 - 1}
    end if
end try
try
```

```
    set x to offset of ("Special " as rich text)
                      in the_content
    if x ≠ 0 then
        set xx to offset of ("Total Mail " as
                            rich text) in
                            the_content
        try
            set font of characters x thru (xx
                                 - 2) to
                                 "Cochin"
        end try
        set size of characters x thru (xx - 2)
                            to 20
        set color of characters x thru (xx - 2)
                            to {43 * 256,
                            0, 256 * 256 -
                            1}
    end if
end try
try
    set x to offset of "Daily Report" as rich
                      text in the_content
    if x ≠ 0 then
        set xx to offset of "Total Mail items
                            processed :" in
                            the_content
        try
            set font of characters x thru (xx
                                 - 3) to
                                 "Cochin"
        end try
        set size of characters x thru (xx - 3)
                            to 20
        set color of characters x thru (xx - 3)
                            to {43 * 256,
                            0, 256 * 256 -
                            1}
    end if
end try
try
    set x to offset of "Weekly" as rich text in
                      the_content
    if x ≠ 0 then
```

```
        set xx to offset of "For the" in the_-
                            content
        try
            set font of characters x thru (xx
                                  - 3) to
                                  "Cochin"
        end try
        set size of characters x thru (xx - 3)
                            to 20
        set color of characters x thru (xx - 3)
                            to {43 * 256,
                            0, 256 * 256 -
                            1}
    end if
end try
try
    set x to offset of "Monthly" as rich text in
                      the_content
    if x ≠ 0 then
        set xx to offset of "For the" in the_-
                            content
        try
            set font of characters x thru (xx
                                  - 3) to
                                  "Cochin"
        end try
        set size of characters x thru (xx - 3)
                            to 20
        set color of characters x thru (xx - 3)
                            to {43 * 256,
                            0, 256 * 256 -
                            1}
    end if
end try
try
    set x to offset of "Daily Average" as rich
                      text in the_content
    if x ≠ 0 then
        set xx to offset of "For the" in the_-
                            content
        try
```

```
            set font of characters x thru (xx - 3) to "Cochin"
        end try
        set size of characters x thru (xx - 3) to 20
        set color of characters x thru (xx - 3) to {43 * 256, 0, 256 * 256 - 1}
    end if
end try
try
    if "Annual" is in the_content and (("(Daily Tallies)" is in the_content or "(Monthly Tallies)" is in the_content)) and ("Monthly Report" is not in the_content and "Daily Report" is not in the_content) then
        set theColorToDraw to {181 * 256, 5 * 256, 143 * 256}
        set x to offset of "Annual" in the_content
        set y to offset of ")" in the_content
        set font of characters x thru y to "Cochin"
        set size of characters x thru y to 20
        set color of characters x thru y to theColorToDraw
    end if
end try
try
    set x to 0
    set xxx to (count of the_content)
    repeat
        set x to x + 1
```

```
if character x of the_content = "•" then
    set xx to 0
    repeat
        set xx to xx + 1
        if character (x + xx) of the_content ≠ "•" then
            set color of characters x through (x + (xx - 1)) to {56342, 2442, 607}
            set x to x + xx
            exit repeat
        end if
    end repeat
end if
```

```
                if character x of the_content = "◆" then
                    set xx to 0
                    repeat
                        set xx to xx + 1
                        if character (x + xx) of the_content ≠ "◆" then
                            set size of characters x through (x + (xx - 1)) to 10
                            set x to x + xx
                            exit repeat
                        end if
                    end repeat
                end if
                if x ≥ xxx then exit repeat
            end repeat
        end try
        set p to 6
    end tell # content
end tell # application "Mail"
#
try
```

```
    set PosixjpgFilePath to ""
    set my posixCompressedFilePath to ""
    if "Hourly" is not in my itemDetails then
        set my posixCompressedFilePath to POSIX path of my zipItFull(my saveTheFilePath)
    end if
end try
try
    tell application "Finder"
        set jpgFilePath to ((path to desktop as text) & "Numbers_Chart.jpg")
        set PosixjpgFilePath to POSIX path of (jpgFilePath as text)
    end tell
end try
set p to 20
# Insert Attachments
tell application "Mail"
    activate
    tell newMessage
        set p to 20.1
        try
            repeat with themailitem in theCCRecipients
                set p to 20.5
                make new to recipient at end of to recipients with properties {address:themailitem}
            end repeat
        end try
        try
            set p to 20.6
            make new attachment with properties {file name:PosixLogoPath as POSIX file} at before first paragraph
        on error errmsg number errnum
            tell application "System Events" to display dialog "eMailIt setting Logo error " & er-
```

