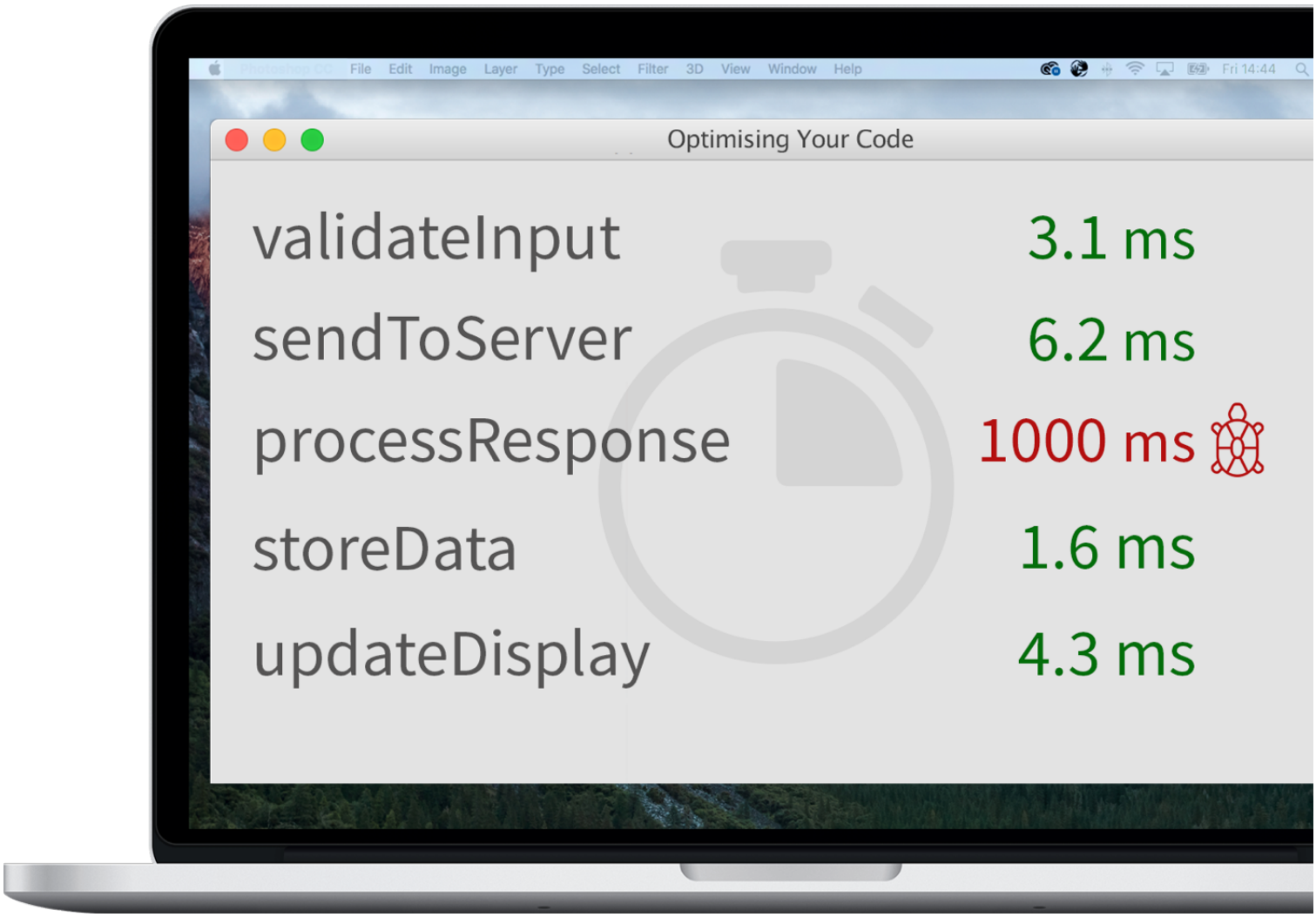


The LiveCode Script Profiler

by Elanor Buchanan (https://livecode.com/author/elanor/) on February 24, 2017 4 comments (https://li livecode-script-profiler/#comments)

Share on Facebook (https://www.facebook.com/sharer.php?u=https%3A%2F%2Flivecoc livecode-script-profiler%2F)

Share on Twitter (https://twitter.com/intent/tweet? text=The%20LiveCode%20Script%20Profiler&url=https://livecode.com/the-livecode-scri

LiveCode's new Script Profiler, available in the LiveCode Business edition, can help you i
costly lines of code in a stack. This can help you optimise your code by identifying ineffi

To use the Script Profiler you open a stack, start the profiler and use the stack. When eac
the stack is executed the execution time will be recorded. When the profiler is stopped a

In this example we will look at an area that can often benefit from some optimisation, loo
This example just uses a simple array with 10000 values in it. The code loops across the
all the values stored in the array.

