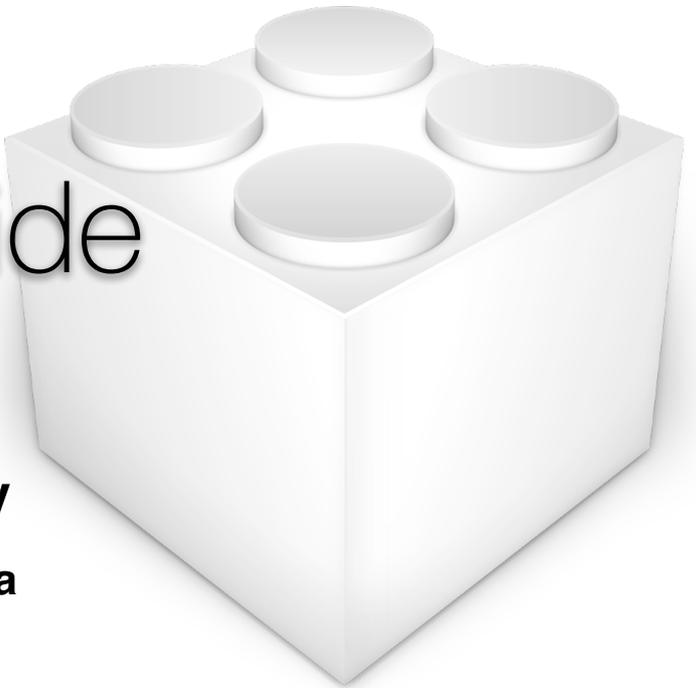


ASObjCExtras.framework Scripting Guide

Framework version 1.2.2 / Document version 1.0

Software Developed By **Shane Stanley**

Document Written By **Takaaki Naganoya**



[About ASObjCExtras.framework / Shane Stanley](#)

[About this "Scripting Guide" / Takaaki Naganoya](#)

[What is this?](#)

[How To Install](#)

[How To Use](#)

[How To Receive error details](#)

[How To Distribute with Applet](#)

[Index](#)

[AppleScript sample-code's syntax color format](#)

[Utility methods](#)

[Class Conversion methods](#)

[Path Manipulation methods](#)

[String Manipulation methods](#)

[List Manipulation methods](#)

[Trigonometry methods](#)

[Misc. methods](#)

— — About ASObjCExtras.framework

AppleScriptObjC relies on the automatic conversion of **text**, **numbers**, **booleans**, **lists** and **records** to and from their Cocoa counterparts. However, there is no conversion of **dates**, **files**, or **data objects**. It also loses precision when converting **floating-point** Cocoa numbers to **reals**. This framework provides these **extra conversions and precision with reals**, as well as some trigonometry functions, several text- and list-handling routines based on those in ASObjC Runner, and a handful of other methods.

You can use the framework by putting it in `~/Library/Frameworks/` or `/Library/Frameworks/`, or by adding it to a script bundle or applet.

Most methods will return **missing value** where there is an error, and if they have an error parameter the returned error will have a description in `localizedError`.

You can use and distribute this framework free of charge, but it must include this header file. There are no guarantees or warranties whatsoever. Use entirely at your own risk. Feedback welcome to [<sstanley@myriad-com.com.au>](mailto:sstanley@myriad-com.com.au).

Shane Stanley

Spread of AppleScriptObjC World and The main target of this scripting guide (blue)

	ASObjC (Xcode)	Cocoa Applet	AppleScript Libraries	ASObjC on Script Editor
OS X ver	Xcode	Script Editor / ASObjC Explorer 4		
10.6	•			
10.7	•	•		
10.8	•	•		
10.9	•	•	•	
10.10	•	•	•	•

Copyright Notice

Mac, OS X, Mac OS X, Siri, Newton, AppleScript and AppleScriptObjC are registered trade mark of US Apple Inc.

UDing and UDing Simulator are registerd trade mark of Toyo Ink SC Holdings.co.,Ltd.

Adobe Photoshop is registered trade mark of US Adobe Software.

ASObjCExtra.framework and ASObjCExplorer 4 are copyrighted software by Shane Stanley.

“AS Hole” (<http://piyocast.com/as>) is an AppleScript blog written by Takaaki Naganoya.

Shane Stanley's AppleScriptObjC Works Mapping



AppleScriptObjC Tools



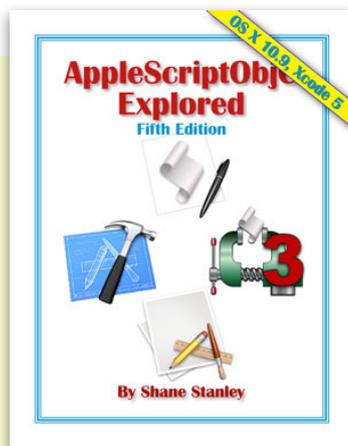
Script Geek

AS/ASOC benchmark speed checker



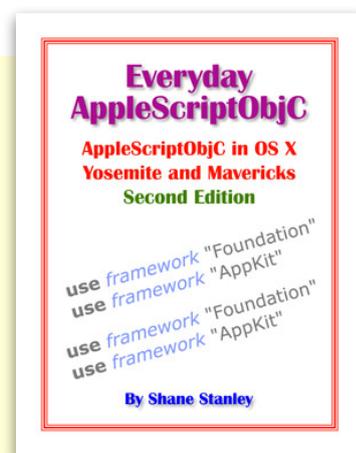
ASObjC Explorer 4

Powerful AppleScriptObjC Editor & Xcode's External editor



AppleScriptObjC Explored

How to develop Gui Based apps with Xcode



Everyday AppleScriptObjC

How to develop Cocoa AppleScript with Script Editor or ASObjC Explorer

AppleScriptObjC e-Books

— — About this “Scripting Guide”

One day, I received a private e-mail from Shane Stanley. He wanted me to evaluate his framework “ASObjCExtras”. All of AppleScripters enjoy his e-book **“AppleScriptObjC Explored”** and **“Everyday AppleScriptObjC”**. His devotion to AppleScript world is huge, important and wonderful (I remember some great people: Bill Cheeseman and Mark Aldritt). Shane has been the only & absolute lighthouse for us for these several years. Who can decline his request? My answer was **“Yes I do”**. There was no choice.

In historical perspective, there were some open member projects in the AppleScript world. I deserted them all. One project did not work on my mother tongue (Japanese) environment, another seemed so charmless. We did not help each other. I thought it was our mistake. It is the time to make action.

ASObjCExtras is a tidy and powerful framework. It made it more safer and faster to use the Cocoa framework. It is very fun to write scripts with ASObjCExtras. Shane’s powerful editor **“ASObjCExplorer 4”**’s cocoa-object/event logging made it easier, too. **AppleScript gets 10-200 times faster** speed by using the Cocoa framework with keeping its simplicity and easiness. ASObjCExtras includes many useful methods for replacing frequently used AppleScript routines. You can replace pure-AppleScript-based sort routines with ASObjCExtras-based high-speed versions. Many heavily looped scripts will be shortened by ASObjCExtras and shorten their processing period dramatically.

When I wrote a lot of AppleScript with earlier versions of ASObjCExtras, sometimes I faced the lack of a scripting guide. So, I wrote this document in my free time. That’s all. I made many questions to Shane and received his great advice. But all of this document and sample scripts are written and checked by me. If this document is wrong, it is not Shane’s fault. It is mine. All of the scripts are checked with ASObjCExplorer 4 and Apple’s Script Editor on OS X, Yosemite. Welcome your comment or feedback to me (<mailto:maro@piyocast.com>). Day by day, I write AppleScript on my blog (<http://piyocast.com/as/>). So, you can see the latest samples there.

Takaaki Naganoya
(Project Manager, Programmer, Planner, Writer and Editor)

— My Works

My earlier and most important project was “**Newt On**” / “**Newt Off**” (2002). This full AppleScript-written program is a small “Knowledge Navigator”. It understands and execute natural language (Japanese). Similar to Apple’s ancient PDA “Newton”’s assistant function. Voice control version existed named “**Kotodama**” (2003). Over 10 years ago, I made a kind of Apple’s “Siri” by only AppleScript. These technologies are put into my later works.



I wrote long and many AppleScript routines in “Newt On” and “Kotodama” project. It was too hard to maintain them all by myself. So, I opened them on my AppleScript Blog “AS Hole” (2008-Today). Today, this site is listed in the first page of Google’s search engine result with keyword “AppleScript” and gains over 1,000,000 page views/year.

TOYO INK, the second largest ink maker in the world has been releasing “UDing” series softwares. I’m an external project manager and programmer in this project. “UDing Simulator” software is made for color blind simulation. It is written in Objective-C++ and AppleScriptObjC; controls Adobe Photoshop and keep correct ICC profiles. This project is evaluated very well in Japan.



— My Books (★joint work)



LEFT: “The Road to REALbasic Master”, Softbank Creative, 1999★

CENTER: “AppleScript Reference”, Softbank Creative, 1999★

RIGHT: “The Road to Mac Master”, Softbank Creative, 2001

What is this?

(1)ASObjCEXtras (&AppleScriptObjC) brings **High-speed data processing** ability to AppleScripters



1D List Uniquefy

Shorter Bar is Faster

{1,2,2,2,2} → {1,2} 100,000 items

■ AppleScript ■ ASOC with ASObjCEXtras

First Run

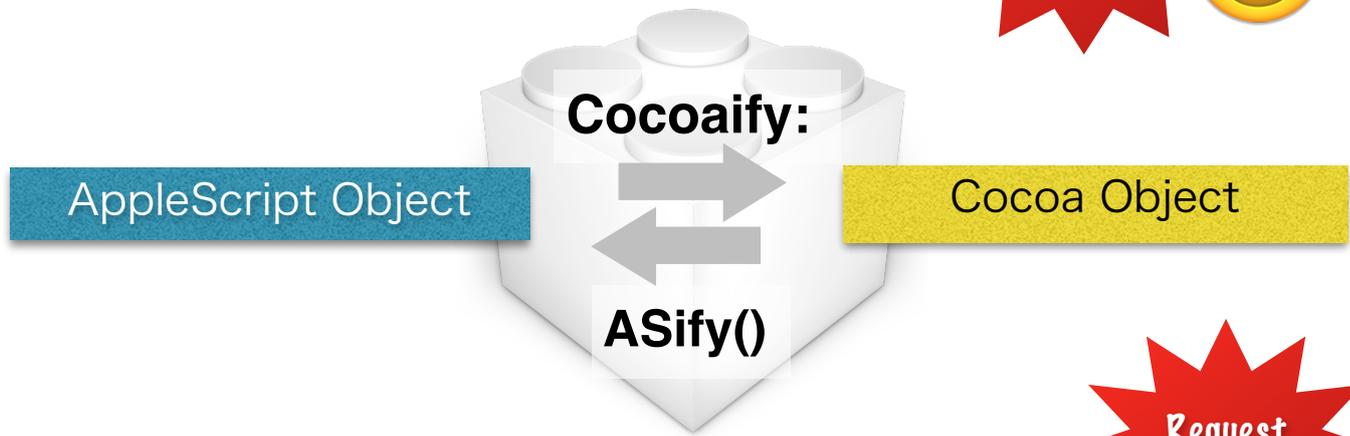


Average



0 1.5 3 4.5 6 Seconds

(2)ASObjCEXtras provides the way to avoid Scripting Bridge's **casting bug**



(3)ASObjCEXtras reflects the skillful scripters' (= your) requests



ASObjCEExtras.framework How To Install

<http://www.macosxautomation.com/applescript/apps/ASObjCEExtras.html>

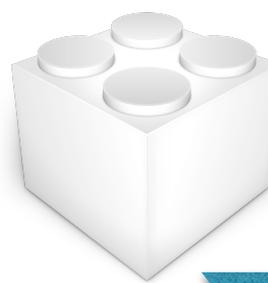


The fourth group provide basic **trigonometry and log** facilities.

The final **miscellaneous group** provides methods for efficiently retrieving file information, and for extracting metadata from image files.

You can [download the latest version of the framework \(1.1.0\)](#), along with [some samples and documentation](#), [here](#).

You can use and distribute this framework free of charge. There are no guarantees or warranties whatsoever. Use at your own risk.



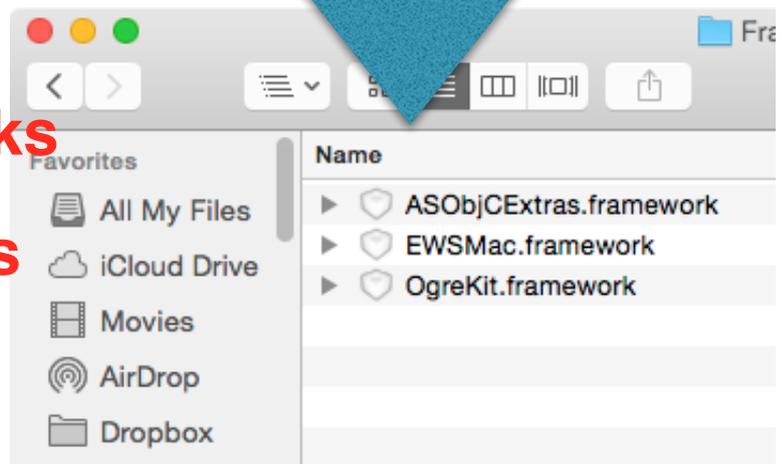
Install

Put “ASObjCEExtras.framework” to

~/Library/Frameworks

or

/Library/Frameworks



You can freely re-distribute and code-signing your script with ASObjCEExtras framework (so, this framework is **not** code-signed).



OS X Yosemite

ASObjCEExtras.framework supports with:

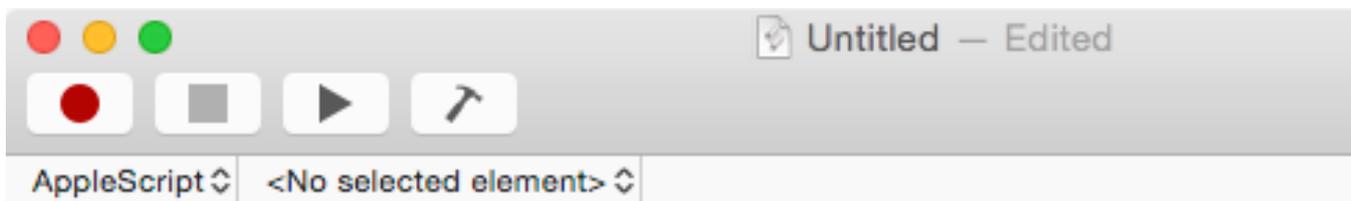
OS X 10.10 or later

ASObjCEExtras.framework

How To Use

Just write: `use framework "ASObjCEExtras"`

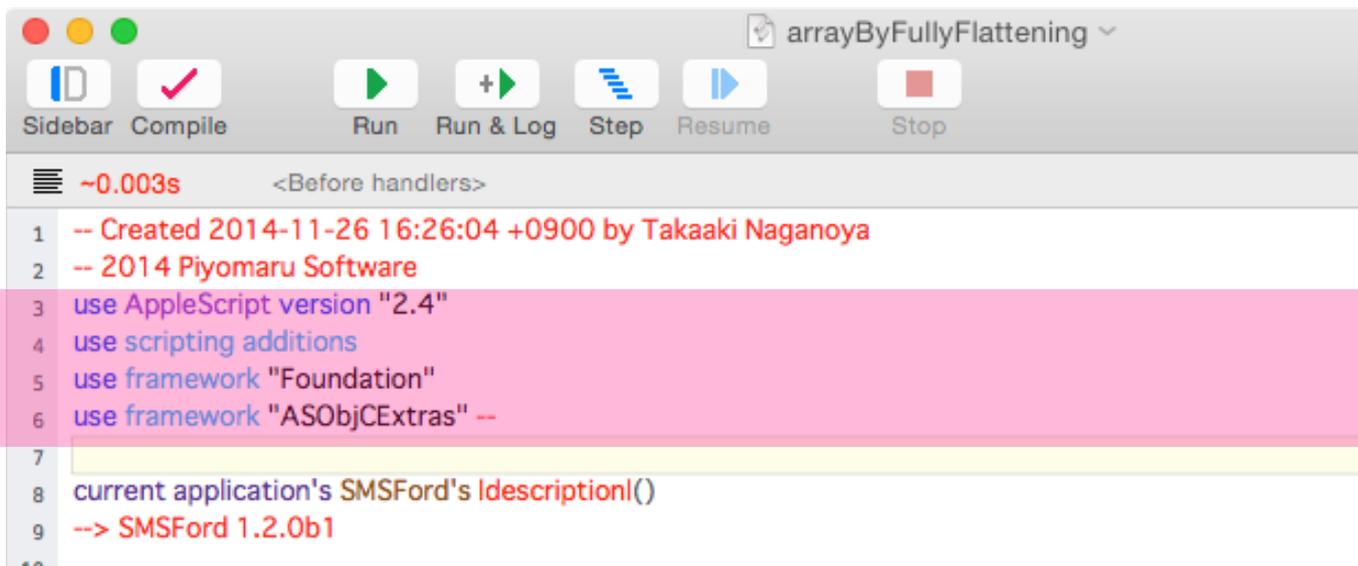
Script Editor



```
use AppleScript version "2.4"  
use scripting additions  
use framework "Foundation"  
use framework "ASObjCEExtras"
```

```
set theVersionInfo to current application's SMSFord's ldescription() as text|
```

ASObjCEXplorer 4



ASObjCEExtras.framework

How To Receive error details

Where a method has an "error" parameter, it should say that it takes **missing value** *or* **reference**.

