translation

OBJECTIVEC TO APPLESCRIPT-OBJECTIVEC (ASOC)

SMALL PREVIEW

This is a free book

Original idea by Jonathan H Burgos S.
Punta Arenas, Chile

Github: https://github.com/jonathanburgossaldivia
Donation via PayPal :
jonathanburgoss@outlook.com
Contact: jonathanburgoss@outlook.com

INTRODUCTION

With the intention of developing more complex applications in Xcode, I make this compilation of code and translation from ObjectiveC to AppleScript-ObjectiveC, there are some practical examples to use in Xcode, in the future I hope to place examples for use with buttons, combox, etcetera , this book has no profit, I hope it is useful…

A timer that fires after a certain time interval has elapsed, sending a specified message to a target object.

NSTIMER

PERFORMSELECTOR:WITHOBJECT:AFTERDELAY:

Invokes a method of the receiver on the current thread using the default mode after a delay.

Parameters

- ***aSelector:*** A Selector that identifies the method to invoke. The method should not have a significant return value and should take a single argument of type id, or no arguments.

- ***anArgument:*** The argument to pass to the method when it is invoked. Pass nil if the method does not take an argument.

- ***delay:*** The minimum time before which the message is sent. Specifying a delay of 0 does not necessarily cause the selector to be performed immediately. The selector is still queued on the thread’s run loop and performed as soon as possible.

URL: https://developer.apple.com/documentation/objectivec/nsobject/1416176-performselector

ObjectiveC:

```
- (void)performSelector:(SEL)aSelector

        withObject:(id)anArgument

        afterDelay:(NSTimeInterval)delay;
```

AppleScript: performSelector_withObject_afterDelay_((SEL)aSelector”,(id)anArgument, (NSTimeInterval)delay)

AppleScript Example:

```
property NSTimer : class "NSTimer" of current application

on LsTest_()

set ListTest to (do shell script "ls -l")

log ListTest

performSelector_withObject_afterDelay_("LsTest:", missing value, 1.0)

end LsTest_

on applicationWillFinishLaunching_(aNotification)

performSelector_withObject_afterDelay_("LsTest:", missing value, 1.0)

end applicationWillFinishLaunching_
```

PERFORMSELECTOR:WITHOBJECT:AFTERDELAY:INMODES:

Invokes a method of the receiver on the current thread using the specified modes after a delay.

Parameters

- ***aSelector:*** A Selector that identifies the method to invoke. The method should not have a significant return value and should take a single argument of type id, or no arguments.

- ***anArgument:*** The argument to pass to the method when it is invoked. Pass nil if the method does not take an argument.

- ***delay:*** The minimum time before which the message is sent. Specifying a delay of 0 does not necessarily cause the selector to be performed immediately. The selector is still queued on the thread’s run loop and performed as soon as possible.

- ***modes:*** An array of strings that identify the modes to associate with the timer that performs the selector. This array must contain at least one string. If you specify nil or an empty array for this parameter, this method returns without performing the specified selector. For information about run loop modes, see Run Loops in Threading Programming Guide.

URL: https://developer.apple.com/documentation/objectivec/nsobject/1415652-performselector?language=objc

ObjectiveC: - (void)performSelector:(SEL)aSelector

```
        withObject:(id)anArgument

        afterDelay:(NSTimeInterval)delay
        inModes:(NSArray<NSRunLoopMode> *)modes;
```

AppleScript: performSelector_withObject_afterDelay_((SEL)aSelector, (id)anArgument, (NSTimeInterval)delay, (NSArray<NSRunLoopMode> *)modes)

PERFORMSELECTORONMAINTHREAD:WITHOBJECT:WAITUNTILDONE:

Invokes a method of the receiver on the main thread using the default mode.

Parameters

- ***aSelector:*** A Selector that identifies the method to invoke. The method should not have a significant return value and should take a single argument of type id, or no arguments.

- ***arg:*** The argument to pass to the method when it is invoked. Pass nil if the method does not take an argument.

- ***wait:*** A Boolean that specifies whether the current thread blocks until after the specified selector is performed on the receiver on the main thread. Specify YES to block this thread; otherwise, specify NO to have this method return immediately. If the current thread is also the main thread, and you specify YES for this parameter, the message is delivered and processed immediately.