Open the stack you want to profile

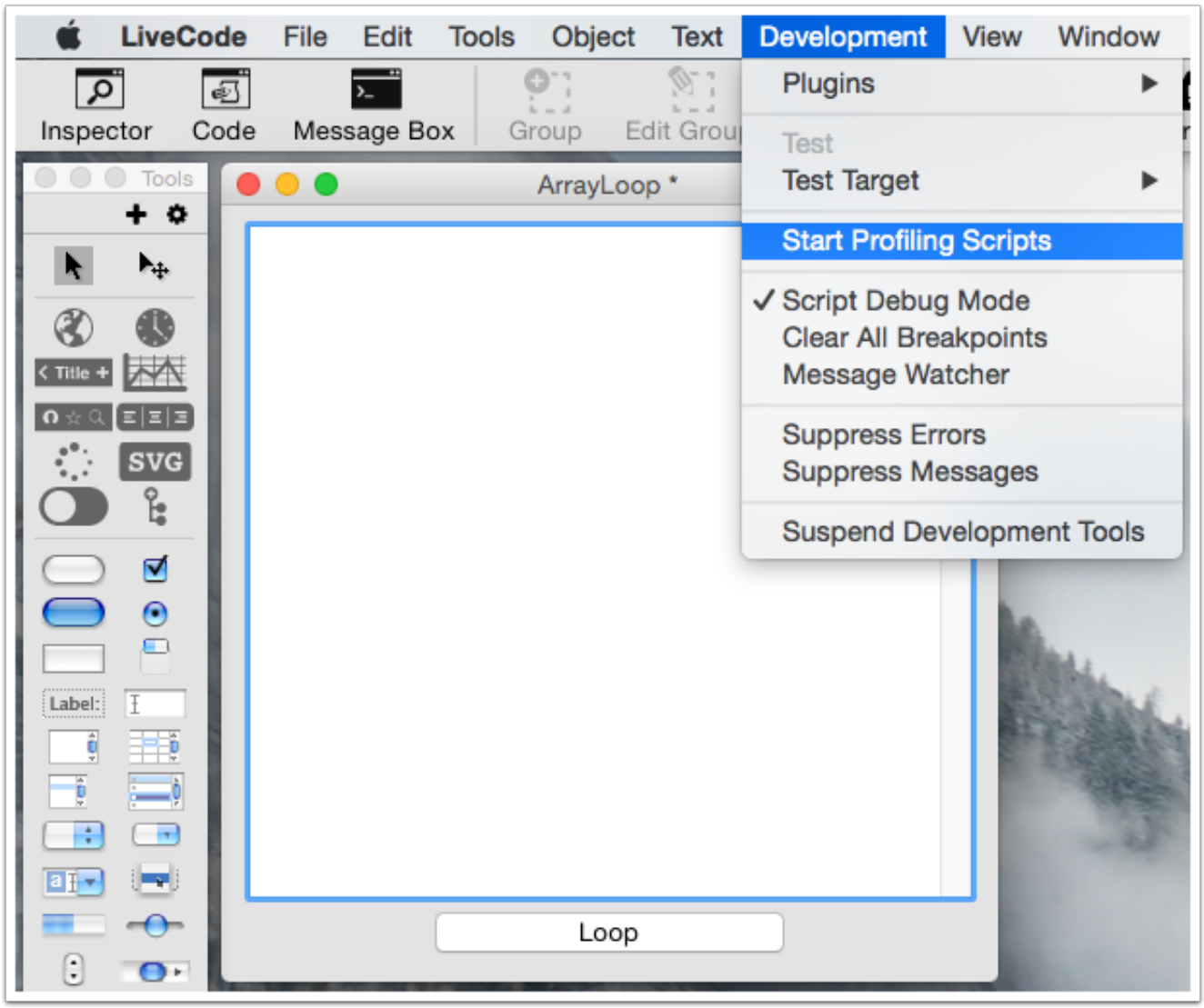
Open the stack you want to profile, and ensure it is the topStack.

The initial looping code is

```
command arrayLoop
   local tList

   put empty into field "list"
   repeat with x = 1 to the number of lines in the keys of sTestArray
      put line x of the keys of sTestArray into tKey
      put sTestArray[tKey] & return after tList
   end repeat

   put tList into field "list"
end arrayLoop
```

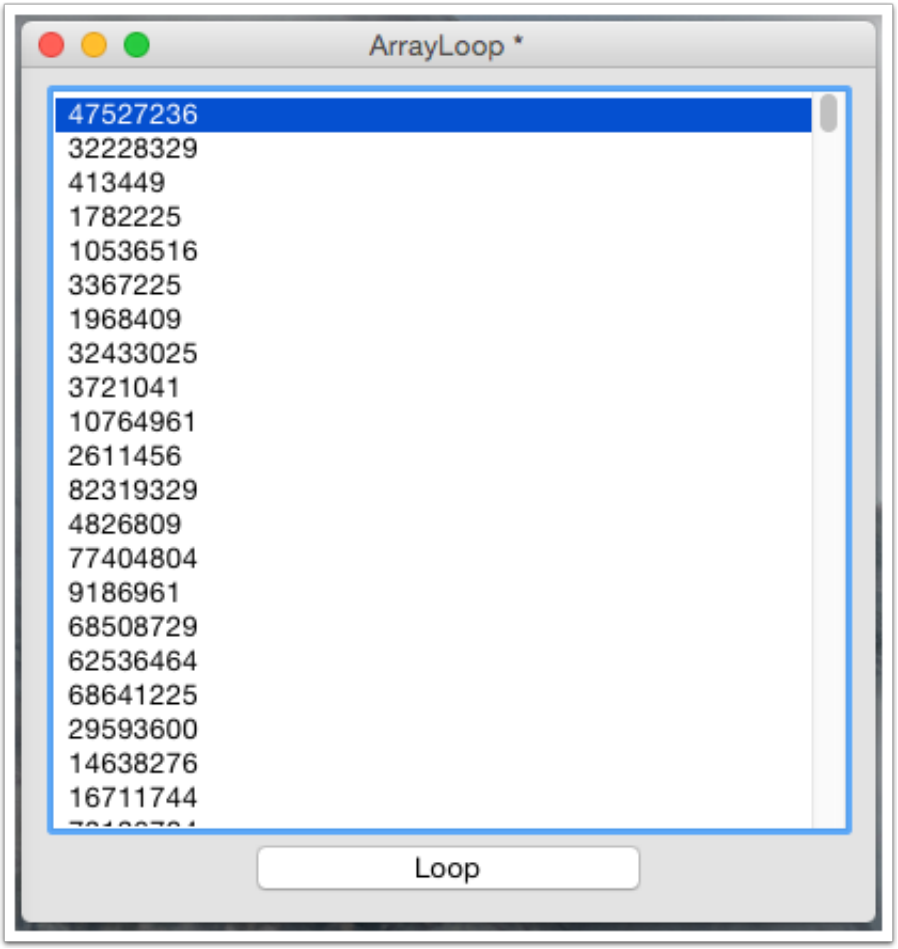
Start the Script Profiler

Choose 'Start Profiling Scripts' from the Development menu, this will begin profiling the t
(https://livecode.com/resources/api/#livecode_script/topstack).