And if you use Shane Stanley's **ASObjCEXplorer 4**, you can check the error details directly.

Methods can return error (Example. Not all)

subarraysIn: paddedWith: error:
subarraysFrom: groupedBy: error:
subarraysIn: withItems: insertedAtIndex: error:
arrayByMovingItemAt: toIndex: inArray: error:
sumMaxMinOf: error:
subarraysIn: sortedByIndexes: ascending: sortTypes: error:

ASObjCEXplorer 4

① Select "Log" tab

② click "Run & Log"

The screenshot shows the ASObjCEXplorer 4 interface. At the top, there are buttons for 'Sidebar', 'Compile', 'Run', 'Run & Log', 'Step', 'Resume', and 'Stop'. The 'Run & Log' button is highlighted with a pink box. Below the buttons, the script editor shows the following code:

```
1 --By Shane Stanley
2 use AppleScript version "2.4"
3 use scripting additions
4 use framework "Foundation"
5 use framework "ASObjCEExtras"
6
7 set listOfLists to {{1, 2, 3, 4}, {11, 22, 33}}
8 set aRes to current application's SMSFord's colsToRowsIn:listOfLists error:(missing value)
9 set {theArray, theError} to current application's SMSFord's colsToRowsIn:listOfLists error:(reference)
10
```

The 'Log' tab is selected, showing the following log entries:

```
--> {{1, 2, 3, 4}, {11, 22, 33}}
16:13:04.262 [8] set aRes to current application's SMSFord's colsToRowsIn:listOfLists error:(missing value)
--> missing value
16:13:04.263 [9] set {theArray, theError} to current application's SMSFord's colsToRowsIn:listOfLists error:(reference)
--> {missing value, (NSError) Error Domain=SMSErrorDomain Code=-10000 "Can't perform cols to rows when lists have differing item counts." UserInfo=0x618000468f00 {NSLocalizedString=Can't perform cols to rows when lists have differing item counts.}}
```

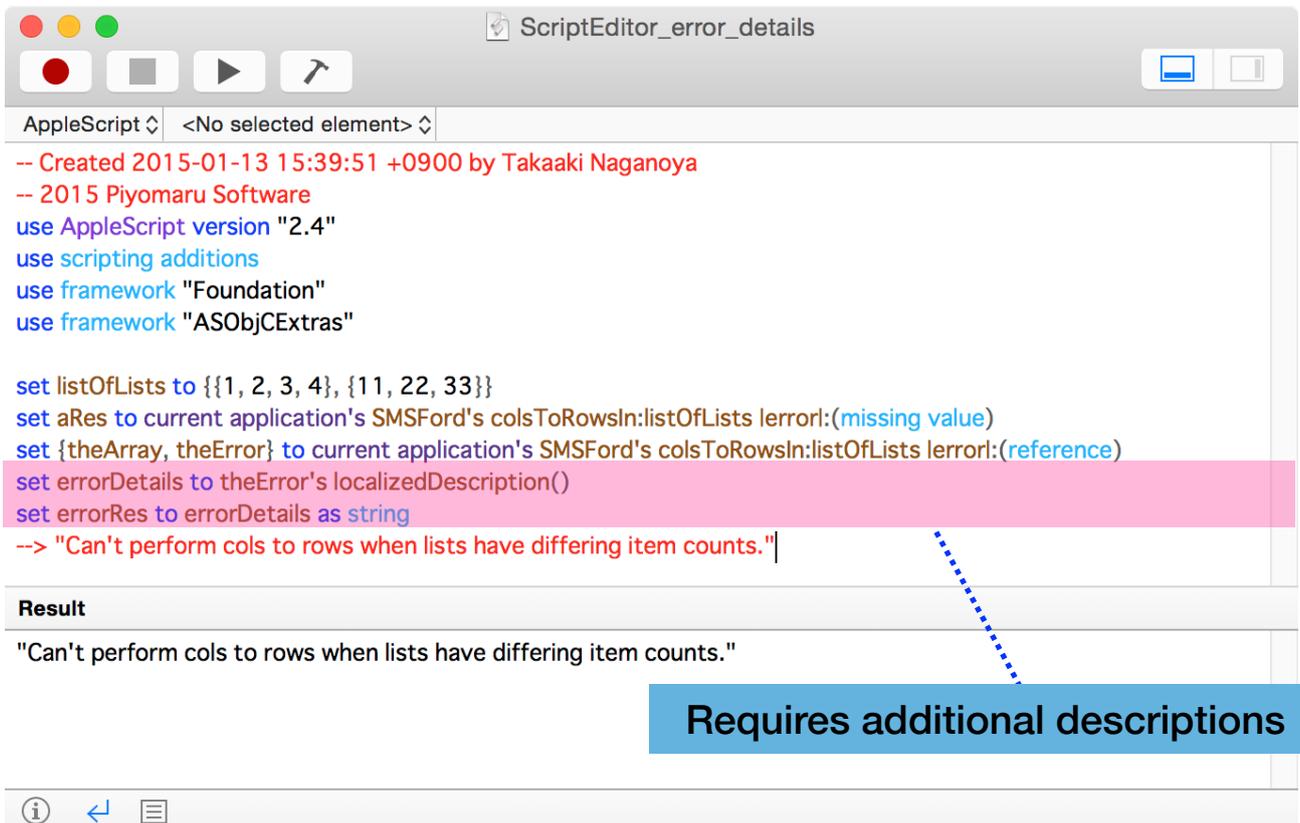
The error details are highlighted in a pink box. Below the log entries, the 'Result of Main Script:' is shown:

```
--> {missing value, (NSError) Error Domain=SMSErrorDomain Code=-10000 "Can't perform cols to rows when lists have differing item counts." UserInfo=0x618000468f00 {NSLocalizedString=Can't perform cols to rows when lists have differing item counts.}}
```

At the bottom, there is a checkbox for 'Run in foreground'.

If you use **Script Editor** to write ASOC scripts, you can get error details with some additional descriptions.

Script Editor



The screenshot shows a window titled "ScriptEditor_error_details" with a menu bar and toolbar. The script content is as follows:

```
-- Created 2015-01-13 15:39:51 +0900 by Takaaki Naganoya
-- 2015 Piyomaru Software
use AppleScript version "2.4"
use scripting additions
use framework "Foundation"
use framework "ASObjCExtras"

set listOfLists to {{1, 2, 3, 4}, {11, 22, 33}}
set aRes to current application's SMSFord's colsToRowsIn:listOfLists lerror:(missing value)
set {theArray, theError} to current application's SMSFord's colsToRowsIn:listOfLists lerror:(reference)
set errorDetails to theError's localizedDescription()
set errorRes to errorDetails as string
--> "Can't perform cols to rows when lists have differing item counts."
```

The error message is displayed in the "Result" pane:

```
"Can't perform cols to rows when lists have differing item counts."
```

A blue callout box with the text "Requires additional descriptions" is connected to the error message by a dotted line.

```
-- Created 2015-01-13 by Takaaki Naganoya
-- 2015 Piyomaru Software
use AppleScript version "2.4"
use scripting additions
use framework "Foundation"
use framework "ASObjCExtras"

set listOfLists to {{1, 2, 3, 4}, {11, 22, 33}}
set aRes to current application's SMSFord's colsToRowsIn:listOfLists lerror:
    (missing value)
set {theArray, theError} to current application's SMSFord's
    colsToRowsIn:listOfLists lerror:(reference)
set errorDetails to theError's localizedDescription()
set errorRes to errorDetails as string
--> "Can't perform cols to rows when lists have differing item counts."
```

ASObjCEExtras.framework

How To Distribute with Applet

You can contain & re-distribute ASObjCEExtras.framework within AppleScript applet's bundle

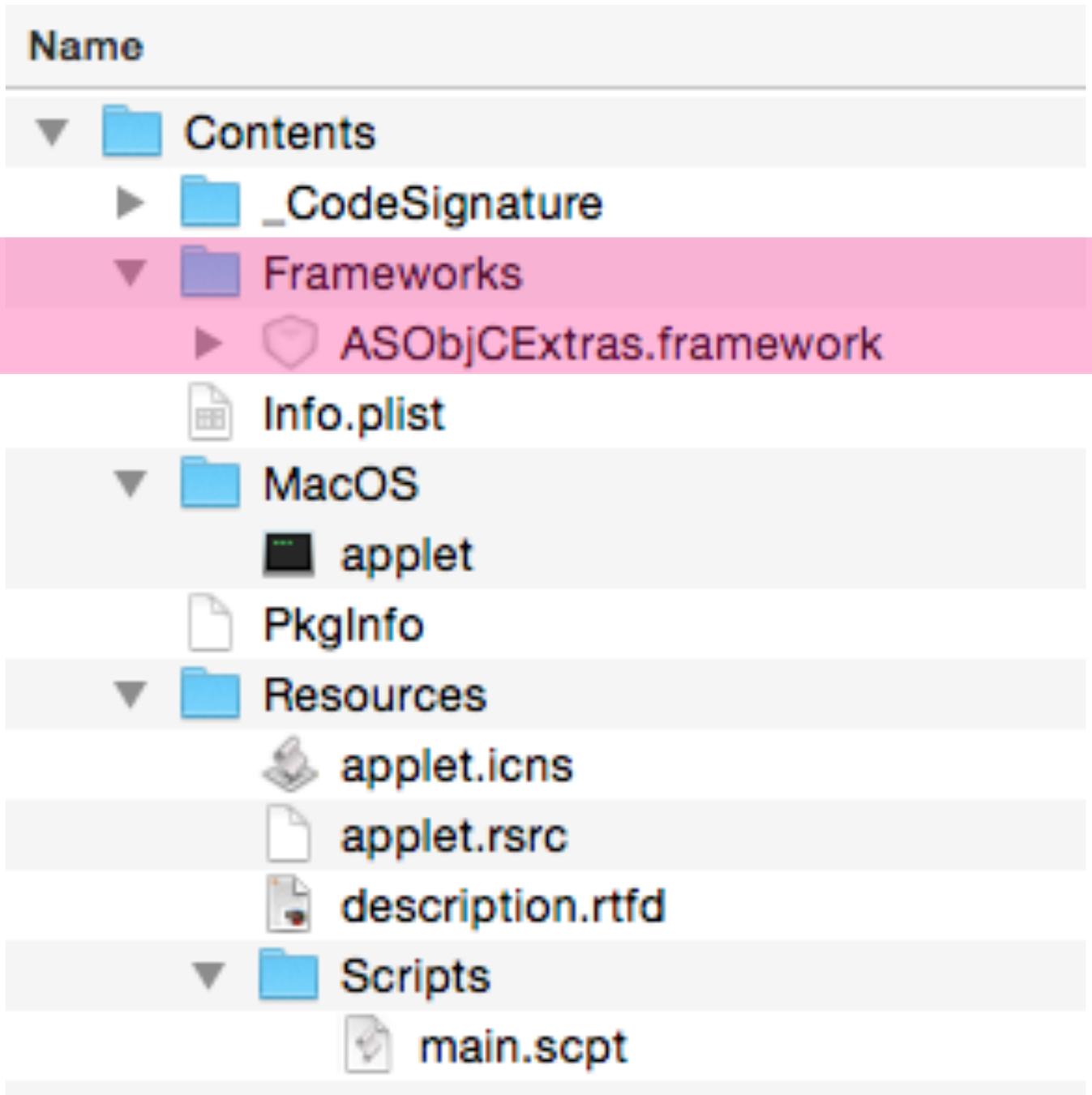


Table Of Contents:

Kind	Method	Appearance version	Dim
Other	description()		
	openHeaderFile()		
Class Conversion	ASify()		
	ASifyInList()	1.2	
	ASifyFor()	1.2	
	ASifyListFor()	1.2	
	Cocoaify:		
	Cocoaify: forTypes:	1.2	
Path Manipulation	URLFrom:		
	fileFromURL:		
	HFSPathFromURL:		
	HFSPathFromURL: colonForPackages:	1.2	
String Manipulation	datesFromStrings: format:		1D→0D 1D→1D
	stringFrom: ICUTransform: inverse:		
	stringFrom: makingIt:		
	arrayFromTSV:		→2D
	arrayFromCSV: commals:		→2D
List Manipulation	subarraysIn: paddedWith: error:		2D
	colsToRowsIn: error:		2D
	arrayByFlattening:		2D→1D
	arrayByFullyFlattening:	1.2	nD→1D

Table Of Contents:

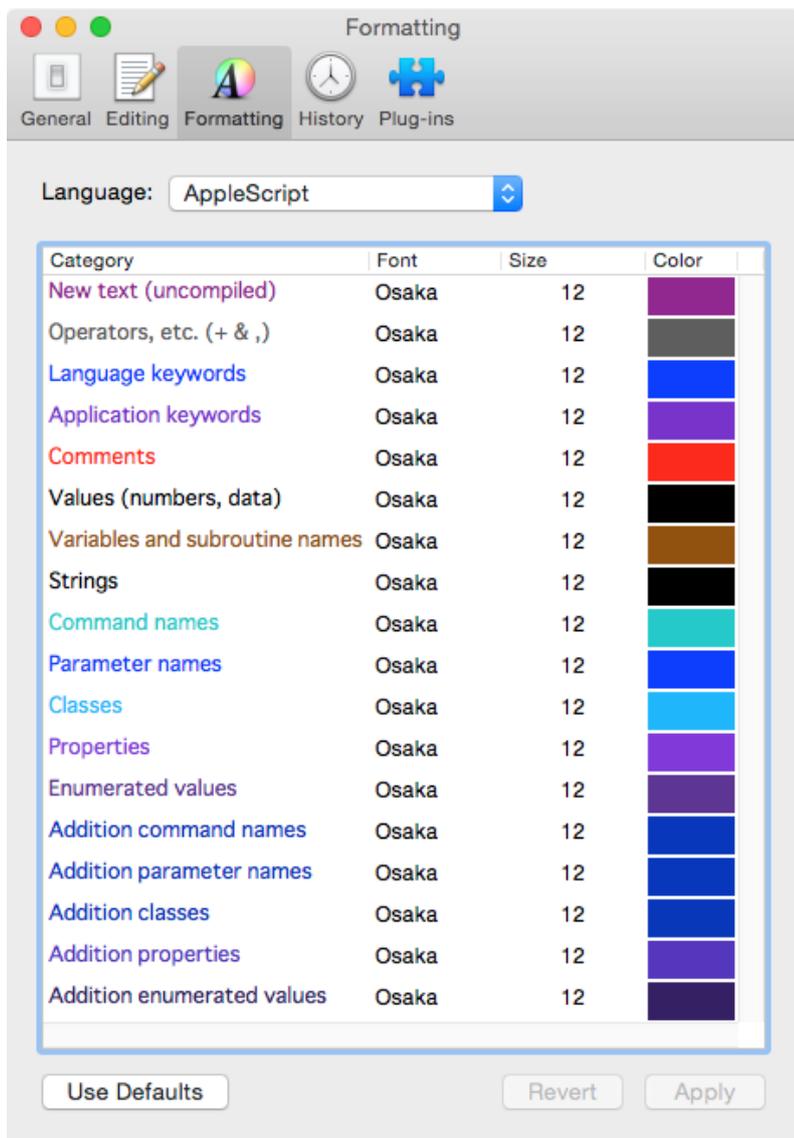
Kind	Method	Appearance version	Dim
List Manipulation	subarraysFrom: groupedBy: error:		1D→2D
	subarraysIn: withItems: insertedAtIndex: error:		2D
	arrayByInsertingItems: inArray: atIndex: error:		
	arrayByMovingItemAt: toIndex: inArray: error:		1D,2D
	arrayByDeletingBlanksIn:		1D
	arrayByTrimmingTrailingBlanksFrom:		1D
	arrayByTrimmingBlanksFrom:		1D
	arrayByReplacingNullsIn: withItem:		1D
	arrayWithPattern: startNumber: endNumber: minDigits:	1.2	1D
	sumMaxMinOf: error:		1D
	subarraysIn: sortedByIndexes: ascending: sortTypes: error:		2D
	subarraysIn: sortedByIndexes: ascending: sortTypes: sortKeys: error:	1.2	2D→2D
	arrayByMergingTextAtIndexes: inArray: inserting: error:		1D
	indexesOfItem: inArray: inverting:		1D
	indexesOfItems: inArray: inverting:		
	subarraysIn: asDictionariesUsingLabels: error:		

