



JA Computing

# ASObject 0.5

Revision 3 4/19/09

# Introduction

## What is ASObject?

At its core, ASObject is nothing more than a simple AppleScript file. However, the functions the file contains enable ASObject to be more than just “another AppleScript.” They allow it to become a fully functional AppleScript library. They are what give ASObject, and the applications/scripts which use it, their true power.

## Why AppleScript?

The better question would be, why not AppleScript?

AppleScript is a core language to OS X. All of the applications which run in OS X have some sort of AppleScript support and most have, developer provided, “hooks” which AppleScript can then use to automate/control the application to its and the user’s advantage. As such, creating an AppleScript library completely in AppleScript just made sense. Users who understand AppleScript can understand and use ASObject... no strings attached.

Another reason AppleScript was chosen as the language for ASObject was for the ease of integration with all OS X users. Because of the integration of AppleScript into the OS very little thought had to be put into the questions “What versions of OS X will ASObject be compatible with?” and “What do we do about making ASObject a Universal Binary?” The answers were already there! Apple has already done the work for us.

Thirdly, ASObject is not completely AppleScript. Many of ASObject’s powerful functions come from command line tools built into the UNIX core of OS X. All ASObject does is give an scripeter simple function names to remember to achieve often complex tasks. No knowledge of the UNIX applications is required.

Finally, the reason AppleScript was chosen as the main language for ASObject is because the tools for integrating script files into new AppleScripts or XCode applications is as simple as two or three lines of code. Load the script and you are on your way to a new and better way of AppleScripting.

## The Licensing

ASObject is licensed under the Lesser GPL or LGPL<sup>1</sup>. This licensing entitles you, the new developer, to use ASObject as a library in your own scripts, keeping the ASObject file separate and open source. Your own scripts may be closed source as long as you do not copy and paste code from ASObject into your own script, in which case, both ASObject and your script must remain open source.

---

<sup>1</sup> See <http://www.gnu.org/licenses> for more details on the LGPL

Finally, the LGPL requires you, the developer, to return any code changes you might make to ASObject to the original developer, JA Computing, so that others may benefit from your additions or bug fixes or whatever else you do to ASObject while it is in your hands.

I have included copies of the GPL and LGPL licenses with ASObject for your convenience

## If You Need Help

If you find yourself confused at any time during the reading of this document or while using ASObject please do not hesitate to send an email of your question to [webmaster@jacomputing.net](mailto:webmaster@jacomputing.net). As ASObject nears a more mature state or becomes more popular a wiki or other “help like” system will be setup. As of this moment though, this documentation, along with your own wits and knowledge of AppleScript, is the only help with ASObject that is available.

# How To Use This Document

## Function Definitions

Function Definitions will be laid out in the following manor:

### **FUNCTION\_TITLE(PARAM1,PARAM2)**

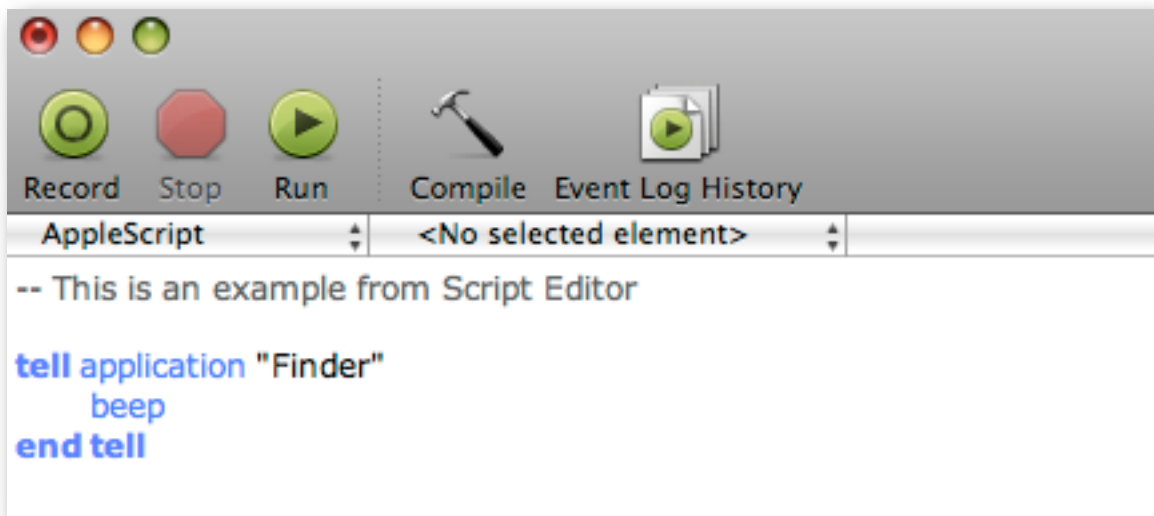
Description	Parameters	Returns	Errors
FUNCTION_TITLE does all kinds of cool stuff...	<ul style="list-style-type: none"> <li>• Param1 - Is stuff you can pass to this function</li> <li>• Param2 - Is more stuff you can pass to this function</li> </ul>	True if no errors	This function will never error.