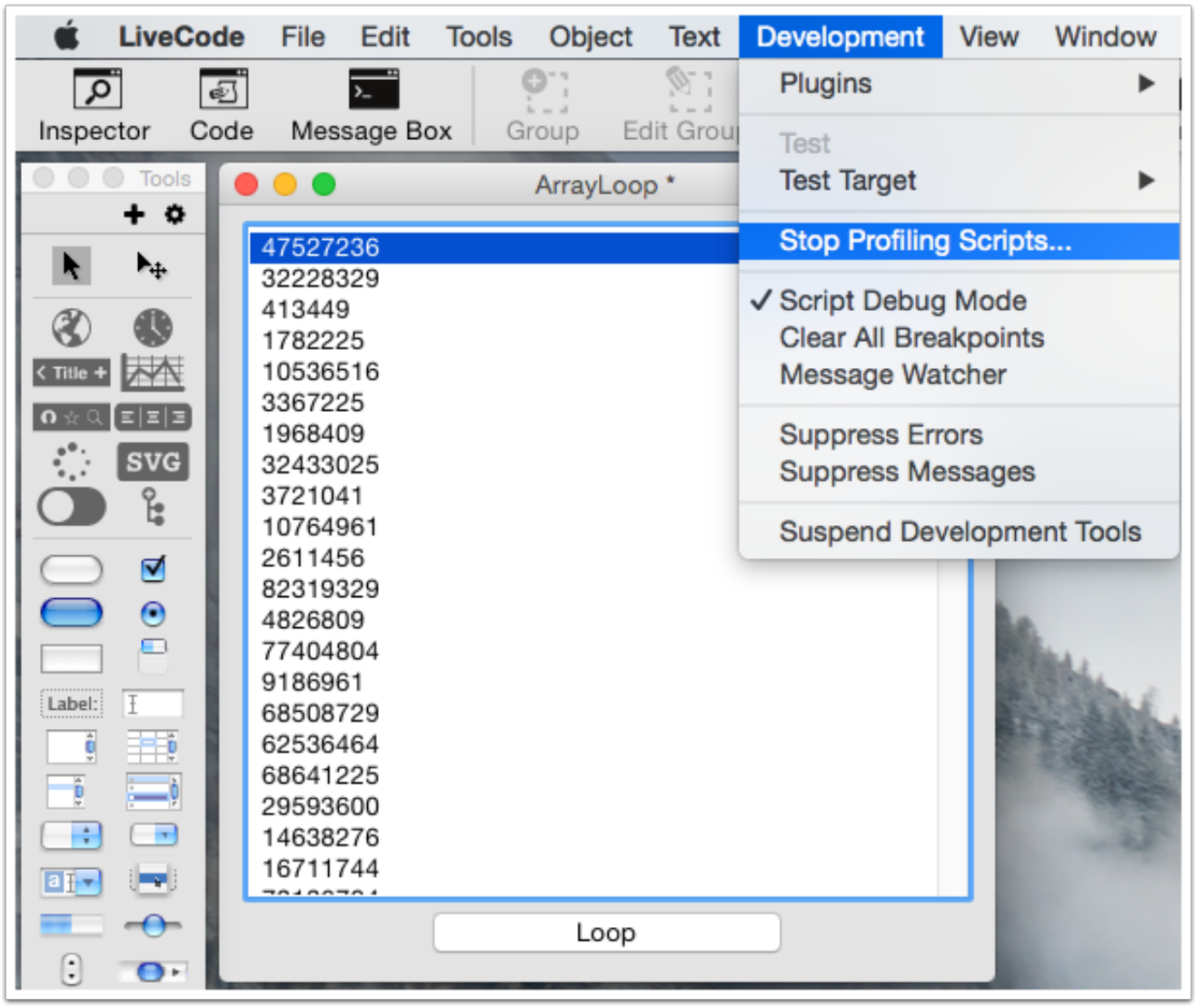


Use the Stack

Starting using the stack to begin the profile, when each line of a script in the stack is exe
time will be recorded.