Table Of Contents:

Kind	Method	Appearance version	Dim
List Manipulation	subarraysFrom: usingKeys: outKeys: error:	1.2	2D/1D
	arrayWithIndexSet:		
	arrayByAddingInteger: inArray:	1.2	1D
Trigonometry	tanValueOf:		0D/1D
	sinValueOf:		0D/1D
	cosValueOf:		0D/1D
	atanValueOf:		0D/1D
	asinValueOf:		0D/1D
	acosValueOf:		0D/1D
	tanhValueOf:		0D/1D
	sinhValueOf:		0D/1D
	coshValueOf:		0D/1D
	atanhValueOf:		0D/1D
	asinhValueOf:		0D/1D
	acoshValueOf:		0D/1D
	logValueOf:		0D/1D
	log10ValueOf:		0D/1D
Misc.	metadataFromImage: error:		
	infoForFile:		
	sizeInfoForFile:		

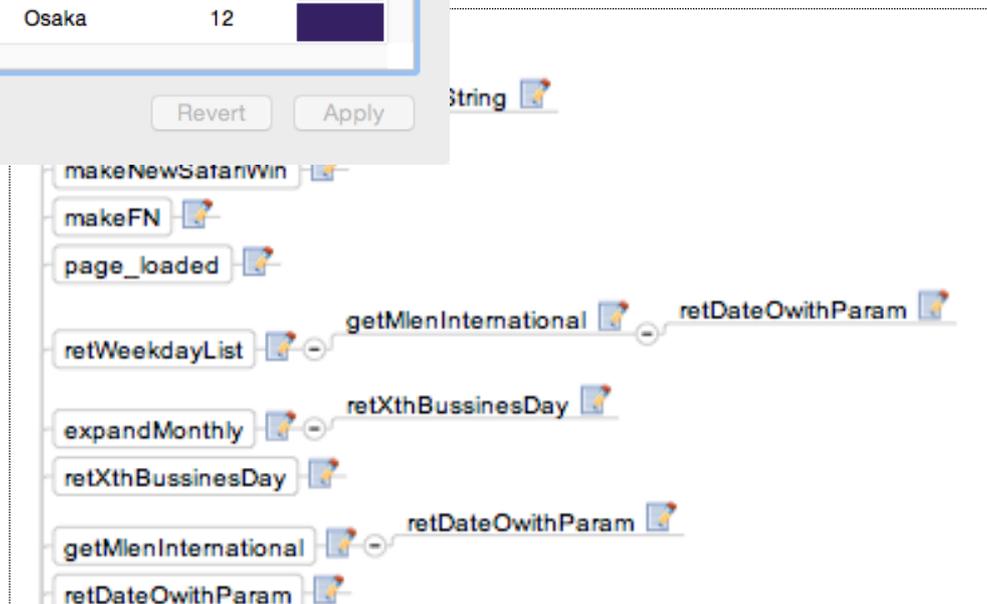
AppleScript sample-code's syntax color format:

Every syntax element's colors are different a quite little. So, scripts can detect each syntax element by itself. My "Context menu assistant" acknowledges each syntax element and replace only variable's name or make routine caller-relation map from handler names.



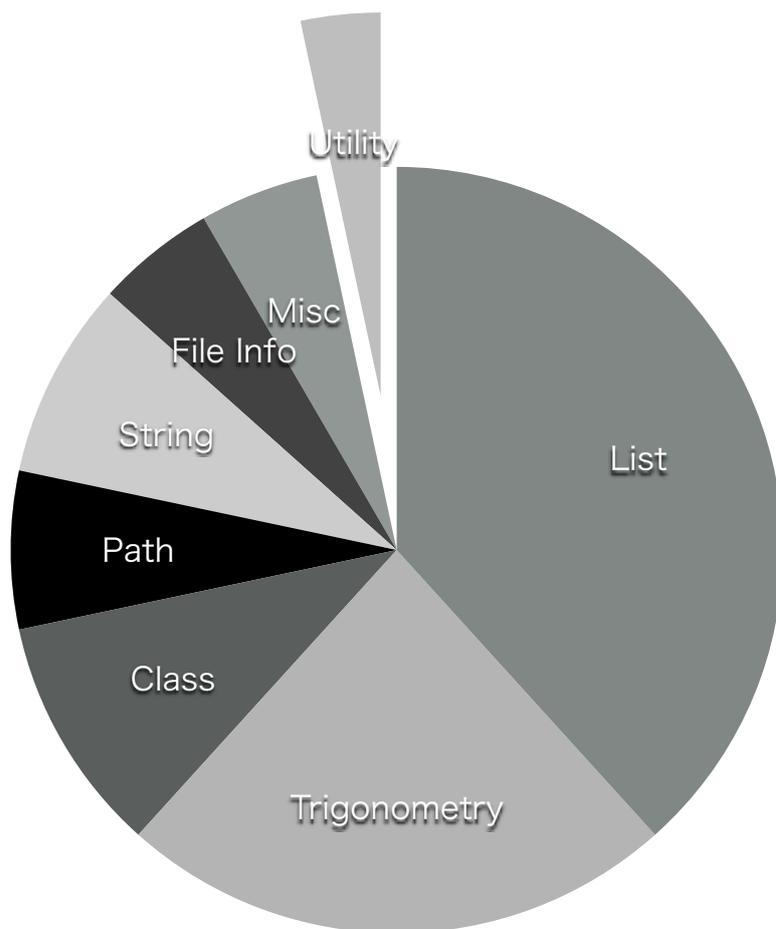
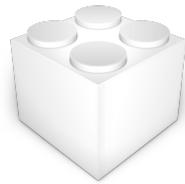
My routine caller-relation mapping generator AppleScript make it easier to understand large size of AppleScript program.

This Script detects syntax elements by its colors. My Script Editor settings are optimized for such a self-analysis solutions & my color taste.



ASObjCEXtras.framework

Utility methods



method	description()
Introduction	Returns ASObjCExtras.framework version information
kind	Utility
Input	No
Output	NSString / ("SMSFord " & version No String.)
Notes	Use this method to check ASObjCExtras.framework is installed or not. And check which version is installed. This "description" word conflicts with another application's (ex. Apple's "Script Editor") reserved word. So, this function requires " " characters to write.

— Sample Code

-- Created 2014-12-29 by Takaaki Naganoya

-- 2014 Piyomaru Software

use AppleScript version "2.4"

use scripting additions

use framework "Foundation"

use framework "ASObjCExtras"

set theVersionInfo to current application's SMSFord's |description|() as text

--> "SMSFord 1.2.2"

method	openHeaderFile()
Introduction	Open ASObjCExtras.framework's header file with Xcode
kind	Utility
Input	No
Output	No (Open Header file with Xcode)
Notes	If Xcode is not installed, nothing happen.

— Sample Code

use AppleScript version "2.4"

use scripting additions

use framework "Foundation"

use framework "ASObjCExtras"

current application's SMSFord's openHeaderFile()

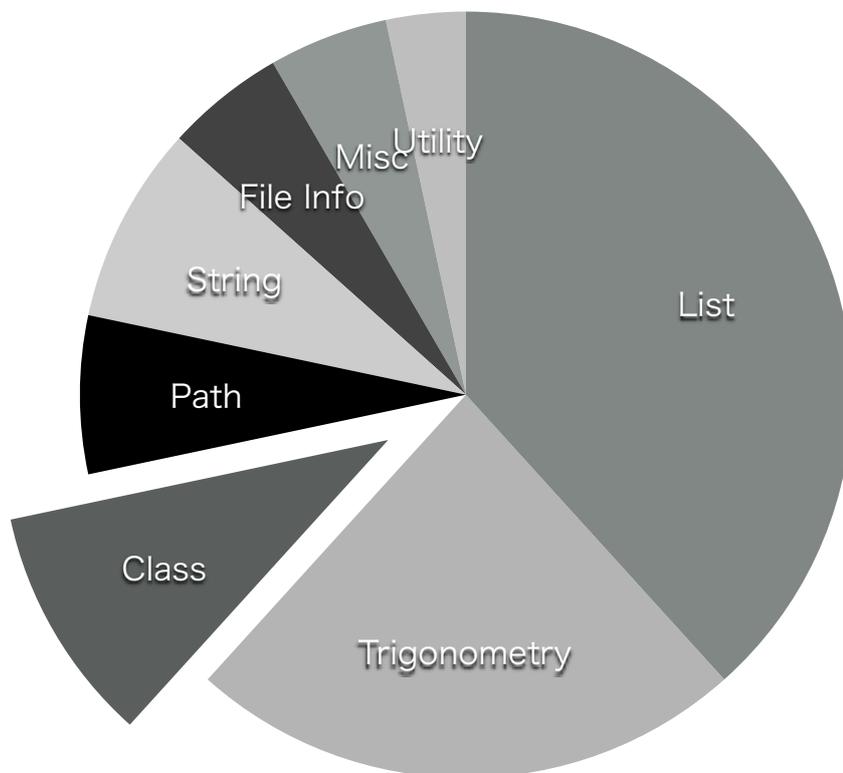
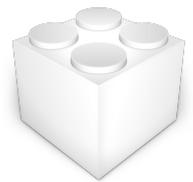
```

1 //
2 // SMSFord.h
3 // ASObjCExtras
4 //
5 // v1.2.0 Copyright (c) 2013-14 Shane Stanley.
6 //
7
8 // v1.2.0 introduces several new methods, as marked below.
9
10 // v1.1.0 introduces its own conversion of Cocoa floating-point NSNumbers to reals, to work around the imprecision of the
11 // built-in conversion. It also introduces ASify and related instance methods plus Cocoaify: and related methods to handle
12 // conversions. The fordIn: and fordOut: methods remain but are deprecated. The stringFrom:modification: method is also
13 // deprecated in favor of stringFrom:makingIt:, to work around an AppleScript bug with enums.
14
15 /*
16 AppleScriptObjC relies on the automatic conversion of text, numbers, booleans, lists and records to and from their Cocoa
17 counterparts. However, there is no conversion of dates, files, or data objects. It also loses precision when converting
18 floating-point Cocoa numbers to reals. This framework provides these extra conversions and precision with reals, as well
19 as some trigonometry functions, several text- and list-handling routines based on those in ASObjC Runner, and a handful
20 of other methods.
21
22 You can use the framework by putting it in ~/Library/Frameworks/ or /Library/Frameworks/, or by adding it to a script bundle
23 or applet.
24
25 Most methods will return missing value where there is an error, and if they have an error parameter the returned error will
26 have a description in localizedError:.
27
28 You can use and distribute this framework free of charge, but it must include this header file. There are no guarantees or
29 warranties whatsoever. Use entirely at your own risk. Feedback welcome to <ssstanley@myriad-com.com.au>.
30
31 */
32
33 #import <Foundation/Foundation.h>
34 #import "SMSFordHidden.h"
35
36 /*
37
38 ////////////////////////////////////////////////////
39 // Enums for backwards compatability
40 ////////////////////////////////////////////////////
41
42 This stuff is no longer used -- an AppleScript bug makes them largely unusable.
43 They are here for backwards compatibility only.
44
45 */
46
47 // These are the modification types used in the stringFrom:modification: method. Use stringFrom:makeIt: instead, and
48 // pass the correct string.
49
50 typedef NS_OPTIONS(NSInteger, SMSModType) {
51     SMSModMD5 = 1, // Calculates the MD5 hash of a string
52     SMSModSmartQuoted = 2, // Converts straight quote marks into typographer's quote marks
53     SMSModUnsmartQuoted = 3, // Converts typographer's quote marks into straight quote marks
54     SMSModEncodedXML = 4, // Encodes the five reserved XML characters only
55     SMSModUnecodedForXML = 5, // Decode the five reserved XML characters only
56     SMSModEncodedHex = 6, // Encodes characters outside ASCII 32-126 in hexadecimal form (&#xHHHH;);
57     SMSModEncodedDecimal = 7, // Encodes characters outside ASCII 32-126 in decimal form (&#DD;), for use in HTML
58     SMSModDecodedDecimal = 8, // Decodes characters that appear in decimal form (&#DD;) or hexadecimal form (&#xHHHH;),
59 // as used in XML and HTML
60     SMSModEmptyLineFree = 9, // Deletes any paragraphs that are empty or contain only spaces and/or tabs
61     SMSModCleanSpaced = 10, // Converts runs of more than one space to a single space character, and trims spaces from
62 // the beginning and end of paragraphs.
63
64 };

```

ASObjCEXtras.framework

Class Conversion methods



method	①'s ASify() as ②
Introduction	Converts any NSDate's to AS dates, and any floating-point numbers to reals, to work around the imprecision of the built-in conversion. Result must still be coerced.
kind	Class Conversion
Input	①Cocoa Object ②AppleScript class
Output	Cocoa Object
Notes	Typical use: set theList to anNSArray's ASify() as list

—Sample Code

-- Created 2014-12-29 by Takaaki Naganoya

-- 2014 Piyomaru Software

use AppleScript version "2.4"

use framework "Foundation"

use framework "ASObjCExtras"

use scripting additions

set aList to {1, 2, 3, 4}

set aArray to current application's NSArray's arrayWithArray:aList

set bList to (aArray's ASify()) as list

--> {1, 2, 3, 4}

description	Before (b)	After (a)
set a to (b's ASify()) as list	NSArray	list
set a to (b's ASify()) as list	NSMutableArray	list
set a to (b's ASify()) as record	NSDictionary	record
set a to (b's ASify()) as record	NSMutableDictionary	record
set a to (b's ASify()) as date	NSDate	date
set a to (b's ASify()) as real	floating-point numbers	real



method	①'s ASifyInList() as ②
Introduction	As ASify(), but returns the result in a single-item list. Use this when unsure of the class of the result, and extract the first item from the result coerced to a list. Works around coercion problem.
kind	Class Conversion
Input	①Cocoa Object ②AppleScript class
Output	NSArray
Notes	Typical use: set theThing to item 1 of (anNSThing's ASifyInList() as list)

— Sample Code

-- Created 2014-12-29 by Takaaki Naganoya

-- 2014 Piyomaru Software

use AppleScript version "2.4"

use framework "Foundation"

use framework "ASObjCEExtras"

use scripting additions

set aText to "ABCED"

set aNSString to current application's NSString's stringWithString:aText

set bRes to item 1 of ((aNSString's ASifyInList) as list)

--> "ABCED"



method	①'s ASifyFor:②
Introduction	Convert Cocoa object to AppleScript object
kind	Class Conversion
Input	①Cocoa Object ②AppleScript string
Output	NSArray
Notes	Similar to ASIfy, but you can control which classes get converted. The string you pass can be comma- or space-delimited.

--Sample Code

-- Created 2014-12-29 by Takaaki Naganoya

-- 2014 Piyomaru Software

use AppleScript version "2.4"

use scripting additions

use framework "Foundation"

use framework "ASObjCEExtras"

set aList to {1.1, 2.1, 3.1, 4.1, current date}

set aArray to current application's SMSFord's Cocoaify:aList

set bList to (aArray's ASifyFor:"reals, dates") as list

--> {1.1, 2.1, 3.1, 4.1, date "Friday, January 9, 2015 at 8:28:21 PM"}

description	After (AppleScript Object)
"reals", "floating-point numbers"	real
"dates", "NSDates"	date
"data", "NSData"	data
"files", "NSURLs"	files («class furl»)



method	①'s ASifyListFor()
Introduction	Convert Cocoa object to AppleScript object
kind	Class Conversion
Input	①Cocoa Object (NSArray)
Output	NSArray
Notes	As ASifyFor:, but returns the result in a single-item list. See ASifyInList. -In this case, you don't have to know what class to coerce the result to -- you coerce it to a list every time. So you can use it in a general handler for several classes.