## Scripting Examples

When a reference is made to AppleScript code it will always be centered and italicized (example below):

*tell application "Finder" to say "Hi"*

When an example of how to use ASObject, or any scripting example for that matter, needs to be displayed it will be shown in the following manor:



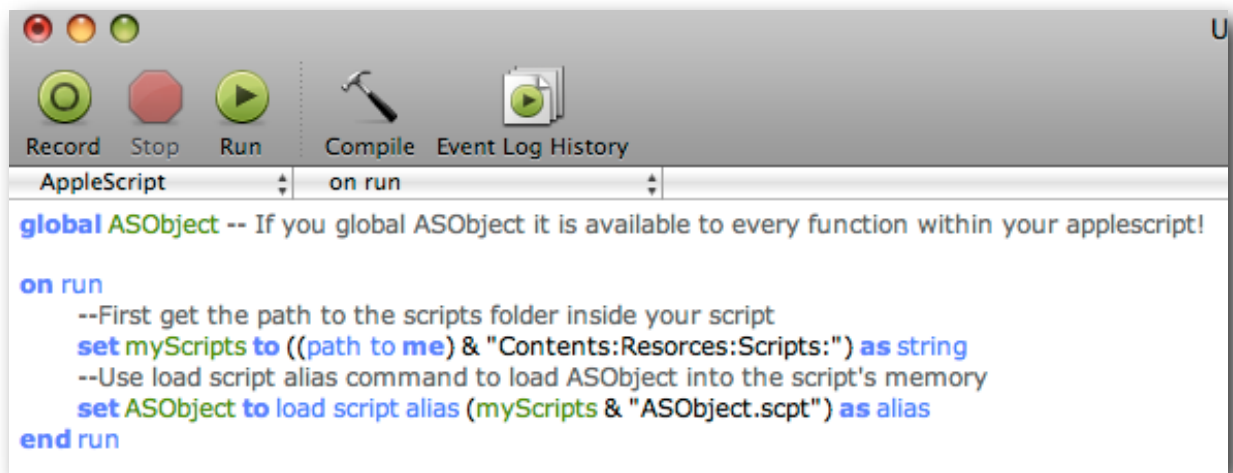
# How To Use ASObject

## Importing ASObject using Script Editor

To import ASObject into an existing AppleScript is as simple as two (or three) lines of code. Insert the following lines of code into your script right after the **on run** line (or near the top of your script if you are not using the **on run** blocks):

```
set myScripts to ((Path to me) & "Contents:Resources:Scripts:") as string
```

```
Set ASObject to load script alias (myScripts & ASObject) as alias
```



Of course, you can name *myScripts* or *ASObject* to any variable name you want. The first line gives you the path to the Scripts folder inside of your script. The second line tells AppleScript to load the contents of another script into memory for this script to use<sup>2</sup>. An example of what the code should look like in the Script Editor application is shown below.

## Saving Your Script

When you save your script out of Script Editor you must save it as an Application Bundle! If you do not there will not be a Scripts folder inside of your script to store the ASObject script file! After you have saved your script as an application bundle, browse to your script in Finder and right-click (control-click) on the script file and choose "Show Package Contents." Then browse to the "Scripts" folder, inside the "Resources" folder, and copy the ASObject script file here with the "main.scpt" file.

<sup>2</sup> For the ASObject script to be available to a script that contains multiple functions already you must globalize the ASObject variable!

