

# What are your feelings on JavaFX?

Asked 15 years, 10 months ago   Modified 8 months ago

Viewed 4k times



9



I currently do a lot of work in ActionScript 3.0, I also love to program in Java. Is JavaFX perfect for me? What is the general feeling on JavaFX, will it become a power house, or go down the same path as Java Applets? Could the designers I work with become comfortable with JavaFX to the same extent they are comfortable with ActionScript and JavaScript?

javafx

ria

Share

Improve this question

Follow

edited Feb 17, 2009 at 20:53



John Topley

115k ● 47 ● 199 ● 240

asked Feb 10, 2009 at 19:52



ForYourOwnGood

40.3k ● 5 ● 31 ● 27

11 Answers

Sorted by:

Highest score (default)



Just wanted to add my \$.02... I've been working in JavaFX for the last 4 days on my first little side-project

12



using it. As some background, I've been programming professionally for about 9 years, starting with C, and have been doing Java and C#/.NET for the last 6 yrs.



IMO, JavaFX its way more frustrating that it should be.



Here are some gripes:

1. The syntax is just odd at times. It could easily be more like Java, since its *JavaFX*. But the syntax isn't an easy transition from Java.
2. The *order* of items in a .fx file actually matters, which means you run into stupid circular reference errors, and "oh you can't use this variable yet because it hasn't been initialized" problems that the compiler should handle with ease, but doesn't.
3. Random things just don't work. Actions/events on Swing controls don't always work, for example *SwingSliderBar*'s *onKeyPressed/released* don't seem to be called.
4. Error handling is just *bad*. If an exception occurs that isn't handled, there is no real way to tell other than the Java console, and UI elements start to react funny. For example, make a *SwingText* box and bind its value to a variable. Now trying to edit the value in the text box will throw an exception because you cant edit the bound variable. However in the UI, the text box just starts having funny things happen. some characters only 1/2 paint, sometimes backspace does nothing, sometimes it deletes a character, sometimes you can press 2 keys ont eh keyboard

like "1" and "2" and the text box will end up having "21" entered in it instead of "12", etc...

Although my absolute #1 problem with JavaFX development right now is **Netbeans**. It is pathetically bad at JFX. Can't debug, errors display wrong in the IDE (I've had it flag comments as errors!), the intellisense only works like 40% of the time, even the code templates preprogrammed in the IDE for drag & dropping controls aren't correct. I forget which one, but one of them drops a "&" at the end of the inserted code that is never valid and always has to be manually deleted... its just plain *awful*, and is unacceptable for a company like Sun.

Another gripe is general documentation. Its just lacking. Somehow the JavaFX API doesn't even come up as the #1 search result on google when searching for methods/classes. Tons of "examples" out on the web don't work any more as every version has major refactoring changes, and classes removed or renamed.

Overall, I give JFX a 4 out of 10. I *want* to like it, but JFX 1.1 just doesn't cut it... its definitely not what I would consider "production ready".

Share Improve this answer

Follow

edited Feb 18, 2010 at 21:14



David Koelle

20.8k ● 23 ● 94 ● 130

answered Apr 5, 2009 at 0:32



CodingWithSpike

43.7k ● 18 ● 105 ● 139

---

+1 The syntax is really weird. While at first glance there seem to be a lot of similarities between Java and JavaFX, there are a lot of subtle differences. – [helpermethod](#) Mar 7, 2010 at 18:01

---

While the syntax between Java and JavaFX are different...the interoperability is really good. Calling Java from JavaFX is fantastic...and since JavaFX is a rich client side technology...there is not really a need for Java to call JavaFX. And in my humble opinion...JavaFX is superior to legacy Java SWING for client side GUI development. – [Eric Warriner](#) Aug 31, 2010 at 18:37

---



A resounding "meh".

3



When I looked at it a year ago, they had a one-way SVG to JavaFx conversion tool. Great, so you can author your visual content once, mark it up with a lot of behaviour, and then the next time you want to make it look good, then what?



If you take a look at [this tutorial](#) you can see what I mean. We're drawing stuff by dragging shapes from a palette into source code. OMGWTF. I am not showing that to my graphics department.

I hope I'm wrong about JavaFx, but I don't think they get it. Please, won't somebody at Sun give us a presentation layer that doesn't have its tentacles inextricably intertwined with code?