--Sample Code

-- Created 2015-01-06 by Takaaki Naganoya

-- 2015 Piyomaru Software

use AppleScript version "2.4"

use scripting additions

use framework "Foundation"

use framework "ASObjCExtras"

set aList to {1.1, 2.1, 3.1, 4.1, current date}

set aArray to current application's SMSFord's Cocoaify:aList

set aRes to item 1 of ((aArray's ASifyInListFor:"dates reals") as list)

--> {1.1, 2.1, 3.1, 4.1, date "Friday, January 9, 2015 at 8:29:33 PM"}

description	After (AppleScript Object)
"reals", "floating-point numbers"	real
"dates", "NSDates"	date
"data", "NSData"	data
"files", "NSURLs"	files («class furl»)

method	Cocoaify:①
Introduction	Convert AppleScript object to Cocoa Object
kind	Class Conversion
Input	①AppleScript Object
Output	Cocoa Object
Notes	For converting from AppleScript objects to Cocoa objects. Conversion is recursive through any contained records and lists.

— Sample Code

-- Created 2014-11-29 by Takaaki Naganoya

-- 2014 Piyomaru Software

use AppleScript version "2.4"

use scripting additions

use framework "Foundation"

use framework "ASObjCEExtras"

set aList to {1, 2, 3, 4.1}

set aArray to current application's SMSFord's Cocoaify:aList

set bList to aArray's ASify() as list

--> {1, 2, 3, 4.1}

Before	After
list	NSArray, NSMutableArray
record	NSDictionary, NSMutableDictionary
date	NSDate
real	floating-point numbers



method	Cocoaify:① forTypes:②
Introduction	Convert AppleScript object to Cocoa Object
kind	Class Conversion
Input	①AppleScript Object ②Cocoa Object Class
Output	NSArray
Notes	This method is for converting from AppleScript objects to Cocoa objects. Conversion is recursive through any contained records and lists.

--Sample Code

-- Created 2014-11-29 by Takaaki Naganoya

-- 2014 Piyomaru Software

use AppleScript version "2.4"

use scripting additions

use framework "Foundation"

use framework "ASObjCEExtras"

```
set aDate to {current date, (current date) + 1, (current date) + 2}
```

```
set bNSDate to (current application's SMSFord's Cocoaify:aDate forTypes:"date")
```

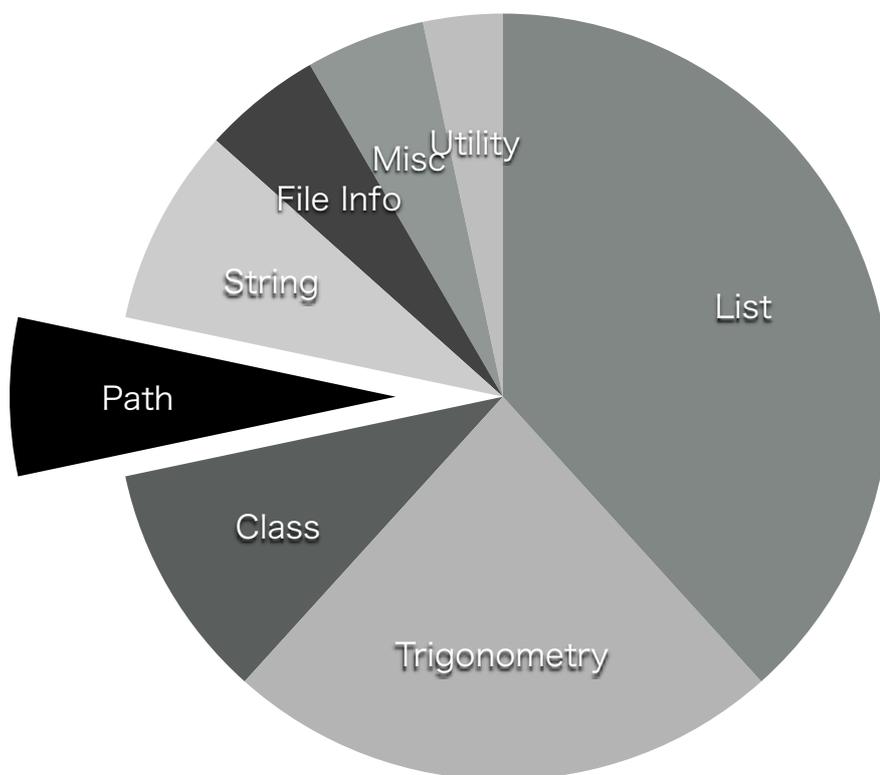
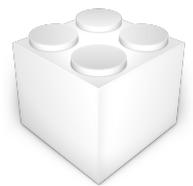
```
set bDateList to bNSDate's ASify() as list
```

```
--> {date "Friday, January 9, 2015 at 8:30:15 PM", date "Friday, January 9, 2015  
at 8:30:16 PM", date "Friday, January 9, 2015 at 8:30:17 PM"}
```

Before (AppleScript Object)	After (Cocoa Object)
dates	NSDates
data	NSData
files, files and aliases	NSURLs

ASObjCEXtras.framework

Path Manipulation methods



method	URLFrom:①
Introduction	Pass an alias, file, HFS path or POSIX path, and an NSURL will be returned
kind	Path Manipulation
Input	①AppleScript alias, file, HFS path string, POSIX path
Output	NSURL
Notes	

— Sample Code

-- Created 2014-11-29 by Takaaki Naganoya

-- 2014 Piyomaru Software

use AppleScript version "2.4"

use scripting additions

use framework "Foundation"

use framework "ASObjCExtras"

set a to choose file

set aURL to current application's SMSFord's URLFrom:a

--> file:///Applications/Automator.app/

--This is NSURL data

method	fileFromURL:①
Introduction	Pass an NSURL, and a file («class furl») will be returned. You will need to coerce the result with "as «class furl»"
kind	Path Manipulation
Input	①NSURL
Output	AppleScript's file
Notes	

— Sample Code

-- Created 2014-11-29 by Takaaki Naganoya

-- 2014 Piyomaru Software

use AppleScript version "2.4"

use scripting additions

use framework "Foundation"

use framework "ASObjCExtras"

set a to choose file

set aURL to current application's SMSFord's URLFrom:a

set aHFSPath to current application's SMSFord's fileFromURL:aURL

--> file "Macintosh HD:Applications:Automator.app:"

method	HFSPathFromURL:①
Introduction	Pass an NSURL, and the HFS path is returned. Directories will have a trailing colon; packages will not
kind	Path Manipulation
Input	①NSURL
Output	Cocoa string (HFS path string)
Notes	

—Sample Code

-- Created 2014-11-29 by Takaaki Naganoya

-- 2014 Piyomaru Software

use AppleScript version "2.4"

use scripting additions

use framework "Foundation"

use framework "ASObjCEExtras"

set a to choose file

set aURL to current application's SMSFord's URLFrom:a

set aHFSPath to current application's SMSFord's HFSPathFromURL:aURL

set aFile to aHFSPath's ASify() as text

--> "Macintosh HD:Applications:Automator.app"

method	HFSPathFromURL:① colonForPackages:②
Introduction	Pass an NSURL, and the HFS path is returned. Directories will have a trailing colon; packages will not. You can specify whether package paths have colons appended
kind	Path Manipulation
Input	①NSURL ②AppleScript boolean (true/false)
Output	NSString (HFS path string)
Notes	



—Sample Code

-- Created 2014-11-26 by Takaaki Naganoya

-- 2014 Piyomaru Software

use AppleScript version "2.4"

use scripting additions

use framework "Foundation"

use framework "ASObjCEExtras"

set a to choose file

set aURL to current application's SMSFord's URLFrom:a

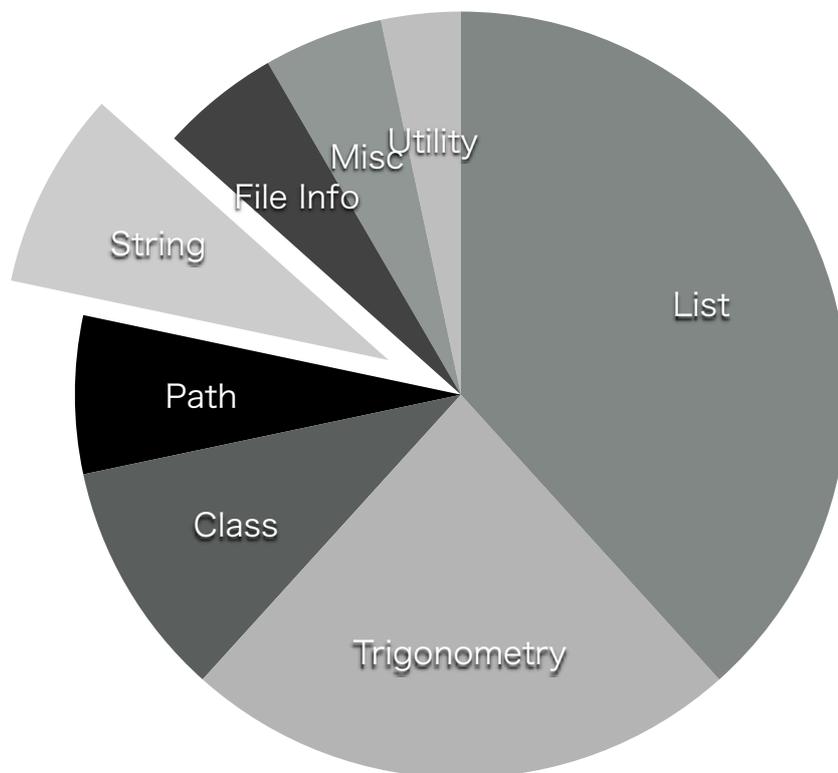
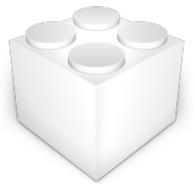
set aHFSPath to current application's SMSFord's HFSPathFromURL:aURL
colonForPackages:true

set aFile to aHFSPath's ASify() as text

--> "Macintosh HD:Applications:Automator.app:"

ASObjCEXtras.framework

String Manipulation methods



method	datesFromStrings:① format:②
Introduction	Supply the format string you wish to use to convert the strings to NSDate. For example: "yyyy-MM-dd". Pass to format: if you want AS dates. See http://unicode.org/reports/tr35/tr35-10.html#Date_Format_Patterns for format rules.
kind	String Manipulation
Input	①AppleScript string ②AppleScript string
Output	NSArray (including AppleScript date)
Notes	

--Sample Code

-- Created 2015-01-03 by Takaaki Naganoya

-- 2015 Piyomaru Software

use AppleScript version "2.4"

use scripting additions

use framework "Foundation"

use framework "ASObjCExtras"

set d1List to {"2015.01.03 00:00:00"}

set aNSDate to (current application's SMSFord's datesFromStrings:d1List |
format:"yyyy.MM.dd HH:mm:ss")

set aDate to aNSDate's ASify() as date

--> date "Saturday, January 3, 2015 at 12:00:00 AM"

set d2List to {"2015.01.03 00:00:00", "2015.01.02 00:00:00", "2015.01.01
00:00:00"}

set aNSDate to (current application's SMSFord's datesFromStrings:d2List |
format:"yyyy.MM.dd HH:mm:ss")

set aDate to (aNSDate's ASifyFor:"dates") as list

--> {date "Saturday, January 3, 2015 at 12:00:00 AM", date "Friday, January 2,
2015 at 12:00:00 AM", date "Thursday, January 1, 2015 at 12:00:00 AM"}

method	stringFrom:① ICUTransform:② inverse:③
Introduction	Apply an ICU transform. See http://userguide.icu-project.org/transforms
kind	String Manipulation
Input	①AppleScript string ②AppleScript string ③AppleScript boolean
Output	NSString
Notes	ICUTransform is just a conversion, not a translation.

--Sample Code

-- Created 2015-01-06 by Takaaki Naganoya

-- 2015 Piyomaru Software

use AppleScript version "2.4"

use scripting additions

use framework "Foundation"

use framework "ASObjCExtras"

set theString to "ながのや, たかあき" --Hiragana

set aRes to (current application's SMSFord's stringFrom:theString
ICUTransform:"Hiragana-Latin" inverse:false) as text

--> "naganoya, takaaki"

set theString to "ながのや, たかあき" --Hiragana

set aRes to (current application's SMSFord's stringFrom:theString
ICUTransform:"Hiragana-Katakana" inverse:false) as text

--> "ナガノヤ, タカアキ"--Katakana

set theString to "Takaaki, Naganoya"

set aRes to (current application's SMSFord's stringFrom:theString
ICUTransform:"Hiragana-Latin" inverse:true) as text

--> "たかあき, ながのや"--Hiragana

set theString to "Takaaki, Naganoya"

set aRes to (current application's SMSFord's stringFrom:theString
ICUTransform:"Katakana-Latin" inverse:true) as text

--> "タカアキ, ナガノヤ"--Katakana

set theString to "Shane, Stanley"

set aRes to (current application's SMSFord's stringFrom:theString
ICUTransform:"Katakana-Latin" inverse:true) as text

--> "シャネ, スタンレイ"--Katakana.....this seems odd. "シェーン, スタンリー" will be a right spelling

method	stringFrom:① makinglt:②
Introduction	Useful string conversion routines
kind	String Manipulation
Input	①AppleScript string ②AppleScript string
Output	NSString
Notes	

—Sample Code

-- Created 2015-01-04 by Takaaki Naganoya

-- 2015 Piyomaru Software

use AppleScript version "2.4"

use scripting additions

use framework "Foundation"

use framework "ASObjCExtras"

set res0 to current application's SMSFord's description() as text

--> "SMSFord 1.2.2"

--Calculates the MD5 hash of a string

set aStr to "0123456789"

set a to (current application's SMSFord's stringFrom:aStr makinglt:"MD5") as string

--> "781e5e245d69b566979b86e28d23f2c7"

--Converts straight quote marks into typographer's quote marks

set bStr to "'a'"

set b to (current application's SMSFord's stringFrom:bStr makinglt:"SmartQuoted") as string

--> "'a'"

--Converts typographer's quote marks into straight quote marks

set cStr to "'a'"

set c to (current application's SMSFord's stringFrom:cStr makinglt:"UnsmartQuoted") as string

--> "'a'"

```

--Encodes the five reserved XML characters only
set d1Str to "&\"<>"
set d1 to (current application's SMSFord's stringFrom:d1Str makingIt:"EncodedXML")
    as string
--> "&quot;&lt;&gt;&apos;"

--Decode the five reserved XML characters only
set d2Str to "&quot;&lt;&gt;&apos;"
set d2 to (current application's SMSFord's stringFrom:d2Str
    makingIt:"UnecodedForXML") as string
--> "&\"<>"

--Encodes characters outside ASCII 32-126 in hexadecimal form (&#xHHHH;)
set eStr to "あいうえお"
set e to (current application's SMSFord's stringFrom:eStr makingIt:"EncodedHex") as
    string
--> "&#x3042;&#x3044;&#x3046;&#x3048;&#x304A;"

--Encodes characters outside ASCII 32-126 in decimal form (&#DD;), for use in HTML
set fStr to "あいうえお"
set f to (current application's SMSFord's stringFrom:fStr makingIt:"EncodedDecimal")
    as string
--> "&#12354;&#12356;&#12358;&#12360;&#12362;"

--Decodes characters that appear in decimal form (&#DD;) or hexadecimal form
    (&#xHHHH;), as used in XML and HTML
set gStr to "&#12354;&#12356;&#12358;&#12360;&#12362;"
set g to (current application's SMSFord's stringFrom:gStr makingIt:"DecodedDecimal")
    as string
--> "あいうえお"

```

--Deletes any paragraphs that are empty or contain only spaces and/or tabs

```
set hStr to "a
```

```
aaa
```

```
a
```

```
a
```

```
a
```

```
"
```

```
set h to (current application's SMSFord's stringFrom:hStr makingIt:"EmptyLineFree")  
as string
```

```
-->
```

```
(*
```

```
"a
```

```
aaa
```

```
a
```

```
a
```

```
a"
```

```
*)
```