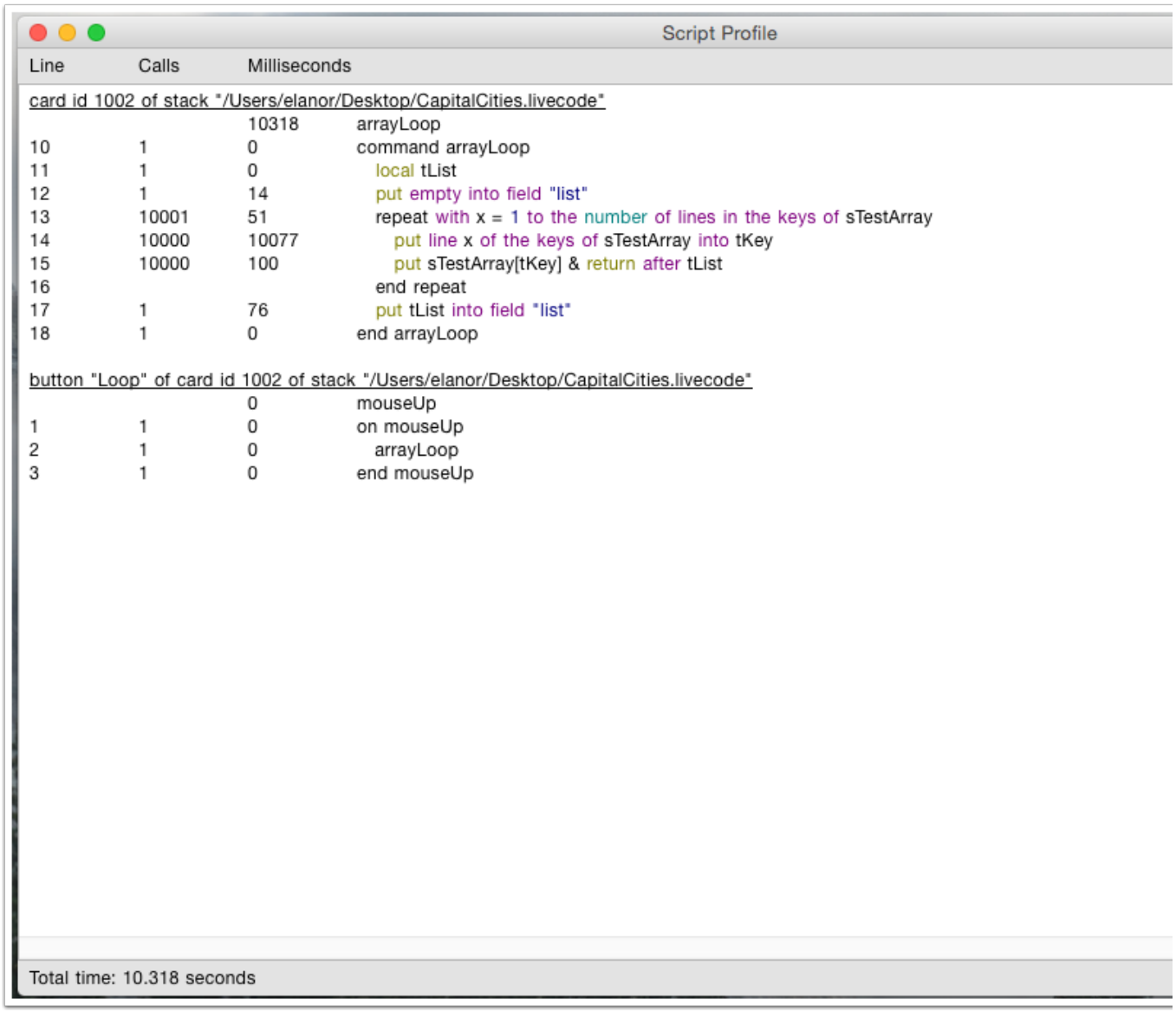
Stop the Script Profiler



To stop the Script Profiler select ‘Stop Profiling Scripts’ from the Development menu.

The Script Profile Report

When the Script Profiler is stopped a report is shown.



The report shows each line of script that was executed

- the line number
- number of times the line was called
- the time to execute the line

The total time in seconds is displayed at the bottom of the report.

Double clicking a line in the report will open the line of script in the Script Editor.

Optimisation

Now we can see where all the time is going we know where we want to optimise our cod

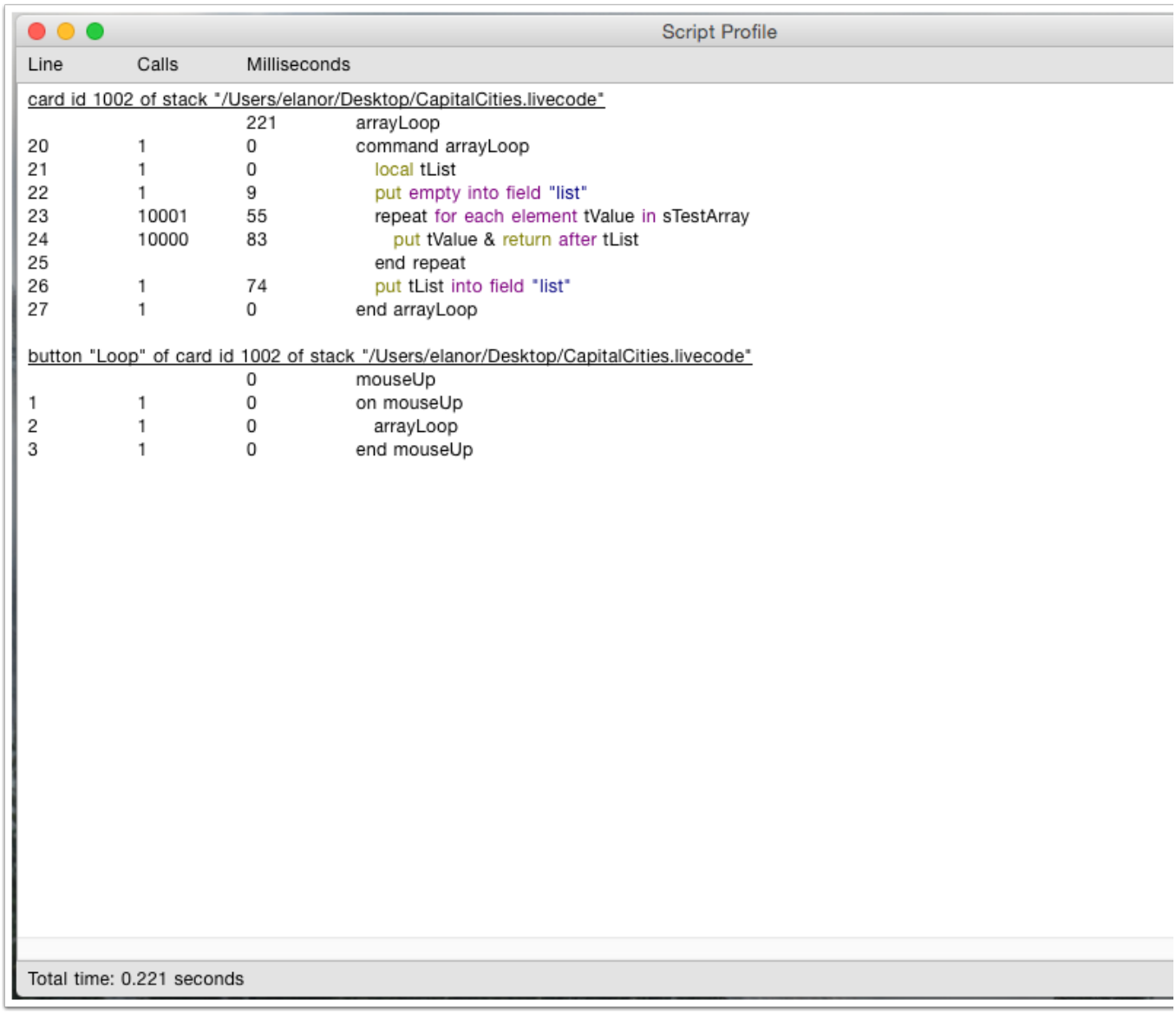
In this example the vast majority of the time is spent on the line

```
put line x of the keys of sTestArray into tKey
```