Congratulations! You have just installed ASObject into your AppleScript. Now you can use all of its functionality in your own script!

## Calling Functions in ASObject

Calling a function in ASObject from your own script is as easy as making a call to a local function. The exception is that instead of using the call:

*Function\_Name() of me*

or

*tell me to Function\_Name()*

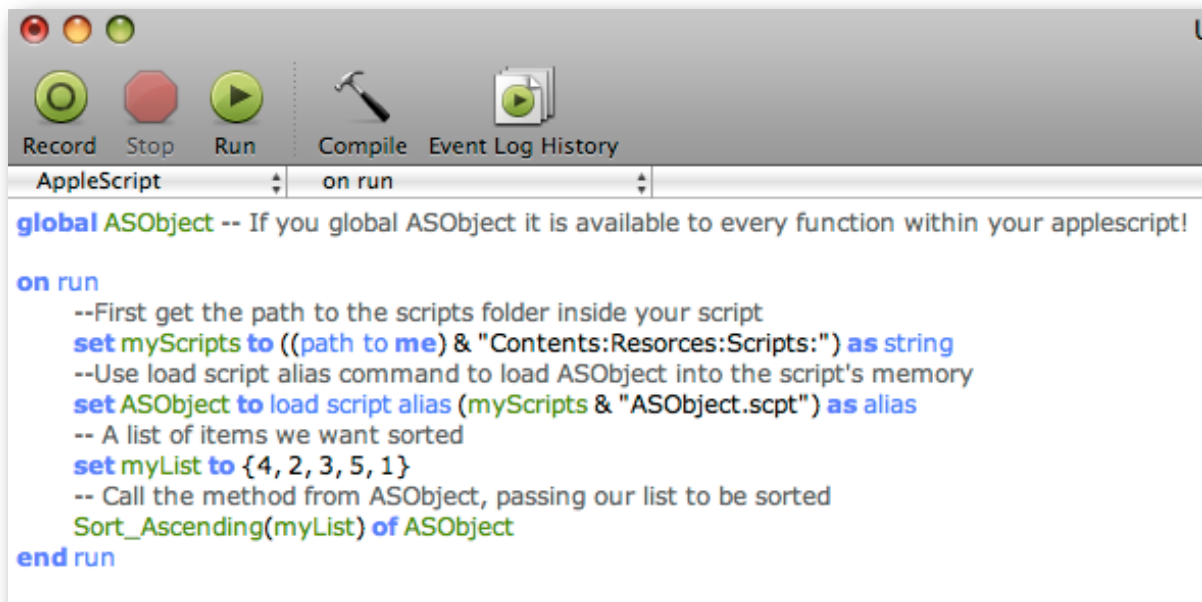
You instead reference the object which you are calling. In this case the object you are calling is the ASObject variable which was initialized early to the contents of the ASObject script. Thus, to make a call to the Sort\_Ascending function in ASObject the code is as simple as:

*Sort\_Ascending(myList) of ASObject*

or

*tell ASObject to Sort\_Ascending(myList)*

And now an example straight from Script Editor:



## Conclusion

And that is it! That is how easy it is to setup your AppleScripts to use ASObject. Build and run your new script and watch ASObject work!

If you find yourself confused at any time during the reading of this document or while using ASObject please do not hesitate to send an email of your question to [webmaster@jacomputing.net](mailto:webmaster@jacomputing.net).

# The Functions

## Countdown\_Dialog(dialogText, totalDelay)

Description	Parameters	Returns	Errors
Countdown_Dialog displays a dialog containing custom text that counts down from a set number of seconds.	<ul style="list-style-type: none"> <li>dialogText - a string describing what you are counting down for</li> <li>totalDelay - the total number of seconds you would like to delay</li> </ul>	<p>True if successful.</p> <p>False if the user canceled</p>	None Reported

## Error\_Dialog(errorTitle, errorMessage, errorType)

Description	Parameters	Returns	Errors
Error_Dialog uses the display alert commands in AppleScript to display an error with a custom title and message. These look similar to the error dialogs used by standard OS X applications.	<ul style="list-style-type: none"> <li>errorTitle - A string, to be displayed in bold, that is the title of your error</li> <li>errorMessage - The sub-message/description of the error.</li> <li>errorType - An integer, either a 0, 1, 2 for critical, informational, or warning errors respectively.</li> </ul>	True if successful	Errors may be returned if the errorType passed is greater than 2.

