

## Summary of JavaFX (Mobile) Strengths and Weaknesses as of March 2009

<b>Killer Strengths</b>	<b>Comments</b>
DSL productivity	Terse code; reflective of intent. Considerably easier to maintain than Java equivalent.
Java integration	Seamless integration enables existing libraries to be leveraged very efficiently.
Bind (the killer keyword)	Results in extremely succinct code (in comparison to Java.)
Key-frame animation	Used wisely, animation can significantly increase the quality of the end-user experience. Can also be used for programming (simple) games. In combination with "bind", key-frame animations can now be constructed easily and intuitively.
Java2D effects	Effects can improve the look of an application substantially and are now very quick and easy to use.
Multimedia	Multimedia is finally (!) easy to integrate in an application.
Designer / developer workflow	Graphical designers and software developers can work efficiently together using JavaFX Plugins for Photoshop / Illustrator. However, more experience is required to seriously test the viability and scalability of this approach.
Light-weight access to RESTful web services	Web-service invocation now requires very little code. Call is automatically handled asynchronously although the developer must manage asynchronous handling of time-consuming processing.
Light-weight XML parsing	Especially powerful in conjunction with web-services.
Reuse potential across desktop/mobile	The rule here is: Develop for mobile from the outset! Then, in a new project, make whatever enhancements are necessary for the desktop version.

<b>Potential Killer Strengths</b>	<b>Comments</b>
3D	Promised for later this year, this feature will almost certainly wrap the Java3D API. But how will it perform under JavaME?
TV	Promised for later this year, this feature will presumably build on JSR927

<b>Killer Weaknesses</b>	<b>Comments</b>
Widgets	JFX text field of limited use. Can use Swing widgets, but no grid (or similar) layout-managers means extra work for the developer. Fortunately, widget sets are coming. The question is: What, exactly, and when?
Codecs	The current codecs from On2 don't support H.264 (a killer requirement for MusicPinboard 2.0, for example.) Depending on your environment, you may be able to install codecs separately, but then your app is no longer truly portable.
App Store	Apple once again has set the standard here. The benefits for vendors and consumers are undeniable. Yet in Barcelona there was a slight hint that something like an App Store might be in the pipeline. Will there be an announcement at JavaOne '09???
Sun	Future unclear. Potential buyer question: How will JavaFX generate wealth?

<b>Weaknesses</b>	<b>Comments</b>
Adobe plug-ins don't support skinning	Sun: Envisaged for the future. No date specified.
Multi-threading	This is currently a bit clunky. See comment on use of RESTful web-service above, for example. Could therefore use some additional library features.
Documentation	API documentation is absolutely lousy at the present time. At least one book is in the pipeline (Wheeler), which will hopefully become the first reliable and comprehensive resource.
Multimedia	Player buggy: Reset of video is unreliable.
IDE Support	NetBeans plug-in is currently limited in scope and buggy at the same time. This could prove to be a big turn-off for the community, since it undermines JavaFX's ease-of-use story.
Animator tool	Photoshop / Illustrator plug-ins only enable the definition of static content. All animation and interactivity is left up to the software developer. Again, Sun has promised support for this.

<b>Killer Weaknesses – Mobile</b>	<b>Comments</b>
Lack of devices	Sony Ericsson and LG deal has not yet resulted in a shipping date (JavaOne '09?) The ultimate showstopper! Nokia (and others) are focussing on Flash.
Performance	Application response times are not yet competitive with the likes of iPhone. Round-trip time for user-interaction needs to be halved, at least. Animation frame-rate needs to be at least doubled. These estimates are based on today's performance on a Sony Ericsson Xperia device.
Stability	Lots of stability issues have apparently already been addressed, especially in the multimedia area – although we have not seen any builds with the fixes running yet.
Widgets	The situation is worse here than for the desktop because not even the Swing widgets are available. Presumably the standard widget set (or a subset thereof) promised by Sun will be designed for use on JavaME.
Multimedia	Currently behaves significantly differently from the desktop in a number of respects – and is even buggier. Presumably this is because it is based on JSR135 (Mobile Media API) whereas JavaSE (desktop) multimedia support comes from the Java Media Framework.
Start-up time	Apps need to start in one second or less. Sun engineers promise huge advances in coming months, but this will prove challenging. MusicPinboard start-up time on Sony Ericsson Xperia: 10s ☹
Desktop/mobile discrepancies	Certain behaviours (video, file-system) vary radically between desktop and mobile device.
Emulator/device discrepancies	Certain behaviours (screen dimensions, video overlay, file-system) vary radically between emulator and device.
Profiling: Emulation / On-device	Nothing publicly available.

<b>Weaknesses – Mobile</b>	<b>Comments</b>
Skinning	Skinning not supported on mobile (a footprint issue). Sun is exploring how to solve this.
Multi-threading	Asynchronous support such as RemoteTextDocument is not available on JavaFX Mobile. Consequently the developer needs to mess around with Java threads – okay if you know Java but not for beginners.
OS	IDE support currently only available under Windows. Sun: Mac support Q4 '09