Follow

answered Feb 10, 2009 at 21:20



[joeforker](#)

41.7k ● 41 ● 158 ● 255

---

1 I have to agree with your last line. Although, frankly, lately, I think that's the case for Java in general. – [Randolpho](#) Feb 11, 2009 at 1:43

---

1 You should take a look at the Production Suite. It has great tools for Photoshop and Illustrator and the FXD Loader API lets you de-couple the UI from the logic pretty nicely. – [Honza](#) Feb 17, 2009 at 21:45

---

The JavaFx production suite looks interesting. Can I do similar when using Swing components in JavaFx? – [joeforker](#) Feb 17, 2009 at 22:09

---



3



I left my last job to move from Java to .NET development.

There were a number of reasons for making the move, but the single biggest reason was that I was sick and tired of trying to build 1st class UI software with Java & Swing. It has been six years and I'm so glad I moved on. I see no reason to believe that Sun finally understands UI development with JavaFX.

I am convinced that Microsoft is finally in the process of giving us a platform to build rich interactive applications in the browser. I say that after having built commercially available software which was delivered as a Netscape Plugin 13 years ago, followed by ActiveX controls and

Java Applets, and seeing all of these platforms fail to become ubiquitous in the enterprise for one reason or another.

I realize that Silverlight 2 is still lacking in depth and maturity, but Microsoft has shown me enough commitment at this point that I believe it will be the dominant RIA platform in a few short years - at least for projects which require a "real" programming language. I am sure Flash et al. is not going away anytime soon, but Flash is not appropriate for the kinds of software my company builds.

The icing on the cake for me is the fact that I will still be able to use Visual Studio, C# and a large percentage of my current code base (the core engine which is entirely separate from the UI). Of course, if you are coming from ActionScript, this would not help you.

One more important point is the fact that Silverlight and WPF share so much in common. Our plan is to share a large amount of implementation between Silverlight and WPF versions of our software. It is only a matter of time before WPF is the standard for Windows applications – I don't know whether that is a couple of years or ten years, but it will clearly happen over time. Being able to target the most popular browsers / OS's with Silverlight and Windows from the same code base is a tremendous advantage IMO.

If you know Java then moving to C# is a piece of cake. And unless you are using one of the nice (not free) Java

IDEs, then even the free versions of Visual Studio will be an improvement over what you are used to. The hurdle will be learning the new way of doing things with XAML – but it's some pretty cool stuff so you might actually enjoy it.

Share Improve this answer

answered Feb 17, 2009 at 21:13

Follow



[Joe Erickson](#)

7,217 ● 2 ● 32 ● 31



2



Although it appears fairly powerful in terms of capabilities, I'm kinda blah about JavaFX because of its structure and implementation. It seems like a really half-hearted attempt at getting into the Flash/Silverlight market. Too scripty.



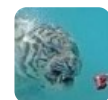
I would argue in favor of going the Silverlight 2 route, but I'm primarily a C# developer so I'm a little biased there. If you don't like that route for whatever reason but still want a richer UX for your users, I'd suggest Flex; it seems much better organized than JavaFX to me.

Just my two cents on the subject.

Share Improve this answer

answered Feb 10, 2009 at 20:10

Follow



[Randolpho](#)

56.4k ● 18 ● 151 ● 180



If you know Java but want to do the stuff you thought was only feasible in Flash - then yes JavaFX would be good

2 for you.



Without a doubt it's going to be much more easier to merge your Java knowledge with the design stuff.



And I believe the tooling will only get better which will make it simpler to use.

Share Improve this answer

answered Feb 10, 2009 at 21:11

Follow



Mark Wilcox

191 ● 1 ● 2



2



Unless you're working on an internal app I would stay away from it. Users generally don't want to have to deal with another program that accomplishes the same thing as Flash. I don't think its install base is large enough yet to make it convenient for end users.



Share Improve this answer

answered Feb 10, 2009 at 21:35

Follow



Jared

39.8k ● 29 ● 113 ● 146

---

Maybe they should figure out how to compile it to Flash bytecode. – [joeforker](#) Feb 10, 2009 at 21:37

---



2

I've been developing Flash Applications with Flex for about 2 years now and I decided to give a try to JavaFX because we are constantly getting user complains that they cant use the applications from their iPhones (and I love Java).





That's one strike for Flash (no wide mobile support).



To be honest I was quite impressed with JavaFX (in a very bad way).



- The documentation is incomplete.
- The script is simply awful; its this weird hybrid between JSON and R with a feeling of a Java-deja-vu.
- I spent the first 3 days painting polygons and making gradients with CODE... WTF!!
- I tried to convince my graphic department to try it and they simply don't seem to grasp how the production suite is supposed to work, they keep complaining that Flex skinning is way easy and looks better in the end (Which is absolutely true).
- The "CSS support" is simply a bad joke.
- It generally feels like a mediocre attempt to offer an option for RIA frontend.