URL: https://developer.apple.com/documentation/objectivec/nsobject/1414900-performselectoronmainthread?language=objc

ObjectiveC: - (void)performSelectorOnMainThread:(SEL)aSelector

withObject:(id)arg

waitUntilDone:(BOOL)wait;

AppleScript: performSelectorOnMainThread_withObject_waitUntilDone_((SEL)aSelector, (id)arg, (BOOL)wait)

PERFORMSELECTORONMAINTHREAD:WITHOBJECT:WAITUNTILDONE:MODES:

Invokes a method of the receiver on the main thread using the specified modes.

Parameters

- ***aSelector:*** A Selector that identifies the method to invoke. The method should not have a significant return value and should take a single argument of type id, or no arguments.

- ***arg:*** The argument to pass to the method when it is invoked. Pass nil if the method does not take an argument.

- ***wait:*** A Boolean that specifies whether the current thread blocks until after the specified selector is performed on the receiver on the main thread. Specify YES to block this thread; otherwise, specify NO to have this method return immediately. If the current thread is also the main thread, and you pass YES, the message is performed immediately, otherwise the perform is queued to run the next time through the run loop.

- ***array:*** An array of strings that identifies the modes in which it is permissible to perform the specified selector. This array must contain at least one string. If you specify nil or an empty array for this parameter, this method returns without performing the specified selector. For information about run loop modes, see Run Loops in Threading Programming Guide.

URL: https://developer.apple.com/documentation/objectivec/nsobject/1411637-performselectoronmainthread?language=objc

ObjectiveC: - (void)performSelectorOnMainThread:(SEL)aSelector

```
                    withObject:(id)arg

                  waitUntilDone:(BOOL)wait
```

```
                    modes:(NSArray<NSString *> *)array;
```

AppleScript:

```
performSelectorOnMainThread_withObject_waitUntilDone_modes_((SEL)aSelector,
(id)arg, (BOOL)wait, (NSArray<NSString *> *)array)
```

PERFORMSELECTOR:ONTHREAD:WITHOBJECT:WAITUNTILDONE:

Invokes a method of the receiver on the specified thread using the default mode.

Parameters

- ***aSelector:*** A Selector that identifies the method to invoke. The method should not have a significant return value and should take a single argument of type id, or no arguments.
- ***thread:*** The thread on which to execute aSelector.
- ***arg:*** The argument to pass to the method when it is invoked. Pass nil if the method does not take an argument.
- ***wait:*** A Boolean that specifies whether the current thread blocks until after the specified selector is performed on the receiver on the specified thread. Specify YES to block this thread; otherwise, specify NO to have this method return immediately. If the current thread and target thread are the same, and you specify YES for this parameter, the selector is performed immediately on the current thread. If you specify NO, this method queues the message on the thread's run loop and returns, just like it does for other threads. The current thread must then dequeue and process the message when it has an opportunity to do so.

URL: https://developer.apple.com/documentation/objectivec/nsobject/1414476-performselector?language=objc

ObjectiveC: - (void)performSelector:(SEL)aSelector

```
       onThread:(NSThread *)thr

      withObject:(id)arg

    waitUntilDone:(BOOL)wait;
```

AppleScript: performSelector_onThread_withObject_waitUntilDone_((SEL)aSelector, (NSThread *)thr, (id)arg, (BOOL)wait)

PERFORMSELECTOR:ONTHREAD:WITHOBJECT:WAITUNTILDONE:MODES:

Invokes a method of the receiver on the specified thread using the specified modes.

Parameters

- ***aSelector:*** A Selector that identifies the method to invoke. It should not have a significant return value and should take a single argument of type id, or no arguments.

- ***thread:*** The thread on which to execute aSelector. This thread represents the target thread.

- ***arg:*** The argument to pass to the method when it is invoked. Pass nil if the method does not take an argument.

- ***wait:*** A Boolean that specifies whether the current thread blocks until after the specified selector is performed on the receiver on the specified thread. Specify YES to block this thread; otherwise, specify NO to have this method return immediately. If the current thread and target thread are the same, and you specify YES for this parameter, the selector is performed immediately. If you specify NO, this method queues the message and returns immediately, regardless of whether the threads are the same or different.

- ***array:*** An array of strings that identifies the modes in which it is permissible to perform the specified selector. This array must contain at least one string. If you specify nil or an empty array for this parameter, this method returns without performing the specified selector. For information about run loop modes, see Run Loops in Threading Programming Guide.