## Total\_List(thelist)

Description	Parameters	Returns	Errors
Total_List adds up all the values in a list and returns the total.	<ul style="list-style-type: none"> <li>theList - a list of integers/ floating numbers</li> </ul>	A number - the total value of the list passed	Errors may be returned if the variable passed to the function is not a list or does not contain numbers.

## Average\_List(thelist)

Description	Parameters	Returns	Errors
Average_List adds up all the values in a list then finds the average value of the list.	<ul style="list-style-type: none"> <li>theList - a list of integers/ floating numbers</li> </ul>	A number - the average of the list passed	Errors may be returned if the variable passed to the function is not a list or does not contain numbers.

## Sort\_Ascending(thelist)

Description	Parameters	Returns	Errors
Sort_Ascending sorts a list in ascending order (either alphabetical or numerical).	<ul style="list-style-type: none"> <li>theList - a list of numbers or strings</li> </ul>	A list - the list of the now sorted numbers or strings	Errors may be returned if the variable passed to the function is not a list

## Sort\_Descending(thelist)

Description	Parameters	Returns	Errors
Sort_Descending sorts a list in descending order (either alphabetical or numerical).	<ul style="list-style-type: none"> <li>theList - a list of numbers or strings</li> </ul>	A list - the list of the now sorted numbers or strings	Errors may be returned if the variable passed to the function is not a list

## Item\_Number(theValue, thelist)

Description	Parameters	Returns	Errors
Item_Number finds the item number of a value within a list.	<ul style="list-style-type: none"> <li>theList - a list of numbers or strings</li> </ul>	A number. The number reference to the item in the list whose value is equal to the value searched for.	<p>Errors may be returned if the variable passed to the function is not a list.</p> <p>An error may be returned if the value searched for in the given list is not found.</p>

## Write\_To\_Log(logContents, appName)

Description	Parameters	Returns	Errors
Write_To_Log writes strings to a dated log file in a folder which reflects the applications name in the ~/Library/Logs/ folder.	<ul style="list-style-type: none"> <li>logContents - A string to write to the log file</li> <li>appName - The name of the application (used to organize the log files by application name)</li> </ul>	<p>True if writing to log file was successful.</p> <p>False if writing to the log file failed.</p>	None Reported

## Read\_Log\_File(logFile)

Description	Parameters	Returns	Errors
Read_Log_File attempts to load the contents of a file passed to it by using the UNIX command "cat"	<ul style="list-style-type: none"> <li>logFile - the posix path to the log file to be read</li> </ul>	Contents of the log file as a string.	Path to the file is not in posix form or cannot be found.

## Open\_Log\_File(appName)

Description	Parameters	Returns	Errors
Open_Log_File attempts to open an application's most recent log (created using the Write_To_Log function) in the Console Application	<ul style="list-style-type: none"> <li>appName - the name of the application whose log file you want to open</li> </ul>	The name of the log file that was opened.	<p>No log files exist</p> <p>No log folder for this application was found</p>

## Read\_Plist(posixString)

Description	Parameters	Returns	Errors
Read_Plist reads the values from a property list file in OS X, xml or binary, and returns the values in a record.	<ul style="list-style-type: none"> <li>posixString - the path to the plist file in posix form</li> </ul>	The properties of the plist file in record form using pipes " " to as encapsulators for key names	Path to file is not in posix form or the file cannot be found.

## Get\_Plist\_Property(keyName, posixString)

Description	Parameters	Returns	Errors
Get_Plist_Property returns the keys and values of a specified property within a specified plist. This can be helpful when working with a plist containing multiple dictionaries.	<ul style="list-style-type: none"> <li>keyName - the key you want to return the values for.</li> <li>posixString - the path to the plist file in posix form.</li> </ul>	The properties of the key specified in the plist file in record form using pipes " " to as encapsulators for key names	<p>Path to file is not in posix form or the file cannot be found.</p> <p>Key can not be found in plist</p>