--Converts runs of more than one space to a single space character, and trims spaces from the beginning and end of paragraphs.

```
set hStr to "  aaaaa bbbb cccc "
```

```
set h to (current application's SMSFord's stringFrom:hStr makingIt:"CleanSpaced") as  
string
```

```
--> "aaaaa bbbb cccc"
```

makinglt: option strings

option string	descriptions
"MD5"	Calculates the MD5 hash of a string
"SmartQuoted"	Converts straight quote marks into typographer's quote marks
"UnsmartQuoted"	Converts typographer's quote marks into straight quote marks
"EncodedXML"	Encodes the five reserved XML characters only
"UnecodedForXML"	Decode the five reserved XML characters only
"EncodedHex"	Encodes characters outside ASCII 32-126 in hexadecimal form (&#xHHHH;)
"EncodedDecimal"	Encodes characters outside ASCII 32-126 in decimal form (&#DD;), for use in HTML
"DecodedDecimal"	Decodes characters that appear in decimal form (&#DD;) or hexadecimal form (&#xHHHH;), as used in XML and HTML
"EmptyLineFree"	Deletes any paragraphs that are empty or contain only spaces and/or tabs
"CleanSpaced"	Converts runs of more than one space to a single space character, and trims spaces from the beginning and end of paragraphs.

method	arrayFromTSV:①
Introduction	Converts tab-separated values to list of lists
kind	String Manipulation
Input	①AppleScript string
Output	NSArray (2D)
Notes	

—Sample Code

-- Created 2014-11-26 by Takaaki Naganoya

-- 2014 Piyomaru Software

use AppleScript version "2.4"

use scripting additions

use framework "Foundation"

use framework "ASObjCExtras"

set a to "1 2 3

2 2 3"

set b to (current application's SMSFord's arrayFromTSV:a) as list

--> {{"1", "2", "3"}, {"2", "2", "3"}}

method	arrayFromCSV:① commals:②
Introduction	Converts comma-separated values to a list of lists. The commals parameter defines (single) character used
kind	String Manipulation
Input	①AppleScript string ②AppleScript string
Output	NSArray (2D)
Notes	

— Sample Code

-- Created 2014-11-26 by Takaaki Naganoya

-- 2014 Piyomaru Software

use AppleScript version "2.4"

use scripting additions

use framework "Foundation"

use framework "ASObjCExtras"

set a to "1, 2, 3

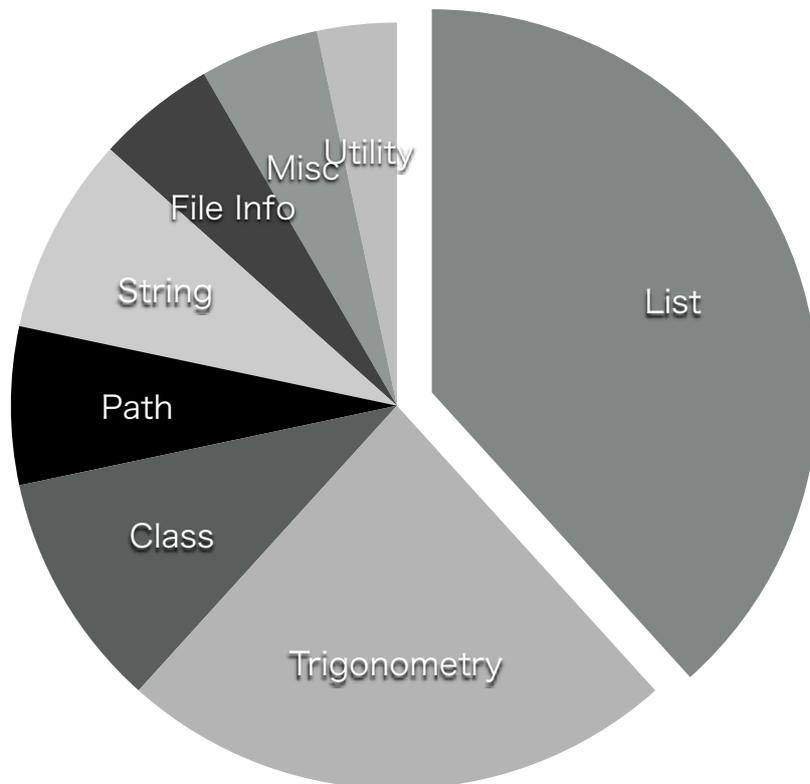
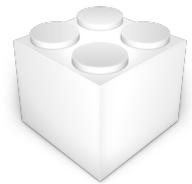
2, 2, 3"

set b to (current application's SMSFord's arrayFromCSV:a commals:",") as list

--> {"1", "2", "3"}, {"2", "2", "3"}}

ASObjCExtras.framework

List Manipulation methods



method	subarraysIn:① paddedWith:② error:③
Introduction	Assumes the array is a list of arrays, and the pad item will be used to pad out any subarrays that are shorter than the longest. This method return an array. You should use ASify() or similar to convert the results to lists if they include real numbers.
kind	List Manipulation
Input	①AppleScript list ②AppleScript object (number, string, boolean etc.) ③AppleScript missing value or reference
Output	NSArray
Notes	This method return an array. You should use ASify() or similar to convert the results to lists if they include real numbers.

—Sample Code

-- Created 2014-11-26 by Takaaki Naganoya

-- 2014 Piyomaru Software

use AppleScript version "2.4"

use framework "Foundation"

use framework "ASObjCExtras"

use scripting additions

set aList to {{1, 2, 3, 4, 5}, {1, 2, 3}, {1, 2}}

set aPad to 0

set b to (current application's SMSFord's subarraysIn:aList paddedWith:aPad lerrorl:
(missing value)) as list

--> {{1, 2, 3, 4, 5}, {1, 2, 3, 0, 0}, {1, 2, 0, 0, 0}}

method	colsToRowsIn:① error:②
Introduction	Works on a list of lists, each containing the same number of entries. The result will also be a list of lists, where the first list will consist of the first item of each of the original lists, and so on. Calling it a second time effectively returns the list to its original value
kind	List Manipulation
Input	①AppleScript list(2D) ②AppleScript missing value or reference
Output	NSArray (2D)
Notes	This method return an array. You should use ASify() or similar to convert the results to lists if they include real numbers.

—Sample Code

-- Created 2014-11-26 by Takaaki Naganoya

-- 2014 Piyomaru Software

use AppleScript version "2.4"

use framework "Foundation"

use framework "ASObjCExtras"

use scripting additions

—3x3 list

```
set aList to {{1, 2, 3}, {11, 12, 13}, {21, 22, 23}}
```

```
set b to (current application's SMSFord's colsToRowsIn:aList lerror:(missing value)) as list
```

```
--> {{1, 11, 21}, {2, 12, 22}, {3, 13, 23}}--3x3
```

—3x5 list

```
set aList to {{1, 2, 3}, {11, 12, 13}, {21, 22, 23}, {31, 32, 33}, {41, 42, 43}}
```

```
set b to (current application's SMSFord's colsToRowsIn:aList lerror:(missing value)) as list
```

```
--> {{1, 11, 21, 31, 41}, {2, 12, 22, 32, 42}, {3, 13, 23, 33, 43}}--5x3
```

method	arrayByFlattening:①
Introduction	Flattens a list of lists to a single list: {{1, 2}, {3, 4}} --> {1, 2, 3, 4}. Flattens one level only.
kind	List Manipulation
Input	①AppleScript list (2D)
Output	NSArray (1D)
Notes	This method return an array. You should use ASify() or similar to convert the results to lists if they include real numbers.

--Sample Code

-- Created 2014-11-26 by Takaaki Naganoya

-- 2014 Piyomaru Software

use AppleScript version "2.4"

use framework "Foundation"

use framework "ASObjCExtras"

use scripting additions

set aList to {{1, 2, 3}, {11, 12, 13}, {21, 22, 23}}

set b to (current application's SMSFord's arrayByFlattening:aList) as list
--> {1, 2, 3, 11, 12, 13, 21, 22, 23}

method	arrayByFullyFlattening:①
Introduction	Flattens a list to a single list: {1, 2, {3, {4, 5}}} --> {1, 2, 3, 4, 5}. Flattens all levels
kind	List Manipulation
Input	①AppleScript list (nD)
Output	NSArray (1D)
Notes	This method return an array. You should use ASify() or similar to convert the results to lists if they include real numbers.

— Sample Code

-- Created 2014-11-26 by Takaaki Naganoya

-- 2014 Piyomaru Software

use AppleScript version "2.4"

use framework "Foundation"

use framework "ASObjCExtras"

use scripting additions

```
set aList to {{1, 2, 3}, {11, 12, 13}, {21, 22, 23, {24, 25, 26}}}
```

```
set b to (current application's SMSFord's arrayByFullyFlattening:aList) as list
```

```
--> {1, 2, 3, 11, 12, 13, 21, 22, 23, 24, 25, 26}
```

method	subarraysFrom:① groupedBy:② error:③
Introduction	Breaks list into subarrays of aNumber items
kind	List Manipulation
Input	①AppleScript list (1D) ②AppleScript number ③AppleScript missing value or reference
Output	NSArray (2D)
Notes	This method return an array. You should use ASify() or similar to convert the results to lists if they include real numbers.

--Sample Code

-- Created 2014-11-26 by Takaaki Naganoya

-- 2014 Piyomaru Software

use AppleScript version "2.4"

use framework "Foundation"

use framework "ASObjCExtras"

use scripting additions

```
set aList to {1, 2, 3, 11, 12, 13, 21, 22, 23}
```

```
set aGroupNum to 3
```

```
set b to (current application's SMSFord's subarraysFrom:aList
    groupedBy:aGroupNum lerror:(missing value)) as list
```

```
--> {{1, 2, 3}, {11, 12, 13}, {21, 22, 23}}
```

```
set aGroupNum to 5
```

```
set c to (current application's SMSFord's subarraysFrom:aList
    groupedBy:aGroupNum lerror:(missing value)) as list
```

```
--> {{1, 2, 3, 11, 12}, {13, 21, 22, 23}}
```

method	subarraysIn:① withItems:② insertedAtIndex:③ error:④
Introduction	Assumes the array is a list of lists. The insertion list must have the same number of elements, and the first element will be inserted into the main list's first list, the second into the main list's second list, and so on. Where it is inserted is determined by the (zero-based) 'insertedAtIndex' parameter.
kind	List Manipulation
Input	①AppleScript list (2D) ②AppleScript list (1D) ③AppleScript number ④AppleScript missing value or reference
Output	NSArray (2D)
Notes	This method return an array. You should use ASify() or similar to convert the results to lists if they include real numbers.

--Sample Code

-- Created 2014-11-26 by Takaaki Naganoya

-- 2014 Piyomaru Software

use AppleScript version "2.4"

use framework "Foundation"

use framework "ASObjCExtras"

use scripting additions

--Case: Number of every item is same

```
set aList to {{1, 2, 3}, {11, 12, 13}, {21, 22, 23}}
```

```
set insArray to {100, 101, 102}
```

```
set anIndex to 0
```

```
set b to (current application's SMSFord's subarraysIn:aList withItems:insArray  
insertedAtIndex:anIndex lerror:(missing value)) as list
```

```
--> {{100, 1, 2, 3}, {101, 11, 12, 13}, {102, 21, 22, 23}}
```

--Case: Number of every item is not same

```
set aList to {{1, 2, 3}, {11, 12, 13}, {21, 22, 23}}
```

```
set insArray to {100, 101}
```

```
set anIndex to 0
```

```
set b to (current application's SMSFord's subarraysIn:aList withItems:insArray  
insertedAtIndex:anIndex lerror:(missing value)) as list
```

```
--> {missing value}
```

method	arrayByInsertingItems:① inArray:② atIndex:③ error:④
Introduction	Inserts a list of items as separate items into the list. Index is zero-based.
kind	List Manipulation
Input	①AppleScript list (2D) ②AppleScript list (2D) ③AppleScript number ④missing value
Output	NSArray
Notes	This method return an array. You should use ASify() or similar to convert the results to lists if they include real numbers.

--Sample Code

-- Created 2014-12-30 by Takaaki Naganoya

-- 2014 Piyomaru Software

use AppleScript version "2.4"

use framework "Foundation"

use framework "ASObjCExtras"

use scripting additions

set aList to {{1, 2, 3}, {11, 12, 13}, {21, 22, 23}}

set insArray to {{100, 101, 102}}

set anIndex to 1

set b to (current application's SMSFord's arrayByInsertingItems:insArray
inArray:aList atIndex:anIndex lerror:(missing value)) as list

--> {{1, 2, 3}, {100, 101, 102}, {11, 12, 13}, {21, 22, 23}}

method	arrayByMovingItemAt:① toIndex:② inArray:③ error:④
Introduction	Does what it says. Indexes are zero-based.
kind	List Manipulation
Input	①AppleScript number ②AppleScript number ③AppleScript list (1D/2D) ④AppleScript missing value or reference
Output	NSArray
Notes	This method return an array. You should use ASify() or similar to convert the results to lists if they include real numbers.

--Sample Code

-- Created 2014-12-30 by Takaaki Naganoya

-- 2014 Piyomaru Software

use AppleScript version "2.4"

use framework "Foundation"

use framework "ASObjCExtras"

use scripting additions

```
set aList to {1, 2, 3, 11, 12, 13, 21, 22, 23}
```

```
set b to (current application's SMSFord's arrayByMovingItemAt:1 toIndex:3  
inArray:aList lerror:(missing value)) as list
```

```
--> {1, 3, 11, 2, 12, 13, 21, 22, 23}
```

```
set aList to {{1, 2, 3}, {11, 12, 13}, {21, 22, 23}}
```

```
set b to (current application's SMSFord's arrayByMovingItemAt:1 toIndex:2  
inArray:aList lerror:(missing value)) as list
```

```
--> {{1, 2, 3}, {21, 22, 23}, {11, 12, 13}}
```

method	arrayByDeletingBlanksIn:①
Introduction	Delete any blank items. Blanks are missing value, empty strings, empty lists, consisting only of the above.
kind	List Manipulation
Input	①AppleScript list
Output	NSArray
Notes	This method return an array. You should use ASify() or similar to convert the results to lists if they include real numbers.

—Sample Code

-- Created 2014-12-30 by Takaaki Naganoya

-- 2014 Piyomaru Software

use AppleScript version "2.4"

use framework "Foundation"

use framework "ASObjCExtras"

use scripting additions

set aList to {1, 2, 3, 11, 12, 13, 21, 22, 23, "", {}}

set b to (current application's SMSFord's arrayByDeletingBlanksIn:aList) as list

-->{1, 2, 3, 11, 12, 13, 21, 22, 23}

method	arrayByTrimmingTrailingBlanksFrom:①
Introduction	Trim any blank items from the trailing end of the list. Blanks are missing value, empty strings, empty lists, and lists consisting only of the above.
kind	List Manipulation
Input	①AppleScript list
Output	NSArray
Notes	This method return an array. You should use ASify() or similar to convert the results to lists if they include real numbers.

-- Created 2014-12-30 by Takaaki Naganoya

-- 2014 Piyomaru Software

use AppleScript version "2.4"

use scripting additions

use framework "Foundation"

use framework "ASObjCExtras"

set aList to {1, 2, 3, 11, 12, 13, 21, 22, 23, "", {}}

set b to (current application's SMSFord's
arrayByTrimmingTrailingBlanksFrom:aList) as list

-->{1, 2, 3, 11, 12, 13, 21, 22, 23}

method	arrayByTrimmingBlanksFrom:①
Introduction	Trim any blank items from both ends of the list. Blanks are missing value, empty strings, empty lists, and lists consisting only of the above.
kind	List Manipulation
Input	①AppleScript list
Output	NSArray
Notes	This method return an array. You should use ASify() or similar to convert the results to lists if they include real numbers.

— Sample Code

-- Created 2014-12-30 by Takaaki Naganoya

-- 2014 Piyomaru Software

use AppleScript version "2.4"

use scripting additions

use framework "Foundation"

use framework "ASObjCExtras"

set aList to {missing value, 1, 2, 3, 11, 12, 13, 21, 22, 23, "", {}}

set b to (current application's SMSFord's arrayByTrimmingBlanksFrom:aList) as list

--> {1, 2, 3, 11, 12, 13, 21, 22, 23}

method	arrayByReplacingNullsIn:① withItem:②
Introduction	Instances of 'missing value' will be replaced with whatever you pass to withItem:
kind	List Manipulation
Input	①AppleScript list ②AppleScript Object
Output	NSArray
Notes	This method return an array. You should use ASify() or similar to convert the results to lists if they include real numbers.