URL: https://developer.apple.com/documentation/objectivec/nsobject/1417922-performselector?language=objc

ObjectiveC: - (void)performSelector:(SEL)aSelector

```
   onThread:(NSThread *)thr

  withObject:(id)arg

waitUntilDone:(BOOL)wait

     modes:(NSArray<NSString *> *)array;
```

AppleScript:
performSelector_onThread_withObject_waitUntilDone_modes_((SEL)aSelector, (NSThread *)thr, (id)arg, (BOOL)wait, (NSArray<NSString *> *)array)

PERFORMSELECTORINBACKGROUND:WITHOBJECT:

Invokes a method of the receiver on a new background thread.

Parameters

- ***aSelector:*** A Selector that identifies the method to invoke. The method should not have a significant return value and should take a single argument of type id, or no arguments.

- ***arg:*** The argument to pass to the method when it is invoked. Pass nil if the method does not take an argument.

URL: https://developer.apple.com/documentation/objectivec/nsobject/1412390-performselectorinbackground?language=objc

ObjectiveC: - (void)performSelectorInBackground:(SEL)aSelector

withObject:(id)arg;

AppleScript: performSelectorInBackground_withObject_((SEL)aSelector, (id)arg)

CANCELPREVIOUSPERFORMREQUESTSWITHTARGET:

Cancels perform requests previously registered with the performSelector:withObject:afterDelay: instance method.

Parameters

- ***aTarget:*** The target for requests previously registered with the performSelector:withObject:afterDelay: instance method.

Discussion

All perform requests having the same target aTarget are canceled. This method removes perform requests only in the current run loop, not all run loops.

URL: https://developer.apple.com/documentation/objectivec/nsobject/1417611-cancelpreviousperformrequestswit?language=objc

ObjectiveC: + (void)cancelPreviousPerformRequestsWithTarget:(id)aTarget;

AppleScript: cancelPreviousPerformRequestsWithTarget_((id)aTarget)

CANCELPREVIOUSPERFORMREQUESTSWITHTARGET:SELECTOR:OBJECT:

Cancels perform requests previously registered with performSelector:withObject:afterDelay:

Parameters

- ***aTarget:*** The target for requests previously registered with the performSelector:withObject:afterDelay: instance method

- ***aSelector:*** The Selector for requests previously registered with the performSelector:withObject:afterDelay: instance method.

- ***anArgument:*** The argument for requests previously registered with the performSelector:withObject:afterDelay: instance method. Argument equality is determined using isEqual:, so the value need not be the same object that was passed originally. Pass nil to match a request for nil that was originally passed as the argument.

URL: https://developer.apple.com/documentation/objectivec/nsobject/1410849-cancelpreviousperformrequestswit?language=objc

ObjectiveC:

```
+ (void)cancelPreviousPerformRequestsWithTarget:(id)aTarget
                                       selector:(SEL)aSelector
                                         object:(id)anArgument;
```

AppleScript: cancelPreviousPerformRequestsWithTarget_ selector_ object_((id)aTarget, (SEL)aSelector, (id)anArgument)

An object representing a subprocess of the current process.

NSTASK

LAUNCHEDTASKWITHLAUNCHPATH:ARGUMENTS:

Creates and launches a task with a specified executable and arguments.

Parameters

- ***path:*** The path to the executable.

- ***arguments:*** An array of NSString objects that supplies the arguments to the task. If arguments is nil, an NSInvalidArgumentException is raised.

URL: https://developer.apple.com/documentation/foundation/nstask/1414375-launchedtaskwithlaunchpath?language=objc

ObjectiveC:+ (NSTask *)launchedTaskWithLaunchPath:(NSString *)path

arguments:(NSArray<NSString *> *)arguments;

AppleScript: launchedTaskWithLaunchPath_arguments_((NSString *)path, (NSArray<NSString *> *)arguments)

AppleScript Example:

on NSTaskLsExample_(sender)

set lsTask to current application's NSTask's launchedTaskWithLaunchPath_arguments_("/bin/ls",{"-l", "/Users/"})

```
end NSTaskLsExample_

on applicationWillFinishLaunching_(aNotification)

my NSTaskLsExample_(me)

end applicationWillFinishLaunching_
```