# Getting AppleScript return value in Obj-C



I'm using some AppleScript in my Obj-C cocoa project to control QuickTime player (play, pause, stop, jog forward and back etc.) with great success, though my knowledge of AppleScript is very limited. However, what I want most of all is the movie's 'Current Time' offset to convert into time-stamps for writing a subtitle script.

The following simple method shows the precise current position in (float) seconds in a dialog, but I'd really like the AppleScript to ***return*** me a ***variable*** that I can use in the rest of app. How could I modify the code below to do that? Is it even possible to access this value? Thanks a million in advance :-)

```
PlayTime:(id)sender

ptString=[NSString stringWithFormat:
e of current frame... (works perfectly)!
lication \"QuickTime Player\"\n"
timeScale to 600\n"
curr_pos to current time of movie 1/timeScale\n"
lay dialog curr_pos\n" // ...not in a practical form to use
\n"];

errorDict= nil;
*appleScriptObject=[[NSAppleScript alloc]
iptString];
scriptor *eventDescriptor=[appleScriptObject
ror:   &errorDict];
errors here (snipped for brevity)
ject release]; // can I retain this?
```

objective-c applescript

asked Jul 24 '11 at 2:08

Bender
172 2 13

## 3 Answers

Here's the appropriate AppleScript that you'd want to run:

```
property timeScale : 600

set currentPosition to missing value

tell application "QuickTime Player"
    set currentPosition to (current time of document 1) / timeScale
end tell

return currentPosition
```

In case you're not familiar with it, `property` is a way to specify a global variable in AppleScript. Also, `missing value` is the AppleScript equivalent of `nil` in Objective-C. So, this script first defines a variable named `currentPosition`, and sets the value to `missing value`. It then enters the `tell` block which, if it succeeds, will alter the `currentPosition` variable. Then, outside of the tell block, it returns the `currentPosition` variable.

In the Objective-C code, when you create an `NSAppleScript` with the above code, its `-executeAndReturnError:` method will return the `currentPosition` variable in an `NSAppleScriptEventDescriptor`.

```
-(IBAction)currentPlayTime:(id)sender {

    NSDictionary *error = nil;

    NSMutableString *scriptText = [NSMutableString stringWithString:@"property
timeScale : 600\n"];
    [scriptText appendString:@"set currentPosition to missing value\n"];
    [scriptText appendString:@"tell application \"QuickTime Player\"\n "];
    [scriptText appendString:@"set currentPosition to (current time of document 1)
/ timeScale\n"];
    [scriptText appendString:@"end tell\n"];
    [scriptText appendString:@"return currentPosition\n"];

    NSAppleScript *script = [[[NSAppleScript alloc] initWithSource:scriptText]
autorelease];

    NSAppleEventDescriptor *result = [script executeAndReturnError:&error];

    NSLog(@"result == %@", result);

    DescType descriptorType = [result descriptorType];

    NSLog(@"descriptorType == %@", NSFileTypeForHFSTypeCode(descriptorType));

    // returns a double

    NSData *data = [result data];
    double currentPosition = 0;

    [data getBytes:&currentPosition length:[data length]];

    NSLog(@"currentPosition == %f", currentPosition);
}
```

You can extract the contents of the `NSAppleEventDescriptor` as shown above.

Using the Scripting Bridge framework does have a slight learning curve, but would allow working with native types such as `NSNumber`s rather than having to go the somewhat "messier" route of extracting the raw bytes out of AppleEvent descriptor.

answered Jul 24 '11 at 6:12

NSGod
**16.9k** 1 29 47

Hi NSGod. Yeess! It works like a charm :-) As you suspected, I wasn't familiar with AppleScript's 'property' variable and know very little about AS in general, except for using the odd crude snippet in my code which has been getting the job done. Also (like Yuji said) Scripting Bridge would've been a simple elegant way to go and I'd have been able to use the familiar data types, but I can't seem to apply it to QT Player unless I'm overlooking something (or doing something dumb). I'll keep searching for a solution to this on the 'net. Thanks again in the meantime. Appreciate your input :-) – Bender Jul 24 '11 at 10:04



Use Scripting Bridge. This is a bridge between AppleScript and Objective-C, and other applications (e.g. QuickTime Player) is represented as an Objectve-C object in your code. So, you don't have to construct AppleScript code by hand.

Some say AppScript is better than Scripting Bridge.

answered Jul 24 '11 at 2:28

Yuji
**29.4k** 3 45 77

Hi Yuji. Scripting Bridge seemed like the most straightforward and elegant solution. Managed to run an iTunes test but amazingly (unless I'm doing something wrong), it doesn't seem to have QT Player capabilities! Can't create the necessary header file with sdef/stp... and Apple's "ScriptingBridgeConcepts.pdf" doesn't mention QT Player once. No info on the net about Scripting Bridge and QT either. I'll also look into AppScript but that'll take another few days yet. I appreciate your help :-) – Bender Jul 24 '11 at 8:56

NSAppleEventDescriptor has some methods to convert to some objective-C types, if you go to my site and download the NDScript project, it has a category of NSAppleEventDescriptor which adds a lot more methods for coercion to Objective-C type. You can use that category without the rest of the project. http://homepage.mac.com/nathan_day/pages/source.xml

answered Jul 24 '11 at 6:15

Nathan Day
**4,019** 2 10 31

Hi Nathan. Thanks a bundle for the link/download to your NDScript project and I'll tinker with it this coming week. Looks like you've put a huge amount of work into that... Appreciate your input :-) – Bender Jul 24 '11 at 9:02

I will split the NSAppleEventDescriptor category out and put it up on GitHub as a lot of people have use for that category without of the rest of NDScript. A lot of the reasons for NDScript are no longer applicable with NSAppleScript and the Scripting Bridge, you can use it to run scripts in a thread, I was working on it for an Application I has written that I want to allow users to use AppleScripts to override internal class and change how it worked. Keep an eye on my GitHub repository, I will have it up this week GitHub – Nathan Day Jul 24 '11 at 12:30

I have now added a separate repository for the NSAppleEventDescriptor category on my GetHub hub if you are still interested. [nathanday][github.com/nathanday} – Nathan Day Jul 30 '11 at 14:36