— Sample Code

-- Created 2014-12-30 by Takaaki Naganoya

-- 2014 Piyomaru Software

use AppleScript version "2.4"

use framework "Foundation"

use framework "ASObjCExtras"

use scripting additions

set aList to {1, 2, 3, 4, missing value}

set repltem to 0

set b to (current application's SMSFord's arrayByReplacingNullsIn:aList
withItem:repltem) as list

--> {1, 2, 3, 4, 0}



method	arrayWithPattern:① startNumber:② endNumber:③ minDigits:④
Introduction	Pattern should be a string where every instance of %@ will be replaced by a number. So a pattern of "label%@", a startNumber of 3, an endNumber of 1 and minDigits 3 will return {"label003", "label002", "label001"}
kind	List Manipulation
Input	①AppleScript string ②AppleScript number ③AppleScript number ④AppleScript number
Output	NSArray
Notes	This method return an array. You should use ASify() or similar to convert the results to lists if they include real numbers.

—Sample Code

-- Created 2014-11-29 by Takaaki Naganoya

-- 2014 Piyomaru Software

use AppleScript version "2.4"

use scripting additions

use framework "Foundation"

use framework "ASObjCExtras"

```
set aArray to current application's SMSFord's arrayWithPattern:"Piyomaru%@", startNumber:1  
endNumber:100 minDigits:3
```

```
set aList to aArray's ASify() as list
```

```
--> {"Piyomaru001", "Piyomaru002", "Piyomaru003", "Piyomaru004", "Piyomaru005",  
"Piyomaru006", "Piyomaru007", "Piyomaru008", "Piyomaru009", "Piyomaru010",  
"Piyomaru011", "Piyomaru012", "Piyomaru013", "Piyomaru014", "Piyomaru015",  
"Piyomaru016", "Piyomaru017", "Piyomaru018", "Piyomaru019", "Piyomaru020",  
"Piyomaru021", "Piyomaru022", "Piyomaru023", "Piyomaru024", "Piyomaru025",  
"Piyomaru026", "Piyomaru027", "Piyomaru028", "Piyomaru029", "Piyomaru030",  
"Piyomaru031", "Piyomaru032", "Piyomaru033", "Piyomaru034", "Piyomaru035",  
"Piyomaru036", "Piyomaru037", "Piyomaru038", "Piyomaru039", "Piyomaru040",  
"Piyomaru041", "Piyomaru042", "Piyomaru043", "Piyomaru044", "Piyomaru045",  
"Piyomaru046", "Piyomaru047", "Piyomaru048", "Piyomaru049", "Piyomaru050",  
"Piyomaru051", "Piyomaru052", "Piyomaru053", "Piyomaru054", "Piyomaru055",  
"Piyomaru056", "Piyomaru057", "Piyomaru058", "Piyomaru059", "Piyomaru060",  
"Piyomaru061", "Piyomaru062", "Piyomaru063", "Piyomaru064", "Piyomaru065",  
"Piyomaru066", "Piyomaru067", "Piyomaru068", "Piyomaru069", "Piyomaru070",  
"Piyomaru071", "Piyomaru072", "Piyomaru073", "Piyomaru074", "Piyomaru075",  
"Piyomaru076", "Piyomaru077", "Piyomaru078", "Piyomaru079", "Piyomaru080",  
"Piyomaru081", "Piyomaru082", "Piyomaru083", "Piyomaru084", "Piyomaru085",  
"Piyomaru086", "Piyomaru087", "Piyomaru088", "Piyomaru089", "Piyomaru090",  
"Piyomaru091", "Piyomaru092", "Piyomaru093", "Piyomaru094", "Piyomaru095",  
"Piyomaru096", "Piyomaru097", "Piyomaru098", "Piyomaru099", "Piyomaru100"}
```

method	sumMaxMinOf:① error:②
Introduction	Returns an array of {sum, max, min} of the list.
kind	List Manipulation
Input	①AppleScript list ②AppleScript missing value or reference
Output	NSArray
Notes	This method return an array. You should use ASify() or similar to convert the results to lists if they include real numbers.

— Sample Code

-- Created 2014-12-30 by Takaaki Naganoya

-- 2014 Piyomaru Software

use AppleScript version "2.4"

use framework "Foundation"

use framework "ASObjCExtras"

use scripting additions

set aList to {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}

set {sumNum, maxNum, minimumNum} to (current application's SMSFord's
sumMaxMinOf:(aList) lerror:(missing value)) as list

--> {55.0, 10.0, 1.0}

method	subarraysIn:① sortedByIndexes:② ascending:③ sortTypes:④ error:⑤
Introduction	Sort a lists of lists based on indexes of the sublists. So if the (zero-based) indexes are {3, 2}, the list will be sorted based on the value of the fourth item of the sublists, and in cases of equality, then on the third. The ascending argument should be a list of booleans, matching the indexes; if it has fewer entries, the last entry will be used for subsequent sort order. Pass an empty list to use true throughout. The type argument should be a list of strings consisting of the comparison selectors to use: "compare:", "localizedCompare:", "caseInsensitiveCompare:", "localizedCaseInsensitiveCompare:" and "localizedStandardCompare:". Pass an empty list to use "compare:" throughout.
kind	List Sorting
Input	①AppleScript list (2D) ②AppleScript list ③AppleScript list ④AppleScript list ⑤AppleScript missing value or reference
Output	NSArray
Notes	This method return an array. You should use ASify() or similar to convert the results to lists if they include real numbers.

—Sample Code

-- Created 2014-12-30 by Takaaki Naganoya

-- 2014 Piyomaru Software

use AppleScript version "2.4"

use scripting additions

use framework "Foundation"

use framework "ASObjCExtras"

```
set aList to {{1, 2, 3}, {11, 12, 13}, {21, 22, 23}, {21, 23, 23}}
```

```
set sortIndexes to {0, 1} --Key Item id: begin from 0
```

```
set sortOrders to {false, true}
```

```
set sortTypes to {"compare:", "compare:"}
```

```
set rList to (current application's SMSFord's subarraysIn:(aList) sortedByIndexes:
    (sortIndexes) ascending:(sortOrders) sortTypes:(sortTypes) lerror:(missing
    value)) as list
```

```
--> {{21, 22, 23}, {21, 23, 23}, {11, 12, 13}, {1, 2, 3}}
```



method	subarraysIn:① sortedByIndexes:② ascending:③ sortTypes:④ sortKeys:⑤ error:⑥
Introduction	As “subarraysIn:① sortedByIndexes:② ascending:③ sortTypes:④ error:⑤”, but you can also pass a list of key strings to be used for the sort. Pass an empty list to use "self" throughout, which is what the above method uses. For example, if you pass {"length"}, the sorting will be done on the basis of the length. The values being sorted must support the key or an error will be thrown.
kind	List Sorting
Input	①AppleScript list (2D) ②AppleScript list(1D) ③AppleScript list (1D) ④AppleScript list (1D) ⑤AppleScript missing value or reference
Output	NSArray
Notes	This method return an array. You should use ASify() or similar to convert the results to lists if they include real numbers.

— Sample Code

-- Created 2014-12-30 by Takaaki Naganoya

-- 2014 Piyomaru Software

use AppleScript version "2.4"

use scripting additions

use framework "Foundation"

use framework "ASObjCExtras"

```
set aList to {{1, 2, "Piyomaru"}, {2, 1, "Hiyoko"}, {1, 1, "Glass"}, {4, 1, "Battery"}}
```

```
set sortIndexes to {2, 1, 0} --Key Item id: begin from 0
```

```
set sortOrders to {false, true, true}
```

```
set sortTypes to {"compare:", "compare:", "compare:"}
```

```
set rList to (current application's SMSFord's subarraysIn:(aList) sortedByIndexes:  
    (sortIndexes) ascending:(sortOrders) sortTypes:(sortTypes) sortKeys:{"length",  
    "self", "self"} lerror:(missing value)) as list
```

```
--> {{1, 2, "Piyomaru"}, {4, 1, "Battery"}, {2, 1, "Hiyoko"}, {1, 1, "Glass"}}
```

method	arrayByMergingTextAtIndexes:① inArray:② inserting:③ error:④
Introduction	Concatenate two text items into one. Provide a list of the (zero-based) indexes of the items to merge; they will be joined using the separator in the order provided, and appear at the first of the indexes. Empty strings will be ignored.
kind	List Manipulation
Input	①AppleScript list ②AppleScript list ③NSString or AppleScript string ④AppleScript missing value or reference
Output	NSArray
Notes	This method return an array. You should use ASify() or similar to convert the results to lists if they include real numbers.

—Sample Code

-- Created 2014-12-30 by Takaaki Naganoya

-- 2014 Piyomaru Software

use AppleScript version "2.4"

use scripting additions

use framework "Foundation"

use framework "ASObjCExtras"

set aList to {"Piyomaru", "Hiyoko", "Glass", "Battery"}

set mergeIndexes to {2, 1} --Key Item id: begin from 0

set rList to (current application's SMSFord's arrayByMergingTextAtIndexes:

(mergeIndexes) inArray:(aList) inserting:("____") lerror:(missing value)) as list

--> {"Piyomaru", "Glass____Hiyoko", "Battery"}

method	indexesOfItem:① inArray:② inverting:③
Introduction	Returns list of indexes where item is found; inverting true returns indexes where the item is not found. Indexes are zero-based.
kind	List Manipulation
Input	①NSString or AppleScript string ②AppleScript list ③AppleScript boolean
Output	NSArray
Notes	This method return an array. You should use ASify() or similar to convert the results to lists if they include real numbers.

— Sample Code

-- Created 2014-12-06 by Takaaki Naganoya

-- 2014 Piyomaru Software

use AppleScript version "2.4"

use framework "Foundation"

use framework "ASObjCExtras"

use scripting additions

```
set aList to {"1", "2", "3", "4", "5", "6", "7", "8", "9", "0", "1", "2", "3", "4",
             "5", "6", "7", "8", "9", "0", "1", "2", "3", "4", "5", "6", "7", "8", "9", "0"}
```

```
set aRes to (current application's SMSFord's indexesOfItem:"0" inArray:aList
            inverting:false) as list
```

--> {9, 19, 29}--0 based

method	indexesOfItems:① inArray:② inverting:③
Introduction	Returns list of indexes where any of the items in the list are found; inverting true returns indexes where the items are not found. Indexes are zero-based.
kind	List Manipulation
Input	①NSArray or AppleScript list ②NSArray or AppleScript list ③AppleScript boolean
Output	NSArray
Notes	This method return an array. You should use ASify() or similar to convert the results to lists if they include real numbers.

— Sample Code

-- Created 2014-12-06 by Takaaki Naganoya

-- 2014 Piyomaru Software

use AppleScript version "2.4"

use framework "Foundation"

use framework "ASObjCExtras"

use scripting additions

```
set aList to {"1", "2", "3", "4", "5", "6", "7", "8", "9", "0", "1", "2", "3", "4",
  "5", "6", "7", "8", "9", "0", "1", "2", "3", "4", "5", "6", "7", "8", "9", "0"}
```

```
set aRes to (current application's SMSFord's indexesOfItems:{"0", "8"}
  inArray:aList inverting:false) as list
```

--> {7, 9, 17, 19, 27, 29}--0 based

method	subarraysIn:① asDictionariesUsingLabels:② error:③
Introduction	Assumes the array is a list of lists, and that each list has the same number of items as the list of labels. The result will be an array of records/dictionaries that use the supplied labels in order.
kind	List Manipulation
Input	①AppleScript list ②AppleScript list ③AppleScript missing value or reference
Output	NSArray
Notes	This method return an array. You should use ASify() or similar to convert the results to lists if they include real numbers.

— Sample Code

-- Created 2014-12-06 by Takaaki Naganoya

-- 2014 Piyomaru Software

use AppleScript version "2.4"

use scripting additions

use framework "Foundation"

use framework "ASObjCExtras"

```
set aList to {"aLabel", "bLabel"}
```

```
set bList to {{1, 2}, {2, 3}, {3, 4}}
```

```
set aArray to current application's SMSFord's subarraysIn:bList
```

```
    asDictionariesUsingLabels:aList lerror:(missing value)
```

```
set bList to aArray's ASify() as list
```

```
--> {{aLabel:1, bLabel:2}, {aLabel:2, bLabel:3}, {aLabel:3, bLabel:4}}
```

method	subarraysFrom:① usingKeys:② outKeys:③ error:④
Introduction	Pass a list of records/dictionaries and a list of labels, and a list of lists will be returned, with the order of the values in each sublist matching the order of the labels. If an empty list is passed for usingKeys, the keys of the first item will be used, sorted in case-insensitive alphabetical order, and these keys will be listed as strings in outKeys.
kind	List Manipulation
Input	①AppleScript list of records ②AppleScript list ③AppleScript list or reference ④AppleScript missing value or reference
Output	NSArray (2D/1D)
Notes	This method return an array. You should use ASify() or similar to convert the results to lists if they include real numbers.



—Sample Code

-- Created 2014-12-06 by Takaaki Naganoya

-- 2014 Piyomaru Software

use AppleScript version "2.4"

use scripting additions

use framework "Foundation"

use framework "ASObjCExtras"

```
set aList to {{age:10, aName:"ccc", weight:70}, {age:20, aName:"bbb", weight:80}, {age:3, aName:"aaaa", weight:10}}
```

```
set aArray to current application's SMSFord's subarraysFrom:aList usingKeys:{"weight", "aName"} outKeys:{"aName"} lerror:(missing value)
```

```
set bList to aArray's ASify() as list
```

```
--> {{70, "ccc"}, {80, "bbb"}, {10, "aaaa"}}
```

--another sample

```
set aList to {{aName:"ccc", weight:70}, {age:20, aName:"bbb", weight:80}, {age:3, aName:"aaaa", weight:10}}
```

```
set {aArray, theKeys} to current application's SMSFord's subarraysFrom:aList usingKeys:{} outKeys:(reference) lerror:(missing value)
```

```
set bList to aArray's ASify() as list
```

```
set cList to theKeys's ASify() as list
```

```
--> {"aName", "weight"}
```

method	arrayWithIndexSet:①
Introduction	ASObjC can't normally extract the values from an IndexSet; this method makes it possible
kind	List Manipulation
Input	①NSMutableIndexSet, NSIndexSet
Output	NSArray
Notes	This method return an array. You should use ASify() or similar to convert the results to lists if they include real numbers.

--Sample Code

-- Created 2014-12-30 by Takaaki Naganoya

-- 2014 Piyomaru Software

use AppleScript version "2.4"

use scripting additions

use framework "Foundation"

use framework "ASObjCExtras"

set aindexSet to current application's NSMutableIndexSet's alloc()'s init()

repeat 100 times

 set aRandom to random number from 1 to 100

 aindexSet's addIndex:aRandom

end repeat

set aList to (current application's SMSFord's arrayWithIndexSet:aindexSet) as list

```
--> {2, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 23, 24, 25,
    28, 29, 30, 31, 32, 33, 34, 37, 38, 39, 40, 43, 52, 53, 56, 57, 59, 61, 63, 67,
    70, 74, 76, 78, 79, 80, 81, 83, 84, 85, 87, 88, 89, 90, 91, 92, 94, 96}
```

--Another Sample

set anIndexSet to current application's NSMutableIndexSet's indexSet()

anIndexSet's addIndexesInRange:(current application's NSMakeRange(5, 20))

set aList to (current application's SMSFord's arrayWithIndexSet:anIndexSet) as list

```
--> {5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24}
```

method	arrayByAddingInteger:① inArray:②
Introduction	Increment each of the values in a list of integers by anInteger. Provide a negative value to decrement the values
kind	List Manipulation
Input	①AppleScript list ②AppleScript number
Output	NSArray
Notes	This method return an array. You should use ASify() or similar to convert the results to lists if they include real numbers.