## Write\_Plist(keyName, keyValue, posixString)

Description	Parameters	Returns	Errors
Write_Plist writes a value for a defined key in a specified plist file.	<ul style="list-style-type: none"> <li>keyName - the name of the key (a string)</li> <li>keyValue - the new value to change the key to (can be different value types: lists, boolean, strings, integers)</li> <li>posixString - the path to the plist file in posix form</li> </ul>	True if writing of new value was successful	<p>Path to file is not in posix form or the file cannot be found.</p> <p>Key cannot be found in plist</p>

## New\_Plist(posixString)

Description	Parameters	Returns	Errors
New_Plist creates a new blank plist file at a specified location.	<ul style="list-style-type: none"> <li>posixString - the path to the plist file in posix form</li> </ul>	True if writing of new value was successful	Path to file is not in posix form.

## Write\_Key(keyName, defaultValue, posixString)

Description	Parameters	Returns	Errors
Write_Key writes a new key to a given plist using the “defaults write” UNIX command.	<ul style="list-style-type: none"> <li>keyName - the name of the key (a string)</li> <li>defaultValue - the default value to assign to the key.</li> <li>posixString - the path to the plist file in posix form</li> </ul>	True if writing of new key was successful	Path to file is not in posix form or the file cannot be found.

## Delete\_Key(keyName, posixString)

Description	Parameters	Returns	Errors
Delete_Key deletes a key and its value from a specified plist file using the “defaults delete” UNIX command.	<ul style="list-style-type: none"> <li>keyName - the name of the key (a string)</li> <li>posixString - the path to the plist file in posix form</li> </ul>	True if writing of new key was successful	<p>Path to file is not in posix form or the file cannot be found.</p> <p>Key could not be found in plist file</p>

## To\_Apple\_String(posixString)

Description	Parameters	Returns	Errors
To_Apple_String takes a posix string and converts it to an AppleString which can then be used by Finder to reference alias'	<ul style="list-style-type: none"> <li>posixString - the posix path of a file or folder to be converted. This can be either full or relative</li> </ul>	String that is in AppleScript path form.	None reported

## Search\_And\_Replace(someString, firstDelimiter, secondDelimiter)

Description	Parameters	Returns	Errors
Search_And_Replace uses AppleScript's text item delimiters to search for a given string, character, and replace it with another.	<ul style="list-style-type: none"> <li>• someString - A string</li> <li>• firstDelimiter - A string or character to remove from someString.</li> <li>• secondDelimiter - A string or character to replace in someString.</li> </ul>	The new string	None reported

## Reverse\_String(someString)<sup>3</sup>

Description	Parameters	Returns	Errors
Reverse_String reverses the order of a given string	<ul style="list-style-type: none"> <li>• someString - A string</li> </ul>	The new reversed string	None reported

## Split(someString, theDelimiter)

Description	Parameters	Returns	Errors
Split take a given string and creates a list based on a specified delimiter using AppleScript's text item delimiters.	<ul style="list-style-type: none"> <li>• someString - A string to be split</li> <li>• theDelimiter - A string or character to delimit items of someString</li> </ul>	A list	None reported

## Split\_Every(numChar, someString)

Description	Parameters	Returns	Errors
Split_Every splits a string into a list using a specified number of characters for each list item.	<ul style="list-style-type: none"> <li>• numChar - The number of characters each list item should contain.</li> <li>• someString - A string to be split</li> </ul>	A list	None reported

<sup>3</sup>The Reverse\_String method in the process of being discontinued as AppleScript has a single command to perform this action built in.

## Join(someList, theDelimiter)

Description	Parameters	Returns	Errors
Join uses AppleScript's text item delimiters to join a list of numbers or strings into a string using a specified delimiter	<ul style="list-style-type: none"> <li>someList - A list of numbers or strings to be joined</li> <li>theDelimiter - The character or string to join the items using</li> </ul>	A string	None reported

## trim(someString)

Description	Parameters	Returns	Errors
Trim removes whitespace from the beginning and end of a specified string	<ul style="list-style-type: none"> <li>someString - A string to trim the whitespace from</li> </ul>	A string	None reported

## Is\_POSIX(someString)

Description	Parameters	Returns	Errors
Is_POSIX takes a string and attempts to verify that it is in posix form by performing some various UNIX commands on it.	<ul style="list-style-type: none"> <li>someString - A string</li> </ul>	<p>True if string is verified as a posix string</p> <p>String passed if not found to be posix string</p>	String is in quoted form.