```
                                rmsg & " number " & errnum
end try
set cp to count of paragraphs
set p to 20.8
try
    if PosixjpgFilePath ≠ "" then
        if my sendTheChart or ((my displayHourlyJobsFlag) and ("Hourly" is in my itemDetails)) then
            set p to 20.9
            make new attachment with properties {file name:PosixjpgFilePath as POSIX file} at before paragraph (cp - 4)
        end if
    end if
on error errmsg number errnum
    tell application "System Events" to display dialog "eMailIt setting jpg Chart error " & errmsg & " number " & errnum
end try
set p to 21
try
    if posixCompressedFilePath ≠ "" and (my sendTheChart) and theSendCompressedFile = 2 then
```

```
        make new attachment with properties {file name:posixCompressedFilePath as POSIX file} at after last paragraph
    end if
on error errmsg number errnum
    tell application "System Events" to display dialog "eMailIt setting compressed File 1 error " & errmsg & " number " & errnum as text
end try
set p to 22
try
    if includeCompressedExcelSpreadSheet then
        set p to 23
        if my saveTheExcelFilePath ≠ "" and (my sendTheChart) and theSendCompressedFile = 2 then
            make new attachment with properties {file name:(POSIX path of saveTheExcelFilePath) as POSIX file} at after last
```

```
                                paragraph
        end if
    end if
on error errmsg number errnum
    tell application "System Events" to display dialog "eMailIt setting compressed File 2 error " & errmsg & " number " & errnum & " p = " & p & "   includeCompressedExcelSpreadSheet " & (includeCompressedExcelSpreadSheet as text)
end try
set p to 23
try
    do shell script ("sleep 2")
    send
    set messageSent to true
    do shell script ("sleep 0.2")
    try
        tell application "System Events" to tell process "Mail"
            click button "Edit Message" of sheet 1 of window 1
            set messageSent to false
        end tell
    end try
on error errmsg
    tell application "System Events" to display dialog "Sending Mail error " & errmsg giving up after 60
end try
try
    if messageSent then close saving no
end try
```

```
                end tell
            end tell
        on error errmsg number errnum
            tell application "System Events" to display dialog
                                "Second Main eMailIt error " & errmsg & "
                                number " & errnum & " p = " & p
        end try
    end repeat
on error errmsg number errnum
    tell application "System Events" to display dialog "First Main
                 eMailIt error " & errmsg & " number " & errnum & " p =
                 " & p
end try
try
    try
        tell application "Finder" to move file ((path to desktop as
                          text) & "Hourly Tally" & ".numbers") to trash
    end try
    tell application "Numbers"
        try
            close every document saving no
        end try
        do shell script ("sleep 0.2")
        quit saving no
    end tell
    do shell script ("sleep 0.2")
    set p to 23
on error errmsg number errnum
    tell application "System Events" to display dialog "Last Main
                 eMailit error " & errmsg & " number " & errnum & " p =
                 " & p
end try
try
    tell application "Finder"
        move file ((my saveTheFilePath) & ".zip" as text) to trash
    end tell
end try
try
    tell application "Finder"
        try
            move file ((my saveTheExcelFilePath) as text) to trash
        end try
        try
```

```
                close every Finder window
            end try
        end tell
    end try
    if my freeGBytes < 20 then
        do shell script ("killall \"Mail Manager\"")
    end if
end eMailit

on saythetext(theText)
    try
        say theText without waiting until completion
    on error errmsg number errnum
        tell application "System Events" to display dialog "Mailers say-
                     TheText " & errmsg as text giving up after 40
    end try
end saythetext
```

If anyone has suggestions for alterations, or additions, please contact me.

Brian Christmas
ozsanta@gmail.com