Suggestions for an embeddable scripting language for iPhone and Java app?

Asked 15 years, 11 months ago   Modified 14 years, 11 months ago

Viewed 4k times

▲

5

▼

🔖

🕓

I'm developing an app for iPhone with a Java desktop companion that it synchs to over the network. I'd like to embed a scripting language into both apps so that end users can write scripts that add new behaviours and interact with the object model, and so that I can more rapidly develop some features.

Any suggestions for this, or anyone done this before and got experiences to share? I'm thinking of something like tcl which I've used in the past, but not tcl itself as I'm not a big fan of the language.

Main criteria are that it should be

- lightweight as poss (memory usage)

- easy to interface and extend with java and objective-c

- (nice to have) readable/approachable for people with limited programming experience

My initial hunch is that I should be using one of python, ruby, or lua. My preference would be ruby, as I already have some experience with it and don't know much about the others. However my main unknown is which of these is easiest to get integrated with iPhone and Java.

edit 2: per Jason Coco in the comments, the SDK terms prohibit embedded scripting languages. Checking into this it does indeed seem to, but I read it to preclude dynamic installation and extension only. I would still be interested in answers here, as the agreement doesn't seem to preclude having prepackaged scripts inside the application bundle itself - Apple would still get to vet that code.

The agreement also seems to allow use of Apples "interpreters"...what are these? Javascript and what else? Any route to use those here?

`java` `cocoa-touch`

Share

Improve this question

Follow

I personally would use Lua, but unfortunately the iPhone terms of service prohibit embedded scripting languages. Your app won't make it to the store. – Jason Coco Jan 8, 2009 at 12:50

Ah I see, I was wondering about that - I thought that was only if providing a full blown interpreter that could be used to run apps in their own right? In this case the scripts would only extend the app itself. – frankodwyer Jan 8, 2009 at 12:56

I think the SDK prohibition is on scripts downloaded from the Internet. If the only way to create them was on the phone itself I believe it would be okay. Still, I'd read the agreement if it's a critical component of your application!
– Stephen Darlington Jan 8, 2009 at 13:20

Looking into this it currently seems to preclude interpreted code in any form to be installed by the app itself, no matter how the app does it. I am going to reword the question accordingly as it would still be interesting to include scripts with the app bundle. – frankodwyer Jan 8, 2009 at 13:23

3 Answers

Sorted by: Highest score (default) ⇕

JavaScript

2

As I understand the iPhone SDK Licence, there is nothing preventing you from using a scripting language in your app - just that you cannot intall any interpreters or runtimes of your own. You can only use those scripting languages for which Apple provides the interpreter in the SDK.

Given that you want to run the same scripts in your iPhone app and in a Java app, the obvious choice is JavaScript. You can use Apple's APIs in your iPhone app, and something like Rhino (http://www.mozilla.org/rhino/) in your Java App.

You have to be aware that there may be slight differences between the two interpreters in the more obscure regions of syntax or object model.

P.S. I assume that users are going to be writing their own scripts to run on their own device/desktop. These would be part of the application data, and should be fine under Apple's licence

Share Improve this answer

Follow

edited Jan 8, 2009 at 23:28

answered Jan 8, 2009 at 14:23

Vihung
13.4k ● 18 ● 67 ● 95

Is their javascript extensible such that a script can invoke functions provided in objective-C in the app? – frankodwyer Jan 8, 2009 at 14:28

I am not sure, but I imagine that in both cases - the Apple runtime and the Java runtime (e.g. Rhino) - you would provide the runtime with an object model that would be manipulated by the script – Vihung Jan 8, 2009 at 14:55

3   The Apple Javascript API (JavascriptCore) is in the PrivateFrameworks folder, and thus cannot be used directly from the best of my knowledge. – Ecton Mar 6, 2009 at 4:25


1

The only thing I've seen that allows a non-objective-c/c/c++ application to run on the iPhone is Unity3d (http://unity3d.com/unity/features/iphone-publishing) - BUT, it uses C# via Mono and does a full static compilation (http://tirania.org/blog/archive/2008/Nov-05.html) down to native code in order to do it. So, by the time the app is on the phone, it's no longer C# so it's allowed by Apple (and several apps have already made it into the App Store - so this does seem to be acceptable).

I don't think you would be able to do the idea you're talking about even if you could do the equivalent for Java/Ruby/Lua/Python/other (so having your desktop app pre-compile and upload just the native code to the device). As far as I know, you can't execute code outside of your application bundle...and if you modify the application bundle, then you invalidate the codesigning Apple does which allows the app to run on your phone in the first place. So even if you could get executable code to the phone, I'm not sure that you would be able to run it if it didn't come along with the app in the first place.

Share   Improve this answer

Follow

answered Jan 8, 2009 at 13:44
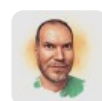
Adam Byram
1,422 ● 10 ● 8

my current thinking is to have static prepackaged scripts inside the app bundle. this means that users can't load their own scripts, which sucks, but it still allows me to script some things. also if a user contributes script for the desktop version, I could package it in the next iphone release.
– frankodwyer Jan 8, 2009 at 13:54

+1 for the C# info by the way, that's interesting - didn't know that was possible. – frankodwyer Jan 8, 2009 at 13:59

I assume the scripts need to be executed at runtime - within the application as it runs on the iPhone or the Java VM on the desktop. The C# solution mentioned above is a compile-time option - allowing you to use C# instead of Objective-C for developing the application. -1 for irrelevance – Vihung Jan 21, 2009 at 16:59

1

I was researching this too, and it *seems* that it's possible to pre-compile a Lua script (by converting it to C using Lua and then compiling the C file). Because all of your code could then be part of the application bundle (including the embedded Lua interpreter), it *should* be acceptable as an iPhone app.

See here for a discussion and sample script: http://lua-users.org/lists/lua-l/2008-11/msg00453.html

*Note that I haven't tried this (yet)

Share Improve this answer

Follow

answered Mar 6, 2009 at 4:23

JoeS
133 ● 2 ● 7