## Modulus\_10(theNum)

Description	Parameters	Returns	Errors
Modulus_10 performs a Luhn Algorithm check on a specified number (Passed as a string)	<ul style="list-style-type: none"> <li>theNum - a number, passed as a string</li> </ul>	<p>True if number passes Luhn Algorithm</p> <p>False if the number does not pass the Luhn Algorithm</p>	None reported

## Generate\_Modulus\_10(numLength)

Description	Parameters	Returns	Errors
Generate_Modulus_10 generates a number of a specific length that will pass the Luhn Algorithm	<ul style="list-style-type: none"> <li>numLength - the length of the number you want generated</li> </ul>	A string of the number of specified length	None reported

## Congruence(numA, numB, numN)

Description	Parameters	Returns	Errors
Congruence performs the mathematical equation "a is congruent to b modulus n"	<ul style="list-style-type: none"> <li>numA - the "a" value in the equation</li> <li>numB - the "b" value in the equation</li> <li>numN - the "n" value in the equation</li> </ul>	<p>True if mathematical equation returns true</p> <p>False if mathematical equation returns false</p>	None reported

## Curl\_Download(argString) --local <localpath> --remote <remotefile>

Description	Parameters	Returns	Errors
Curl_Download uses the UNIX command "curl" to download files from http or ftp servers.	<ul style="list-style-type: none"> <li>argString - a string containing the --local flag for where the output file should be placed and the --remote flag for where the remote file is located (as a fully qualified path, including http:// or ftp://)</li> </ul>	Posix path to the downloaded file if download is successful	Returns dynamic errors from UNIX commands.

## Curl\_Post(fileLocation, uploadLocation)

Description	Parameters	Returns	Errors
Curl_Post uses the UNIX command “curl” to post files to an http or ftp server.	<ul style="list-style-type: none"> <li>fileLocation - the local path to the file you want to upload (in posix form)</li> <li>uploadLocation - the location where you want to upload the file to (must be fully qualified with http:// or ftp://)</li> </ul>	True if successful	Returns dynamic errors from UNIX commands.

## Get\_Record\_Properties(theRecord)

Description	Parameters	Returns	Errors
Get_Record_Properties gets the names of the properties of an AppleScript record	<ul style="list-style-type: none"> <li>theRecord - An AppleScript record</li> </ul>	A list of the record property names	Returns dynamic errors

## MD5\_Hash(posixPath)

Description	Parameters	Returns	Errors
MD5_Hash uses the UNIX command “md5” to create an MD5 hash of the file.	<ul style="list-style-type: none"> <li>posixPath - the posix path to the file you want to generate an md5 hash from</li> </ul>	MD5 hash as string	File does not exist

## MD5\_String(theString)

Description	Parameters	Returns	Errors
MD5_String uses the UNIX command “md5” to generate an md5 hash of a string	<ul style="list-style-type: none"> <li>theString - a string to generate an md5 hash of.</li> </ul>	MD5 hash as string	None reported

## Compare\_Files(file1, file2)

Description	Parameters	Returns	Errors
Compare_Files uses MD5 hashes to compare two files	<ul style="list-style-type: none"> <li>file1 - the posix path to a file</li> <li>file2 - the posix path to a file</li> </ul>	True if files are the same  False if files are different	File does not exist

## Compare\_MD5(FilePath, md5String)

Description	Parameters	Returns	Errors
Compare_MD5 compares the md5 hash of a file to a specified md5 string	<ul style="list-style-type: none"> <li>FilePath - the posix path to the file to be tested</li> <li>md5String - the md5 hash to compare the file to.</li> </ul>	True if files are the same  False if files are different	File does not exist

## Is\_Running(appName)

Description	Parameters	Returns	Errors
Is_Running uses the UNIX command "ps" to find out if a given process is running	<ul style="list-style-type: none"> <li>appName - the name of the application to search for</li> </ul>	True if app is running  False if app is not running	None reported

## Kill\_Process(appName, doRepeat)

Description	Parameters	Returns	Errors
Kill_Process uses the UNIX command "kill" to attempt to kill a given application	<ul style="list-style-type: none"> <li>appName - the name of the application to kill</li> <li>doRepeat - a boolean value to run a check to ensure that the application has been quit</li> </ul>	0 (zero) if kill -3 was used  1 if kill -9 was used	The specified process is not currently running.