With a little thought we can see that we can change the form of the repeat loop we are us
more efficient. We are only really interested in the values stored in the array, not the keys,
the code to loop across the **elements** of the array.

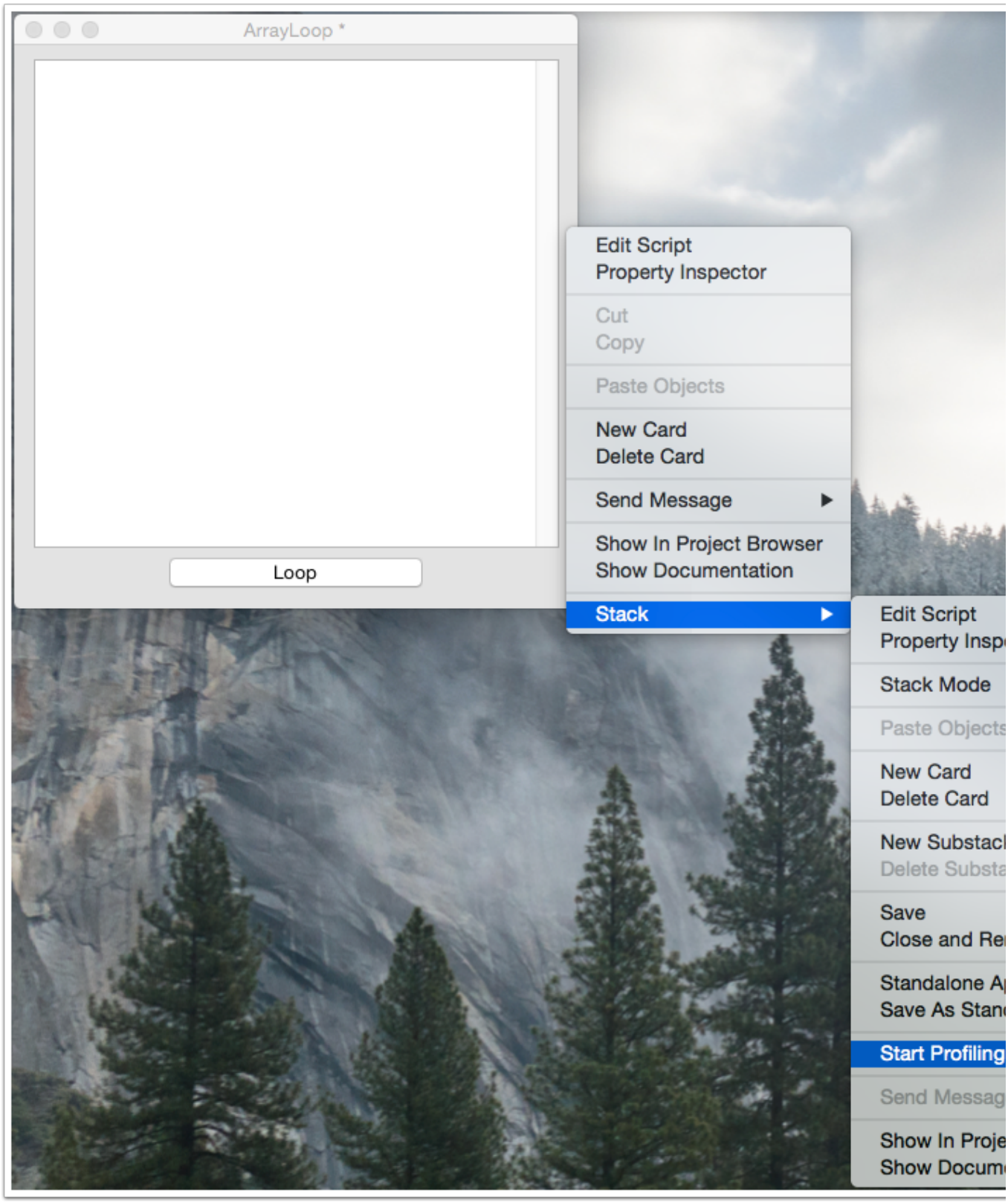
```
command arrayLoop
   local tList

   put empty into field "list"
   repeat for each element tValue in sTestArray
      put tValue & return after tList
   end repeat

   put tList into field "list"
end arrayLoop
```