—Sample Code

-- Created 2014-11-30 by Takaaki Naganoya

-- 2014 Piyomaru Software

use AppleScript version "2.4"

use scripting additions

use framework "Foundation"

use framework "ASObjCExtras"

```
set aList to {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}
```

```
set aArray to current application's SMSFord's arrayByAddingInteger:1
  inArray:aList
```

```
set bList to aArray's ASify() as list
```

```
--> {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}
```

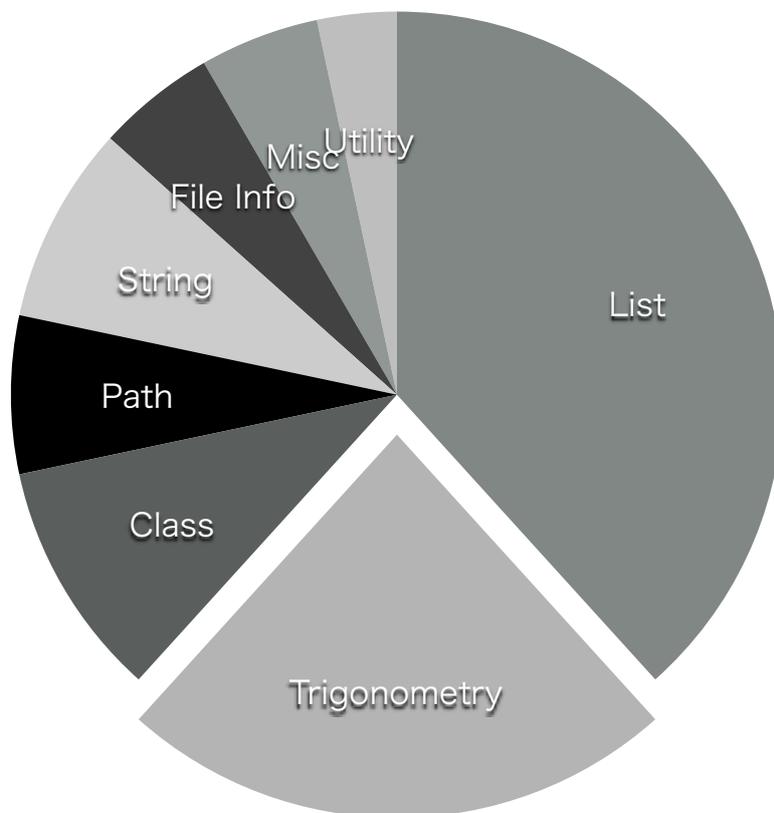
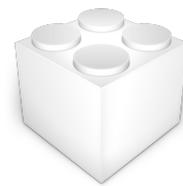
```
set bArray to current application's SMSFord's arrayByAddingInteger:-1
  inArray:aList
```

```
set cList to bArray's ASify() as list
```

```
--> {-1, 0, 1, 2, 3, 4, 5, 6, 7, 8}
```

ASObjCEXtras.framework

Trigonometry methods



method	tanValueOf:①
Introduction	Pass a single number or a list of numbers. Results must be coerced using "as real" or "as list". Angles are in radians, and errors are returned as missing value.
kind	Trigonometry
Input	①AppleScript number (real) or AppleScript list
Output	NSNumber or NSArray
Notes	

— Sample Code

-- Created 2014-11-26 by Takaaki Naganoya

-- 2014 Piyomaru Software

use AppleScript version "2.4"

use framework "Foundation"

use framework "ASObjCEExtras"

use scripting additions

set aList to {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}

--Real Value

set aVal to current application's SMSFord's tanValueOf:30

set bVal to aVal's ASify() as real

--> -6.405331196646

--List of Real Value

set aArray to (current application's SMSFord's tanValueOf:aList)

set bList to aArray's ASify() as list

--> {1.557407724655, -2.185039863262, -0.142546543074, 1.15782128235,
-3.380515006247, -0.291006191385, 0.871447982724, -6.79971145522,
-0.452315659442, 0.648360827459}

method	sinValueOf:①
Introduction	Pass a single number or a list of numbers. Results must be coerced using "as real" or "as list". Angles are in radians, and errors are returned as missing value.
kind	Trigonometry
Input	①AppleScript number (real) or AppleScript list
Output	NSNumber or NSArray
Notes	

--Sample Code

-- Created 2014-11-30 by Takaaki Naganoya

-- 2014 Piyomaru Software

use AppleScript version "2.4"

use framework "Foundation"

use framework "ASObjCEExtras"

use scripting additions

```
set aList to {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}
```

--Real Value

```
set aVal to current application's SMSFord's sinValueOf:30
```

```
set bVal to aVal's ASify() as real
```

```
--> -0.988031624093
```

--List of Real Value

```
set aArray to (current application's SMSFord's sinValueOf:aList)
```

```
set bList to aArray's ASify() as list
```

```
--> {0.841470984808, 0.909297426826, 0.14112000806,  
-0.756802495308, -0.958924274663, -0.279415498199,  
0.656986598719, 0.989358246623, 0.412118485242,  
-0.544021110889}
```

method	cosValueOf:①
Introduction	Pass a single number or a list of numbers. Results must be coerced using "as real" or "as list". Angles are in radians, and errors are returned as missing value.
kind	Trigonometry
Input	①AppleScript number (real) or AppleScript list
Output	NSNumber or NSArray
Notes	

--Sample Code

-- Created 2014-11-30 by Takaaki Naganoya

-- 2014 Piyomaru Software

use AppleScript version "2.4"

use framework "Foundation"

use framework "ASObjCExtras"

use scripting additions

--Real Value

set aVal to current application's SMSFord's cosValueOf:-0.5

set bVal to aVal's ASify() as real

--> 0.87758256189

--List of Real Value

set aList to {0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9}

set aArray to (current application's SMSFord's cosValueOf:aList)

set bList to aArray's ASify() as list

-->{0.995004165278, 0.980066577841, 0.955336489126,
0.921060994003, 0.87758256189, 0.82533561491, 0.764842187284,
0.696706709347, 0.621609968271}

method	atanValueOf:①
Introduction	Pass a single number or a list of numbers. Results must be coerced using "as real" or "as list". Angles are in radians, and errors are returned as missing value.
kind	Trigonometry
Input	①AppleScript number (real) or AppleScript list
Output	NSNumber or NSArray
Notes	

--Sample Code

-- Created 2014-11-30 by Takaaki Naganoya

-- 2014 Piyomaru Software

use AppleScript version "2.4"

use framework "Foundation"

use framework "ASObjCEExtras"

use scripting additions

--Real Value

set aVal to current application's SMSFord's atanValueOf:-0.5

set bVal to aVal's ASify() as real

--> -0.463647609001

--List of Real Value

set aList to {0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9}

set aArray to (current application's SMSFord's atanValueOf:aList)

set bList to aArray's ASify() as list

--> {0.099668652491, 0.19739555985, 0.291456794478, 0.380506377112,
0.463647609001, 0.540419500271, 0.610725964389,
0.674740942224, 0.732815101787}

method	asinValueOf:①
Introduction	Pass a single number or a list of numbers. Results must be coerced using "as real" or "as list". Angles are in radians, and errors are returned as missing value.
kind	Trigonometry
Input	①AppleScript number (real) or AppleScript list
Output	NSNumber or NSArray
Notes	

--Sample Code

-- Created 2014-11-30 by Takaaki Naganoya

-- 2014 Piyomaru Software

use AppleScript version "2.4"

use framework "Foundation"

use framework "ASObjCEExtras"

use scripting additions

--Real Value

set aVal to current application's SMSFord's asinValueOf:-0.5

set bVal to aVal's ASify() as real

--> -0.523598775598

--List of Real Value

set aList to {0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9}

set aArray to (current application's SMSFord's asinValueOf:aList)

set bList to aArray's ASify() as list

--> {0.100167421162, 0.20135792079, 0.304692654015, 0.411516846067,
0.523598775598, 0.643501108793, 0.775397496611,
0.927295218002, 1.119769514999}

method	acosValueOf:①
Introduction	Pass a single number or a list of numbers. Results must be coerced using "as real" or "as list". Angles are in radians, and errors are returned as missing value.
kind	Trigonometry
Input	①AppleScript number (real) or AppleScript list
Output	NSNumber or NSArray
Notes	

--Sample Code

-- Created 2014-11-30 by Takaaki Naganoya

-- 2014 Piyomaru Software

use AppleScript version "2.4"

use framework "Foundation"

use framework "ASObjCExtras"

use scripting additions

--Real Value

set aVal to current application's SMSFord's acosValueOf:-0.5

set bVal to aVal's ASify() as real

--> 2.094395102393

--List of Real Value

set aList to {0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9}

set aArray to (current application's SMSFord's acosValueOf:aList)

set bList to aArray's ASify() as list

--> {1.470628905633, 1.369438406005, 1.26610367278, 1.159279480727,
1.047197551197, 0.927295218002, 0.795398830184, 0.643501108793,
0.451026811796}

method	<code>tanhValueOf:</code> ①
Introduction	Pass a single number or a list of numbers. Results must be coerced using "as real" or "as list". Angles are in radians, and errors are returned as missing value.
kind	Trigonometry
Input	①AppleScript number (real) or AppleScript list
Output	NSNumber or NSArray
Notes	

--Sample Code

-- Created 2014-11-30 by Takaaki Naganoya

-- 2014 Piyomaru Software

use AppleScript version "2.4"

use framework "Foundation"

use framework "ASObjCExtras"

use scripting additions

--Real Value

set aVal to current application's SMSFord's tanhValueOf:-0.5

set bVal to aVal's ASify() as real

--> -0.46211715726

--List of Real Value

set aList to {0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9}

set aArray to (current application's SMSFord's tanhValueOf:aList)

set bList to aArray's ASify() as list

--> {0.099667994625, 0.197375320225, 0.291312612452,
0.379948962255, 0.46211715726, 0.537049566998, 0.604367777117,
0.664036770268, 0.716297870199}

method	<code>sinhValueOf:</code> ①
Introduction	Pass a single number or a list of numbers. Results must be coerced using "as real" or "as list". Angles are in radians, and errors are returned as missing value.
kind	Trigonometry
Input	①AppleScript number (real) or AppleScript list
Output	NSNumber or NSArray
Notes	

--Sample Code

-- Created 2014-11-30 by Takaaki Naganoya

-- 2014 Piyomaru Software

use AppleScript version "2.4"

use framework "Foundation"

use framework "ASObjCExtras"

use scripting additions

--Real Value

set aVal to current application's SMSFord's sinhValueOf:-0.5

set bVal to aVal's ASify() as real

--> -0.521095305494

--List of Real Value

set aList to {0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9}

set aArray to (current application's SMSFord's sinhValueOf:aList)

set bList to aArray's ASify() as list

--> {0.10016675002, 0.201336002541, 0.304520293447, 0.410752325803,
0.521095305494, 0.636653582148, 0.75858370184, 0.888105982188,
1.026516725708}

method	<code>coshValueOf:</code> ①
Introduction	Pass a single number or a list of numbers. Results must be coerced using "as real" or "as list". Angles are in radians, and errors are returned as missing value.
kind	Trigonometry
Input	①AppleScript number (real) or AppleScript list
Output	NSNumber or NSArray
Notes	

--Sample Code

-- Created 2014-11-30 by Takaaki Naganoya

-- 2014 Piyomaru Software

use AppleScript version "2.4"

use framework "Foundation"

use framework "ASObjCExtras"

use scripting additions

--Real Value

set aVal to current application's SMSFord's coshValueOf:-0.5

set bVal to aVal's ASify() as real

--> 1.127625965206

--List of Real Value

set aList to {0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9}

set aArray to (current application's SMSFord's coshValueOf:aList)

set bList to aArray's ASify() as list

-->{1.005004168056, 1.020066755619, 1.045338514129,
1.081072371838, 1.127625965206, 1.185465218242,
1.255169005631, 1.337434946305, 1.433086385449}

method	atanhValueOf:①
Introduction	Pass a single number or a list of numbers. Results must be coerced using "as real" or "as list". Angles are in radians, and errors are returned as missing value.
kind	Trigonometry
Input	①AppleScript number (real) or AppleScript list
Output	NSNumber or NSArray
Notes	

--Sample Code

-- Created 2014-11-30 by Takaaki Naganoya

-- 2014 Piyomaru Software

use AppleScript version "2.4"

use framework "Foundation"

use framework "ASObjCExtras"

use scripting additions

--Real Value

set aVal to current application's SMSFord's atanhValueOf:-0.5

set bVal to aVal's ASify() as real

--> -0.549306144334

--List of Real Value

set aList to {0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9}

set aArray to (current application's SMSFord's atanhValueOf:aList)

set bList to aArray's ASify() as list

-->{0.100335347731, 0.202732554054, 0.309519604203,
0.423648930194, 0.549306144334, 0.69314718056, 0.867300527694,
1.098612288668, 1.472219489583}

method	asinhValueOf:①
Introduction	Pass a single number or a list of numbers. Results must be coerced using "as real" or "as list". Angles are in radians, and errors are returned as missing value.
kind	Trigonometry
Input	①AppleScript number (real) or AppleScript list
Output	NSNumber or NSArray
Notes	

--Sample Code

-- Created 2014-11-30 by Takaaki Naganoya

-- 2014 Piyomaru Software

use AppleScript version "2.4"

use framework "Foundation"

use framework "ASObjCEExtras"

use scripting additions

--Real Value

set aVal to current application's SMSFord's asinhValueOf:-0.5

set bVal to aVal's ASify() as real

--> -0.48121182506

--List of Real Value

set aList to {0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9}

set aArray to (current application's SMSFord's asinhValueOf:aList)

set bList to aArray's ASify() as list

--> {0.099834078899, 0.198690110349, 0.295673047563,
0.390035319771, 0.48121182506, 0.568824898732, 0.652666566082,
0.732668256045, 0.808866935653}

method	acoshValueOf:①
Introduction	Pass a single number or a list of numbers. Results must be coerced using "as real" or "as list". Angles are in radians, and errors are returned as missing value.
kind	Trigonometry
Input	①AppleScript number (real) or AppleScript list
Output	NSNumber or NSArray
Notes	

--Sample Code

-- Created 2014-11-30 by Takaaki Naganoya

-- 2014 Piyomaru Software

use AppleScript version "2.4"

use framework "Foundation"

use framework "ASObjCExtras"

use scripting additions

--Real Value

set aVal to current application's SMSFord's acoshValueOf:20

set bVal to aVal's ASify() as real

--> 3.688253867361

--List of Real Value

set aList to {10, 20, 30, 40, 50, 60, 70, 80, 90}

set aArray to (current application's SMSFord's acoshValueOf:aList)

set bList to aArray's ASify() as list

--> {2.993222846126, 3.688253867361, 4.094066668632, 4.38187034804,
4.605070170985, 4.787422291103, 4.941591398296, 5.075134750445,
5.192925985264}

method	logValueOf:①
Introduction	Pass a single number or a list of numbers. Results must be coerced using "as real" or "as list". Angles are in radians, and errors are returned as missing value.
kind	Trigonometry
Input	①AppleScript number (real) or AppleScript list
Output	NSNumber or NSArray
Notes	

--Sample Code

-- Created 2014-11-30 by Takaaki Naganoya

-- 2014 Piyomaru Software

use AppleScript version "2.4"

use framework "Foundation"

use framework "ASObjCExtras"

use scripting additions

--Real Value

set aVal to current application's SMSFord's logValueOf:10

set bVal to aVal's ASify() as real

--> 2.302585092994

--List of Real Value

set aList to {10, 20, 30, 40, 50, 60, 70, 80, 90}

set aArray to (current application's SMSFord's logValueOf:aList)

set bList to aArray's ASify() as list

--> {2.302585092994, 2.995732273554, 3.401197381662,
3.688879454114, 3.912023005428, 4.094344562222,
4.248495242049, 4.382026634674, 4.49980967033}

method	log10ValueOf:①
Introduction	Pass a single number or a list of numbers. Results must be coerced using "as real" or "as list". Angles are in radians, and errors are returned as missing value.
kind	Trigonometry
Input	①AppleScript number (real) or AppleScript list
Output	NSNumber or NSArray
Notes	