## Get\_Proxy\_Settings(proxyType)

Description	Parameters	Returns	Errors
Get_Proxy_Settings uses the UNIX command "networksetup" to get the proxy settings for a given connection type on the first available (and active) network interface.	<ul style="list-style-type: none"> <li>proxyType - the type of proxy to get info on (webproxy, securewebproxy, ftpproxy)</li> </ul>	<p>A string of the proxy and port separated by a colon ":".</p> <p>A blank string if no proxies are specified</p>	Proxy type is not valid.

## Get\_OS\_Version()

Description	Parameters	Returns	Errors
Get_OS_Version uses the command line utility sw_vers to get the the current running version of OS X.		A string of the version of OS X.	None

## Parse\_Arguments(argString)

Description	Parameters	Returns	Errors
Parse_Arguments parses a given string for flags, delimited by a double-dash "--" and then values following the flag.	<ul style="list-style-type: none"> <li>argString - a string containing flags and values</li> </ul>	A record containing a list of the flag names (called argFlags) and values for flags of same location in argFlags list (list of values is called argValues)	<p>Improper flags</p> <p>No arguments specified</p>

**Check\_For\_Update(argString) --serverURL <softwareupdatefile> --appName <applicationname> --appVersion <appversion><sup>4</sup>**

Description	Parameters	Returns	Errors
<p>Check_For_Update uses other internal ASObject commands to download a text file from a server and parse it for application information and then compares it to known application information to determine if an update is available. If an update is available it will ask if you want to download it from the known location of the update file</p>	<ul style="list-style-type: none"> <li>serverURL - the url of the text file containing the application version and update information</li> <li>appName - the name of the application you are looking to update</li> <li>appVersion - the version of the application you are looking to update</li> </ul>	<p>The posix location of the downloaded update (if one is available)</p> <p>“User canceled update”</p> <p>“No update Available”</p>	<p>“No software update record was found for this application.”</p> <p>Curl_Download Errors</p> <p>Argument missing or value missing.</p> <p>Unable to communicate with update server!</p>

**mail(messageRecipient, subjectLine, messageText)<sup>5</sup>**

Description	Parameters	Returns	Errors
<p>Mail uses the UNIX command “mail” to attempt to send an email to a recipient.</p>	<ul style="list-style-type: none"> <li>messageRecipient - the email address of the person you want to email (a string)</li> <li>subjectLine - the subject of the message (a string)</li> <li>messageText - the message to email (a string)</li> </ul>	<p>“Mail sent to “ specified recipient</p>	<p>See footnote!</p>

<sup>4</sup> Version 0.5 of ASObject changed the format of the text file for software updates to a plist file. This will make it easier to make changes/additions to this method in the future. Download the example plist to see the formatting of this file.

<sup>5</sup> The mail function in ASObject has known bugs where it does not always deliver the message if it cannot find a mail server from DNS.

**mail\_sendmail(argString) --msgTo <emailAddress> --msgFrom <emailAddress> --msgReplyTo <emailAddress> --msgSubject <string> --msgText <string><sup>6</sup>**

Description	Parameters	Returns	Errors
Mail_Sendmail uses the UNIX command "sendmail" to send email.	<ul style="list-style-type: none"> <li>• msgTo - the email address of the person(s) you want to email (separated by commas)</li> <li>• msgFrom - the address you want the email to be from</li> <li>• msgReplyTo - the address you would like emails to reply to (optional)</li> <li>• msgSubject - the subject of the message</li> <li>• msgText - the message to email</li> </ul>	"Sendmail sent to " specified recipient(s)	See Footnote!  Argument flag and value errors

---

<sup>6</sup> The mail\_Sendmail function is in need of refinement for specifying an email server to use. If one is not specified by DNS, mail may fail to deliver.

# DB Event Functions

The DB even functions in ASObject were designed to function with the Database Events application in OS X. This application allows for simple SQL Lite databases to be used with AppleScript and other XCode applications. The DB events in ASObject attempt to make the creation and administration of SQL Lite databases in AppleScript easier than ever.<sup>7</sup>

---

<sup>7</sup> While the DB event functions are currently a part of ASObject they may be split into their own DBOject project in order to become more inclusive and better suited for scripting SQL Lite in OS X.