I can only think of a couple of good things about it:

- It can be run from an iPhone / iPad and almost all mobile devices.
- You have access to all the Java code you want which is great considering the limitations that ActionScript has (no overloading, no private constructors, etc). This is a great thing for us the programmers, but lets remember for a second that this is a

frontend/presentation technology... that means that users will have to actually see the thing, so if it doesn't look good and have cool animations / effects they won't dig it.

- The Script is way less verbose than MXML files are (with the cost of being unintelligible).
- Talking about performance... Flash Player is this huge green blob that keeps growing and growing until no RAM is left compared to how JavaFX runs (JVM rocks! unfortunately this has nothing to do with the actual JavaFX API it's just that the JVM... well it rocks!).
- It has this cool feature where you can drag the applet outside the Web Browser.

In the end, I'm happy I have an option to go mobile but this is light years away of the matureness that Flex/Flash has accomplished regarding RIA applications. The future of Flex/Flash as a wide distributed web technology is not clear (it may end up being used only for annoying banners and online games), no one wants to depend on a closed technology as the Flash Player is, that's why the whole Web community is striving to get Flash out of the picture (HTML5 video support, No Flash Player for Apple devices, etc). So an attempt to have an open one is always welcomed, it's just that JavaFX feels like this incomplete rushed beta version of something that Sun felt obligated to come up with in a weekend during a bad hangover.

I Hope this is useful to someone (and offensive to someone at Sun/Oracle =p ).

Share Improve this answer

edited Apr 9, 2010 at 18:22

Follow

answered Apr 9, 2010 at 18:15



[Chepech](#)

5,531 ● 4 ● 49 ● 72



0

I've spent the weekend 'playing with it. I see nothing useful in it. It's a iteration of swing / awt. I guess it will be nice for mobile devices but beyond that its nothing useful.



Ideally I'd like to use flash but find it painful to intergrate with a backend of any type.



Share Improve this answer

answered Mar 29, 2009 at 17:34

Follow



[Steve](#)

21.5k ● 24 ● 71 ● 92

- 1 It's not only for GUI programming, it's a full blown programming / scripting language. I think it addresses many ugly things in Swing (action listener boilerplate code e.g.) very nicely. – [helpermethod](#) Jan 22, 2010 at 18:08



0

Well, the syntax of both ActionScript and JavaFX seem to share a lot of similarities, so maybe "Yes".



I'm learning JavaFX script at the moment and I actually like it. But what I don't like, and is maybe it's biggest drawback, is it's awful documentation, which is often not up to date or incomplete.

Share Improve this answer

answered Feb 20, 2010 at 9:37

Follow



[helpermethod](#)

62k ● 71 ● 196 ● 280



0



I've been working on a JavaFX application for several months now. Personally, I love the language. They seemed to me to have made some very smart decisions in choosing the syntax and language constructs (I can bore you with a list if you'd like). I've been programming in it for a few months now and it seems like a very efficient and even enjoyable language to program in.

I think its best use right now is for desktop applications and/or applications deployed through webstart. On the desktop it has a rich set of features and can still make use of the other features of Swing and the rest of Java. From what I hear applets are still slow on some systems, and without Android support the mobile capabilities are non-features. The applet/mobile/TV/web support seems to me more like a bonus for desktop developers then as key features that would get you to use the technology.

So it really all depends on what you plan to use it for. If you are building desktop applications that you want to run on the Java VM that can make use of easy multimedia and rich ui controls, then I think there are good reasons to

look at the language. WebStart has improved quite a bit and makes for a nice deployment tool. If you are looking to build web applications, then it might be interesting, but for now I'd say HTML5/ajax are more relevant (you might want to look at ZK in this case). However even with HTML5 ajax has its limitations, and if you find yourself running into them then JavaFX may offer you options. For mobile platforms it won't be relevant until there is stable Android support - in that case I'd just stick with the Android platform itself for now.

Share Improve this answer

answered Aug 17, 2010 at 13:19

Follow



Praeus

196 ● 4



0



FXML and it's CSS are just modified versions of what was already made. It's tedious to learn something new. I got it because I thought it used html and non-modified css, but I was wrong.



I know it's hard to translate a bunch of text then write functions for JavaFX to do everything, it's a lot - but they've been working on it for 11 years, and it feels half-assed when I try to do something and apparently the value's been renamed, or it's missing, or whatever.

Share Improve this answer

edited Apr 11 at 11:21

Follow

answered Apr 11 at 9:11

---