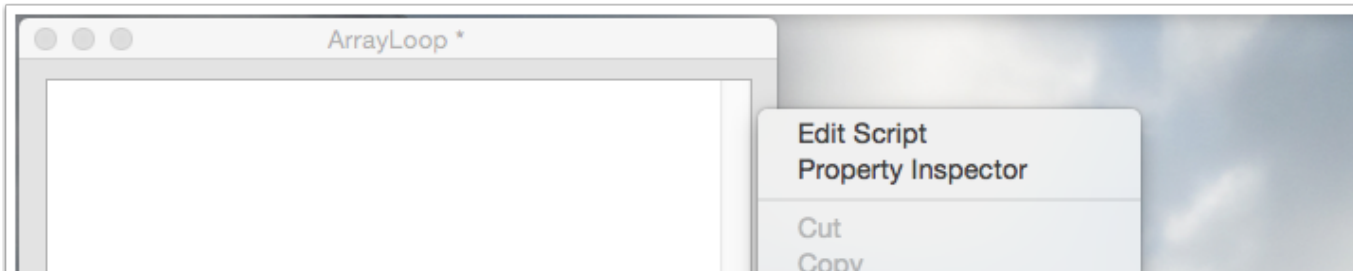
Now run the Script Profiler again and see how much faster the code is, 0.221 seconds ra
seconds.