## DB\_Create(databaseName, databaseLocation)

Description	Parameters	Returns	Errors
DB_Create creates a new SQL Lite Database at a specified location.	<ul style="list-style-type: none"> <li>databaseName - the name of the database</li> <li>databaseLocation - the location of the new database in posix form.</li> </ul>	Full path to the Database including the name and extension for SQL Lite databases.	None Reported

## DB\_Open(databaseLocation)

Description	Parameters	Returns	Errors
DB_Open opens a SQL Lite database for reading or writing (automatically called in most cases.)	<ul style="list-style-type: none"> <li>databaseLocation - posix path to the database (including the extension)</li> </ul>	Name of the database that was opened	<p>Database does not exist.</p> <p>Path does not contain name extension.</p>

## DB\_Close(databaseName, bInSave)

Description	Parameters	Returns	Errors
DB_Close closes a specified database with or without saving.	<ul style="list-style-type: none"> <li>databaseName - the name of the database to close</li> <li>bInSave - boolean value to save the database</li> </ul>	TRUE	Database is not open

## DB\_Count\_Databases()

Description	Parameters	Returns	Errors
DB_Count_Databases counts the number of open databases		Number of open databases	None Reported

## **DB\_Insert(argString, databaseLocation) --record <recordName> --field <fieldName> --value <fieldvalue>**

Description	Parameters	Returns	Errors
DB_Insert creates new records and fields, inserts fields into existing records, or changes the value of existing fields in existing records. Saves database when finished.	<ul style="list-style-type: none"> <li>• argString - string of arguments that will either insert new records, fields, or values depending on the specified string.</li> <li>• databaseLocation - the posix path to the database (including the name extension)</li> </ul>	Properties of new field, record, or value.	None Reported

## **DB\_Delete(recordName, fieldName, databaseLocation)**

Description	Parameters	Returns	Errors
DB_Delete deletes fields from given records.	<ul style="list-style-type: none"> <li>• recordName - the name of the record (CANNOT BE THE ID)</li> <li>• fieldName - the name of the field</li> <li>• databaseLocation - the posix path to the database (including the name extension)</li> </ul>	TRUE	The record specified does not exist.

## DB\_Select(argString, databaseLocation) --record <recordname> --field <fieldname>

Description	Parameters	Returns	Errors
DB_Select selects fields or records based on name or every record or field	<ul style="list-style-type: none"> <li>argString - arguments containing the name of the record (empty for wildcard) and the name of the field (empty for wildcard)</li> <li>databaseLocation - the posix path to the database (including the name extension)</li> </ul>	Every record  Value of every field of specified record  Name of every record  Value of specified field of specified record  name of every field	Arguments not specified.  You must select from a record or a field.

## DB\_Add\_Record(recordName, databaseLocation)

Description	Parameters	Returns	Errors
DB_Add_Record adds a specified record to a given database (no fields created)	<ul style="list-style-type: none"> <li>recordName - The name of the new record</li> <li>databaseLocation - the posix path to the database (including the name extension)</li> </ul>	Properties of new record	Record Exists

## DB\_Remove\_Record(recordName, databaseLocation)

Description	Parameters	Returns	Errors
DB_Remove_Record removes a record and all attached fields	<ul style="list-style-type: none"> <li>recordName - The name of the new record</li> <li>databaseLocation - the posix path to the database (including the name extension)</li> </ul>	TRUE	None reported

**DB\_Get\_Record\_ID(recordName, databaseLocation)**

Description	Parameters	Returns	Errors
DB_Get_Record_ID gets the id number of the first record whose name is specified in a given database	<ul style="list-style-type: none"> <li>recordName - The name of the new record</li> <li>databaseLocation - the posix path to the database (including the name extension)</li> </ul>	ID of record as number	None reported

**DB\_Get\_Name(databaseLocation)**

Description	Parameters	Returns	Errors
DB_Get_Name gets the name of a database given its posix path	<ul style="list-style-type: none"> <li>databaseLocation - the posix path to the database (including the name extension)</li> </ul>	String - the name of the database	None reported