--Sample Code

-- Created 2014-11-30 by Takaaki Naganoya

-- 2014 Piyomaru Software

use AppleScript version "2.4"

use framework "Foundation"

use framework "ASObjCExtras"

use scripting additions

--Real Value

set aVal to current application's SMSFord's log10ValueOf:10

set bVal to aVal's ASify() as real

--> 1.0

--List of Real Value

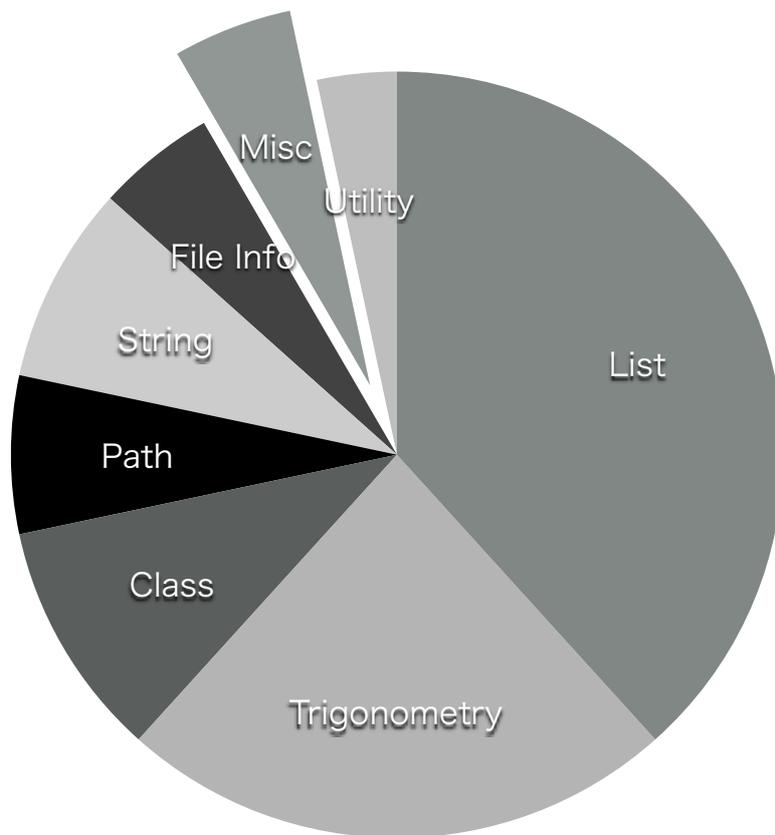
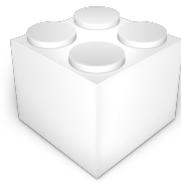
set aList to {10, 20, 30, 40, 50, 60, 70, 80, 90}

set aArray to (current application's SMSFord's log10ValueOf:aList)

set bList to aArray's ASify() as list

-->{1.0, 1.301029995664, 1.47712125472, 1.602059991328,
1.698970004336, 1.778151250384, 1.845098040014,
1.903089986992, 1.954242509439}

ASObjCExtras.framework
Misc. methods



method	metadataFromImage:① error:②
Introduction	Pass an alias, file, HFS path, POSIX path or NSURL for an image file, and receive back a dictionary/record of the metadata.
kind	Misc.
Input	①AppleScript alias ②AppleScript missing value or reference
Output	NSDictionary
Notes	

-- Sample Code

-- Created 2014-11-15 by Takaaki Naganoya

-- 2014 Piyomaru Software

use AppleScript version "2.4"

use scripting additions

use framework "Foundation"

use framework "ASObjCEExtras"

set anAlias to choose file

set b to (current application's SMSFord's metadataFromImage:anAlias lerror:(missing value))

set c to b's ASify() as record

--Jpeg

--> {{{jif}}:XDensity:1, YDensity:1, JFIFVersion:{1, 0, 1}, DensityUnit:0}, ColorModel:"RGB", PixelHeight:1920, PixelWidth:1080, Depth:8}

--Jpeg (From digital Camera 1)

--> {{{tiff}}:ResolutionUnit:2, DateTime:"2004:10:22 22:53:49", XResolution:72.0, ImageDescription:" ", Orientation:1, YResolution:72.0, Model:"CYBERSHOT U", Imake:"SONY"}, ProfileName:"sRGB IEC61966-2.1", DPIWidth:72.0, I{{exif}}:DateOriginal:"2004:10:22 22:53:49", ComponentsConfiguration:{1, 2, 3, 0}, MaxApertureValue:2.9375, ExposureBiasValue:0.0, FNumber:2.8, CompressedBitsPerPixel:2.0, FocalLength:5.0, SceneType:1, FileSource:3, SceneCaptureType:0, ColorSpace:1, PixelYDimension:960, WhiteBalance:0, LightSource:0, FlashPixVersion:{1, 0}, DateTimeDigitized:"2004:10:22 22:53:49", ISOSpeedRatings:{160}, ExposureMode:0, PixelXDimension:1280, ExifVersion:{2, 2}, CustomRendered:0, ExposureProgram:2, Flash:15, ExposureTime:0.025, MeteringMode:2}, Depth:8, ColorModel:"RGB", PixelHeight:960, PixelWidth:1280, Orientation:1, DPIHeight:72.0}

--Jpeg (From Digital Camera 2)

--> {{{tiff}}:YResolution:72.0, ResolutionUnit:2, Software:"CX3 Firmware ", DateTime:"2010:08:05 16:12:20", XResolution:72.0, ImageDescription:"Exif_JPEG_PICTURE ", Orientation:1, Copyright:" ", Model:"CX3 ", Imake:"RICOH "}, ProfileName:"sRGB IEC61966-2.1", DPIWidth:72.0, I{{exif}}:DateOriginal:"2010:08:05 16:12:20", ComponentsConfiguration:{1, 2, 3, 0}, MaxApertureValue:3.5, BrightnessValue:8.1, ExposureBiasValue:1.0, FNumber:5.1, CompressedBitsPerPixel:2.0, FocalLength:23.6, SceneCaptureType:0, ApertureValue:4.7, Sharpness:0, ColorSpace:1, PixelYDimension:480, WhiteBalance:0, UserComment:" ", LightSource:0, FlashPixVersion:{1, 0}, DateTimeDigitized:"2010:08:05 16:12:20", ISOSpeedRatings:{400}, ExposureMode:0, ExifVersion:{2, 2, 1}, PixelXDimension:640, ExposureProgram:2, Flash:16, ExposureTime:0.001754385965, MeteringMode:5}, Depth:8, ColorModel:"RGB", PixelHeight:480, PixelWidth:640, Orientation:1, DPIHeight:72.0}

--bmp

--> {DPIHeight:99.974395751953, DPIWidth:99.974395751953, IsIndexed:true, ColorModel:"RGB", PixelHeight:240, Depth:8, PixelWidth:221}

```

--tiff
--> {ProfileName:"sRGB IEC61966-2.1", I{tiff}I:{Compression:5, Orientation:1, PhotometricInterpretation:2}, Depth:
      8, ColorModel:"RGB", PixelHeight:900, HasAlpha:true, Orientation:1, PixelWidth:1440}

--giff
--> {Depth:8, ColorModel:"RGB", PixelHeight:250, I{gif}I:{DelayTime:0.10000000149, UnclampedDelayTime:0.0},
      ProfileName:"sRGB IEC61966-2.1", PixelWidth:300}

--animation giff
-->{ProfileName:"sRGB IEC61966-2.1", I{gif}I:{DelayTime:5.0, UnclampedDelayTime:5.0}, ColorModel:"RGB",
      PixelHeight:110, Depth:8, HasAlpha:true, PixelWidth:700}

--PSD
--> {I{tiff}I:{DateTime:"2014:11:06 18:16:14", Orientation:1, ResolutionUnit:2, Software:"Adobe Photoshop CC
      2014 (Macintosh)", XResolution:72.0, YResolution:72.0}, I{exif}I:{PixelXDimension:700,
      DateTimeDigitized:"2014:09:18 16:01:10", PixelYDimension:110, ColorSpace:1}, HasAlpha:true, PixelHeight:
      110, Depth:8, I{8bim}I:{I{version}I:1, LayerNames:{"Layer #3", "Layer #2", "Layer #1_copy1", "Email:
      xxxxx@xxxxxx.com", "Layer #1_copy2"}}, ProfileName:"sRGB IEC61966-2.1", DPIWidth:72.0, DPIHeight:72.0,
      ColorModel:"RGB", Orientation:1, PixelWidth:700}

--icns
--> {ProfileName:"generic RGB Profile", DPIHeight:72.0, DPIWidth:72.0, ColorModel:"RGB", PixelHeight:512,
      kCGImagePropertyICNSIndexSelector:"ic09", Depth:8, HasAlpha:true, PixelWidth:512}

--DNG (Raw File)
--> {I{tiff}I:{Compression:1, PhotometricInterpretation:32803, DateTime:"2008:11:03 12:32:32", Orientation:1,
      Model:"GR Digital ", ImakeI:"RICOH ", Copyright:"(C) by GR Digital User ", I{exif}I:
      {DateTimeOriginal:"2008:11:03 12:32:32", MaxApertureValue:2.4, ExposureBiasValue:1.0,
      BrightnessValue:-0.3, FNumber:2.4, FocalLength:5.9, SceneCaptureType:0, ShutterSpeedValue:3.70044,
      ApertureValue:2.5, Sharpness:1, ColorSpace:1, CFAPattern:{0, 1, 1, 2}, WhiteBalance:0, UserComment:"
      ", LightSource:0, DateTimeDigitized:"2008:11:03 12:32:32", ISOSpeedRatings:{154}, ExposureMode:0,
      ExifVersion:{2, 2, 1}, ExposureProgram:3, Flash:16, ExposureTime:0.076923076923, MeteringMode:5}, I
      {iptc}I:{StarRating:0.0, CopyrightNotice:"(C) by GR Digital User ", ProfileName:"Adobe RGB (1998)",
      PixelHeight:2448, I{dng}I:{DNGBackwardVersion:{1, 0}, UniqueCameraModel:"Ricoh GR Digital", DNGVersion:
      {1, 0}}, I{exifaux}I:{LensModel:"5.9 mm", LensInfo:{5.900000095367, 5.900000095367, 0.0, 0.0}},
      ColorModel:"RGB", Depth:16, Orientation:1, PixelWidth:3264}

```

method	infoForFile:①
Introduction	Takes an alias, file, HFS path, POSIX path or NSURL and returns a dictionary/record. The labels can be found by looking up NSURL; the prefix "NSURL" and suffix "Key" are removed, and the initial character made lowercase.
kind	Misc.
Input	①AppleScript alias
Output	NSDictionary
Notes	

```
-- Sample Code
-- Created 2014-11-15 by Takaaki Naganoya
-- 2014 Piyomaru Software
```

```
use AppleScript version "2.4"
use scripting additions
use framework "Foundation"
use framework "ASObjCExtras"
```

```
set anAlias to choose file
set b to (current application's SMSFord's infoForFile:anAlias)
set c to b's ASify() as record
--> {isAliasFile:false, fileResourceType:"NSURLFileResourceTypeRegular",
isExcludedFromBackup:false, contentModificationDate:date "Friday, January
9, 2015 at 8:24:50 PM", isReadable:true, isSystemImmutable:false,
isSymbolicLink:false, isExecutable:false, parentDirectoryURL:«class ocid» id
«data optr0000000070109B40EA7F0000», hasHiddenExtension:true,
labelNumber:0, isWritable:true, lpath!:"/Users/maro/Desktop/Screen Shot
2015-01-09 at 8.24.50 PM.png", isPackage:false, lname!"Screen Shot
2015-01-09 at 8.24.50 PM.png", isDirectory:false, localizedName:"Screen
Shot 2015-01-09 at 8.24.50 PM", localizedTypeDescription:"Portable
Network Graphics image", linkCount:1, attributeModificationDate:date "Friday,
January 9, 2015 at 8:24:50 PM", creationDate:date "Friday, January 9, 2015
at 8:24:50 PM", isHidden:false, isRegularFile:true, contentAccessDate:date
"Friday, January 9, 2015 at 8:24:56 PM", isUserImmutable:false}
```

method	sizeInfoForFile:①
Introduction	Takes an alias, file, HFS path, POSIX path or NSURL and returns a dictionary/record.
kind	Misc.
Input	①AppleScript alias
Output	NSDictionary
Notes	<p>For files, the values are returned like this: {totalFileSize:94927, totalFileAllocatedSize:98304, fileAllocatedSize:98304, fileSize:94927}</p> <p>The "total" values include metadata; "Allocated" means disk space used. Finder reports totalFileSize.</p> <p>For directories and packages, only the total values are returned: {totalFileSize:271230605, totalFileAllocatedSize:286121984}.</p>

```
-- Sample Code
-- Created 2014-11-15 by Takaaki Naganoya
-- 2014 Piyomaru Software
```

```
use AppleScript version "2.4"
use scripting additions
use framework "Foundation"
use framework "ASObjCExtras"
```

```
set anAlias to choose file
set b to (current application's SMSFord's sizeInfoForFile:anAlias) as record
--> {totalFileSize:199432, totalFileAllocatedSize:200704, fileAllocatedSize:
    200704, fileSize:199432}
```

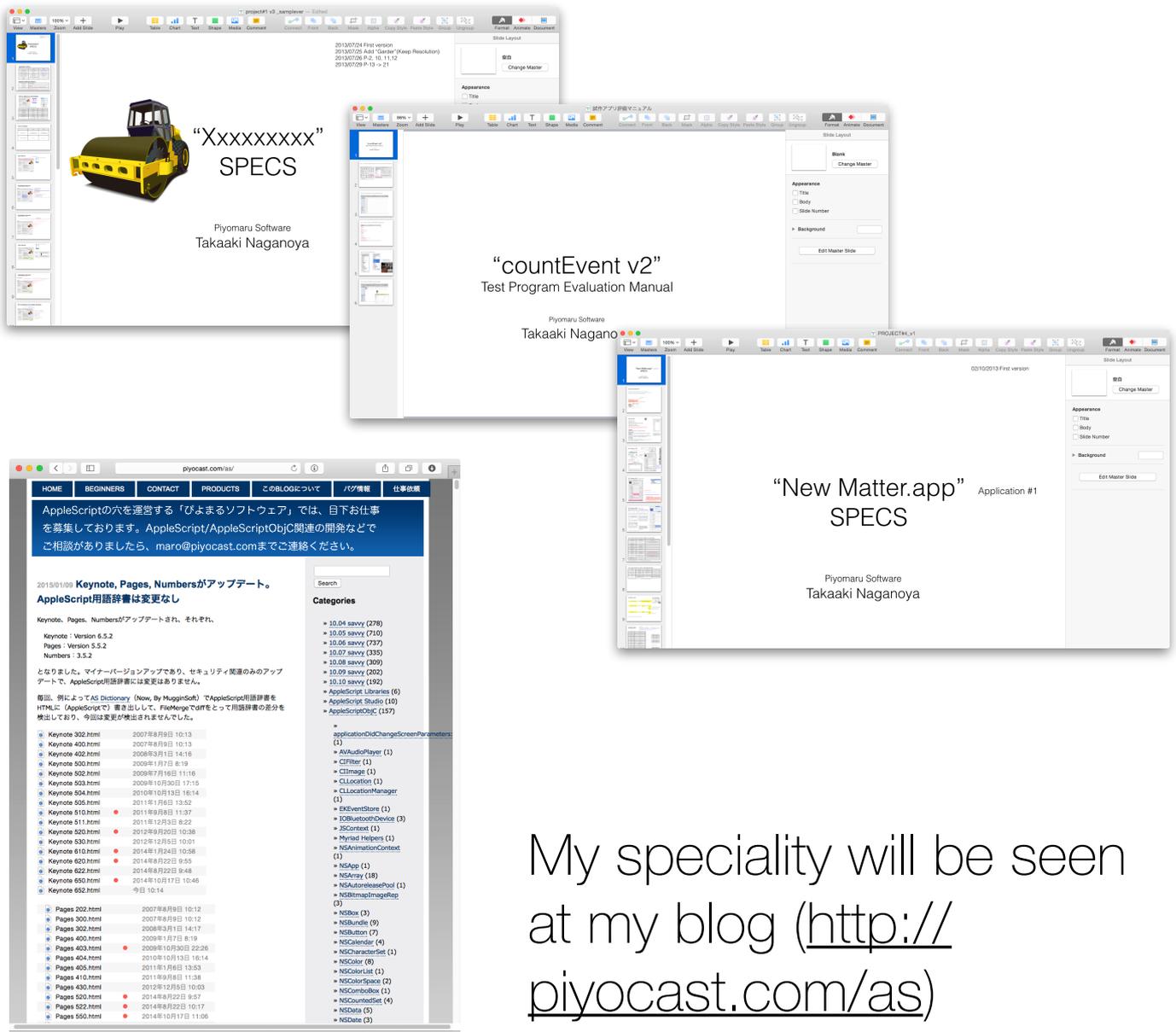
Errata & Document modification History

January 14, 2015 First Release

Piyomaru Software

A Professional AppleScript Developer in Tokyo, Japan

We develop applications/scripts with **visual specification documents!**



My speciality will be seen at my blog (<http://piyocast.com/as>)

mailto: maro@piyocast.com

twitter: @Piyomaru

iMessage: piyomarusoft@mac.com