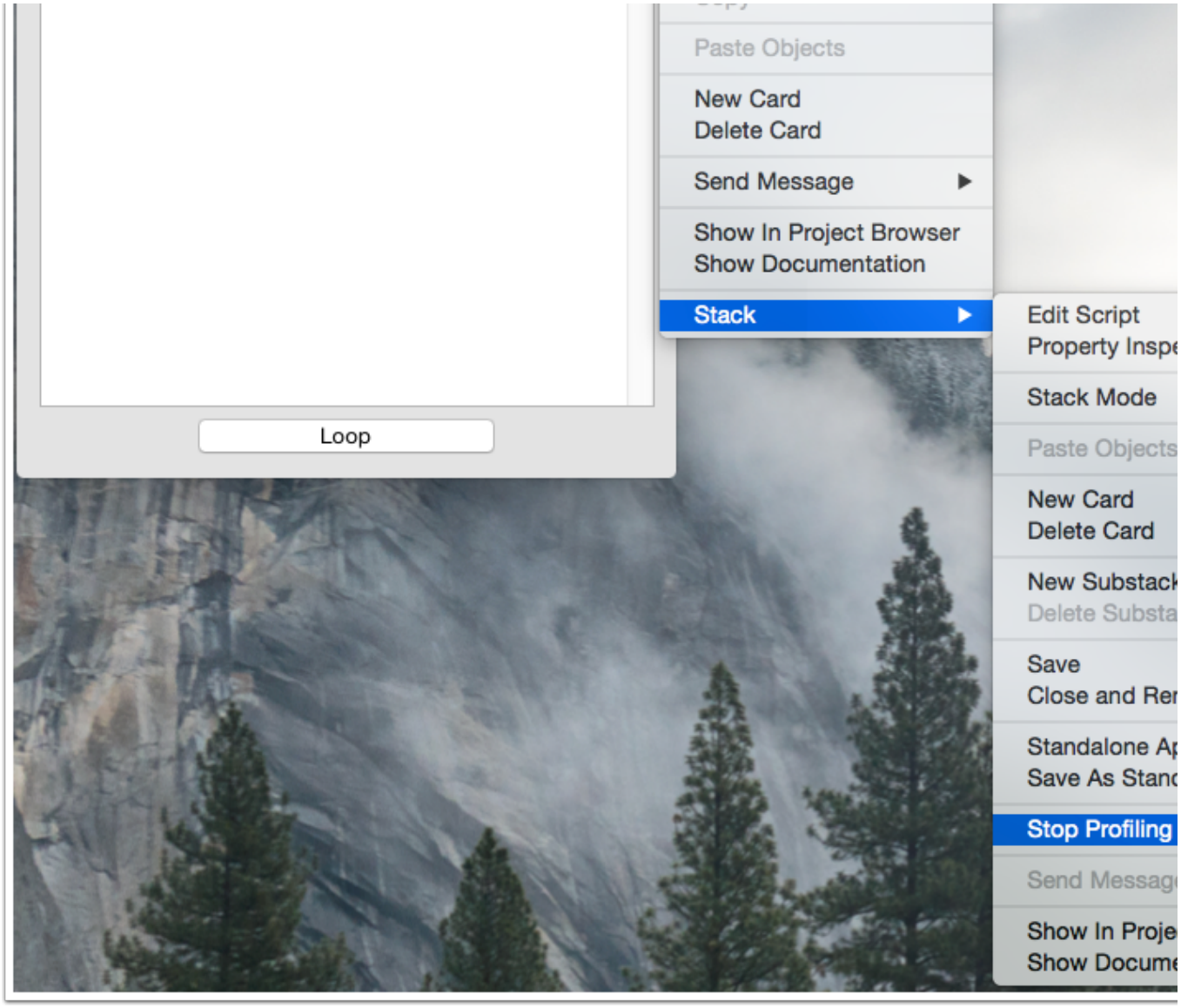


Alternative Ways to Start and Stop the Script Profiler

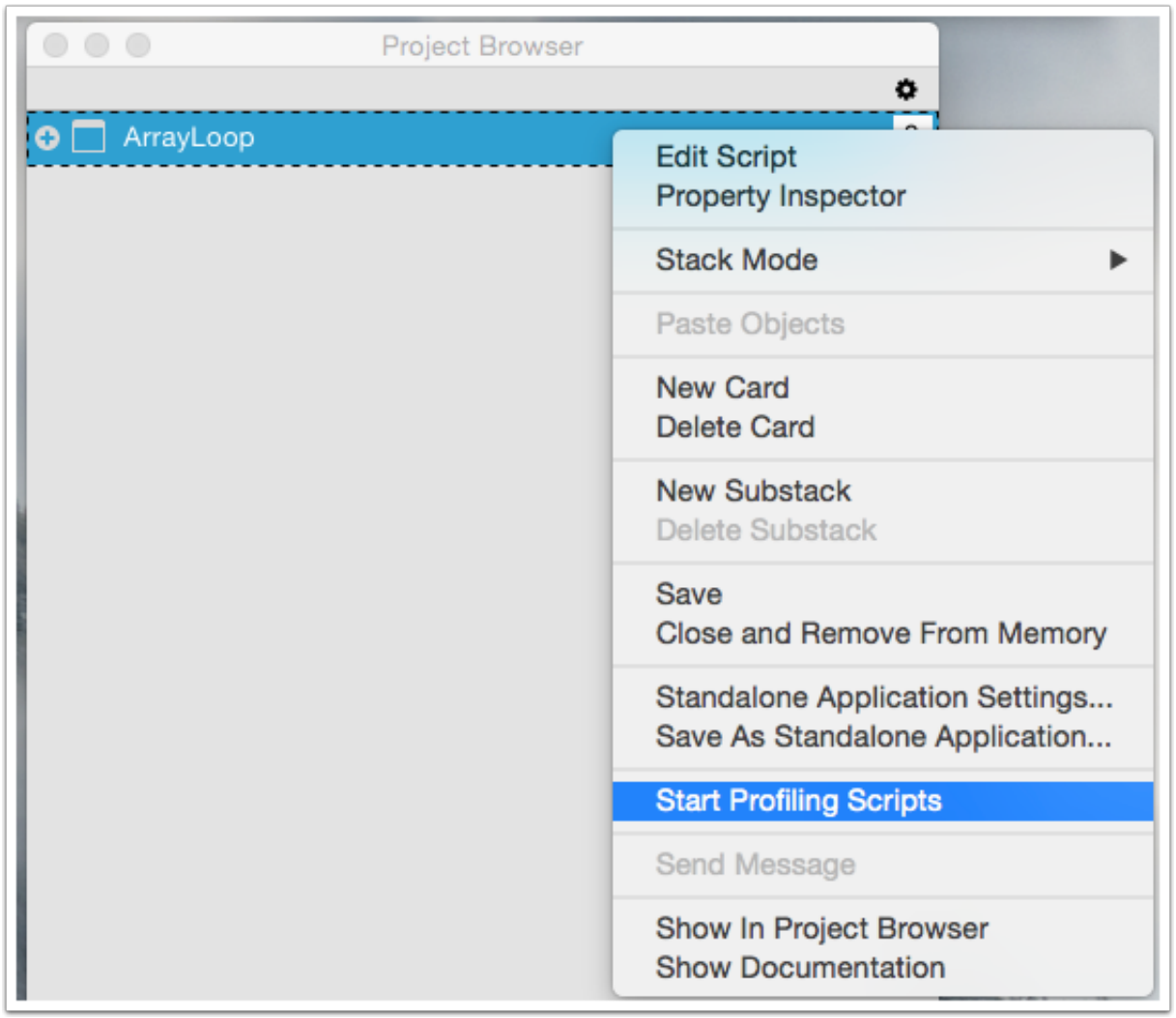
You can also start and stop the Script Profiler using the stack context menu, opened by c
the stack, or via the stack context menu in the Project Browser.

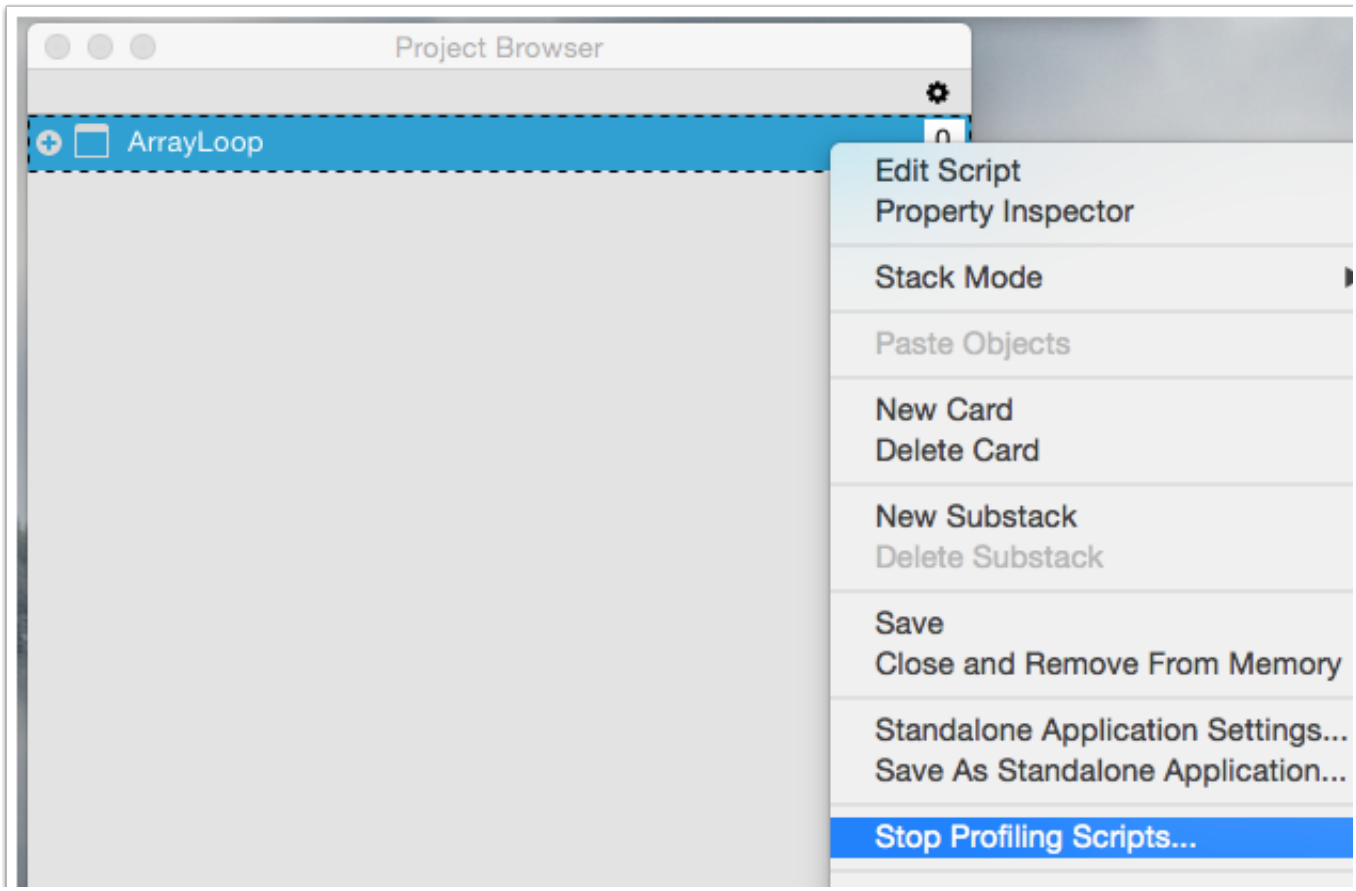

ArrayLoop *
Loop
Edit Script
Property Inspector
Cut
Copy
Paste Objects
New Card
Delete Card
Send Message
Show In Project Browser
Show Documentation
Stack
Edit Script
Stack Mode
Paste Objects
New Card
Delete Card
Save
Start Profiling


ArrayLoop *
Edit Script
Property Inspector
Cut



Using the stack context menu


Project Browser
ArrayLoop
Edit Script
Property Inspector
Stack Mode
Paste Objects
New Card
Delete Card
New Substack
Delete Substack
Save
Close and Remove From Memory
Standalone Application Settings...
Save As Standalone Application...
Start Profiling Scripts
Send Message
Show In Project Browser
Show Documentation


Project Browser
ArrayLoop
Edit Script
Property Inspector
Stack Mode
Paste Objects
New Card
Delete Card
New Substack
Delete Substack
Save
Close and Remove From Memory
Standalone Application Settings...
Save As Standalone Application...
Stop Profiling Scripts...

Using the Project Browser

Available in LiveCode Business Edition 9.0.0 DP-5 onwards.

Business (https://livecode.com/tag/business/) ,

LiveCode 9 (https://livecode.com/tag/livecode-9/) , *Script Profiler (https://livecode.com/tag/script-profiler/)*

4 comments (https://livecode.com/the-livecode-script-profiler/#comments)

Previous (https://livecode.com/debug-your-standalones-with-the-remote-debugger/)

Next (https://livecode.com/german-liveco

Related Posts

Take a look at these posts

Android audio recording library using Java FFI

(https://livecode.com/android-

Debug your standalones with the remote debugger

(https://livecode.com/debug-

LiveCode V "Spinner" A

(https://liveco

audio-recording-library-using-java-ffi/)	your-standalones-with-the-remote-debugger/)	widgets-the-s indicator/)

Profile

Sign in with Twitter (https://livecode.com/index.php?social_controller=auth&social_action=authorize&key=twitter&post_id=20813) Sign in with Facebook (https://livecode.com/index.php?social_controller=auth&social_action=authorize&key=facebook&p or

Comment

Name

Email Not published

Website

Post It

- 4 Replies
- 4 Comments
- 0 Tweets
- 0 Facebook
- 0 Pingbacks

Last reply was March 6, 2017

1. *James Hale*

   View February 24, 2017 (https://livecode.com/the-livecode-script-profiler/#comment-35046)
   So does this mean that “script profiling” is in effect a stack property’?
   By this I mean that in a multi stack project I can, for instance, turn on script profiling for one o and then only get the profiler kicking in whenever a call is made to a handle/function in the ‘ac stack.

   Reply (/the-livecode-script-profiler/?replytocom=35046#respond)

   -  *Elanor Buchanan*replied:

     View March 3, 2017 (https://livecode.com/the-livecode-script-profiler/#comment-35650
     Hi James

Yes, the Script Profiler works by setting the traceStack. So if you start it on your library s
report the calls made to that library stack.

I hope that helps.

Elanor

Reply (/the-livecode-script-profiler/?replytocom=35650#respond)

2. *lorent (http://www.baloune.com)*

View March 6, 2017 (https://livecode.com/the-livecode-script-profiler/#comment-35901)
Hello!
I'm very bad in this stuff but your post help me to understand something!
Thanks!

Reply (/the-livecode-script-profiler/?replytocom=35901#respond)

 *Elanor Buchanan*replied:

View March 6, 2017 (https://livecode.com/the-livecode-script-profiler/#comment-35903
Hi Lorent, I'm glad the article helped.

Elanor

Reply (/the-livecode-script-profiler/?replytocom=35903#respond)

startup/)

LiveCode Conference 2019 (https://livecode.com/california19)

Looking for the LiveCode open source project?

Open Source (http://livecode.org)

Looking for LiveCode FileMaker?

LiveCode for FM (https://filemaker.livecode.com)

Contribute to LiveCode (https://github.com/livecode/livecode/blob/develop/CONTRIBUTI

Contribute to Docs (https://github.com/livecode/livecode/blob/develop/docs/contribu

Release Process (https://livecode.com/resources/release-process/)

User Groups (https://livecode.com/resources/user-groups/)

Support (https://livecode.com/resources/support/)

Directory (https://livecode.com/directory/)

LiveCode Services (https://livecode.com/services/)

Sitemap (/sitemap)

Terms (/terms) Privacy Policy (/privacy-policy) EULA (/eula)