



INFORMATION TECHNOLOGY

WHEN ANDROID CALLS OPEN SOURCE FRONTIERS IN SMARTPHONES

(As published in *The SciTech Lawyer*, Volume 7, Number 2, Fall 2010)

by Kelly L. Frey, Sr.*

iPhone¹ versus the Droid, Nexus One and Droid Incredible.² The iDo's versus the iDon'ts. The "blue states" versus the "red states" (of 3G cellular coverage). Touch screen versus the keyboard. So what's the real difference? To the application development community the difference is all about proprietary versus open source development platforms.

The iPhone Phenomenon

When iPhone launched, application developers jumped on board, creating more than 100,000 new iPhone apps using Apple's proprietary iPhone OS development platform and toolset.³ Ten thousand new iPhone apps are now created each week. Consumers have downloaded more than two billion copies of iPhone applications. With this type of adoption curve, you would think it would be game over in the smartphone application developer's community: Apple wins on the adoption curve, with hockey-stick growth.

But for software developers, it's all about hedging your bets in the multi-billion dollar applications development market for smartphones. Anyone familiar with evolutionary biology and the concept of the "bottleneck," understands the dilemma both Apple and developers now have: because Apple has based development upon a proprietary system and has a centralized review and approval process, each application developed must be submitted to Apple and go through a single queue for approval before it can be released. Plus, new iPhone apps have to contend for survival in the most crowded and competitive application marketplace since the introduction of the PC.

Is Open Source the Way?

Google, Motorola, HTC, and Verizon are relying upon the undeclared war that has waged for decades between proprietary systems developers and the open source community. The Droid, Nexus One, and Droid Incredible are here and it's all about the open source Google Android Operating System that facilitates the applications and growing development possibilities available for these smartphones.⁴

For proponents of proprietary systems, the mantra has long been that there is no money to be made developing open source (only selling or servicing open source systems and platforms). However, this mentality is far from accurate. In fact, there is the potential for a lot of money to be made, especially in the open source environment.

For developers in the open source community, the biggest question and concern is "Which open source license governs the source code

that is developed?" Over the last several years, the courts have become clear that open source software licenses have the same contractual standing as any other type of software license,⁵ including the obligation under some open source licenses requiring that any developer "give back" to the community any creations the developer makes: a concept referred to as reciprocity or "copy left".⁶

Android and the Benefits of Open Source

Determining which open source license applies in the new smartphone environment is critical, because that license will dictate the level of reciprocity required downstream, the level and enforceability of a new developer's intellectual property rights, and the developer's ability to relicense the source code with his or her additions under a different (more restrictive and potentially more lucrative) license. The Android operating system uses Open Source Initiative⁷ approved licenses: the underlying Linux kernel is licensed under the GPLv2,⁸ and the user facing software infrastructure of the Android platform is licensed under Version 2 of the Apache Software License (Apache License).⁹

What does this mean for programmers and developers working on the platform as part of the overall Android Open Source Project (Project)? It means that Android is the first free, open source, fully customizable mobile platform to provide an operating system, middleware, key mobile applications, and APIs for third-party developers.

First, in line with the principles and requirements of the Apache Software Foundation,¹⁰ including supporting collaborative development across both nonprofit and commercial organizations, individual developers of code (and corporations that have assigned employees to work on the code governed by the Apache license) must complete and sign an Individual or Corporation Contributor License Grant.¹¹ These agreements set out the legal terms under which development can occur and are put in place to protect the contributor/developer, as well as the Project.

Second, under the Apache License and the Project, developers are not required to re-license their creations under the same license terms as they are granted. For the development entrepreneur, this fact is significant because it allows the developer to use the open source code for purposes outside of Apache, and to use it without having to turn proprietary enhancements back over to the open source software community.

Finally, the Apache License also carries some notable legal protections for developers and users. Under the Apache License (1) there is an explicit grant of a "no-cost" copyright and patent license, and any litigation by a licensee claiming that any work under the Apache license infringes a patent has his or her license automatically terminated; (2)



there is a disclaimer for noninfringement to protect the developer from downstream user claims; (3) there is an express prohibition against the use of trade names, trademarks, or service marks or product names of the licensor (except to the extent required for customary use to describe the origin of the work) that could inhibit downstream use of code; (4) if the original work includes a notice text file, then any derivative work must include a copy of the attribution notices within such text file so that they remain in any derivative work. These may seem to be arcane legal points, but each has practical significance to a developer, as pointed out below.

The first exception is huge because applications developed on the Android platform could contain concepts or ideas that are patentable. As the smartphone market matures, larger corporate developers with a history of obtaining patents are going to be major players. Development on a different platform or use of different code or even completely independent creation of an application (without reference to prior apps) doesn't protect a potential subsequent infringer of the patent rights, as it is the concept that is protected under patent, not the code itself. Additionally, "intent" to infringe isn't required. If a software developer uses or incorporates someone else's patent into his or her new code, the software developer may become an infringer, whether the software developer intended to infringe or not. What this means for the open source developer is that large corporate developers can no longer freeze other competitive developers out of the market by assertion of their prior patent positions (as is possible in the Apple development OS environment). In addition, developers are protected from third-party liability to licensees (with the additional "stick" of revoking any license to a licensee that makes such patent assertions).

The same is true with respect to issues regarding infringement. In most commercial environments, and with most proprietary system development, every developer usually has to make a representation and warranty that the new application code doesn't infringe any third party rights (and therefore becomes financially responsible for any losses to their licensee related to such infringement). With the disclaimer in the Apache License, a developer can contractually rid his/herself of that liability to licensees. Additionally, consistent with the Apache License's position regarding the disclaimer, the Apache License does not require that a developer provide an indemnity for claims of infringement.

Regarding the use of trade names and trademarks, the Apache License's prohibition against their use (except for the reason stated above) supports the goals of the Apache Software Foundation by ensuring that the code and development can be used for the benefit of the community while still recognizing and supporting the work of previous developers.

And the Winner Is...?

So who wins, Apple with the iPhone, Motorola with the Droid, Google with the Nexus One, or HTC with the Droid Incredible? Only time and the market will be able to provide the answer.

Regardless of the future outcome, device manufacturers and application developers manufacturers and application developers, using the Android operating system are well positioned to forge ahead into the lucrative market for smartphone and mobile applications. The advantages are clear. No proprietary system restrictions. No centralized approval process. And a legal environment in open source that has survived and prospered with PCs and is now poised to stretch to the next level.

¹ Apple released the iPhone 4G on June 7, 2010, exclusively on the AT&T network.

² The Motorola Droid was introduced on Nov. 6, 2009, exclusively on the Verizon network. Google launched the Nexus One on Jan. 5, 2010. Verizon also released the Droid Incredible by HTC on Apr. 29, 2010.

³ Jenna Wortham, Apple's Game Changer, *Downloading Now* N.Y. Times, Dec. 6, 2009, at BU1

⁴ The Android Operating Systems has been out for more than two years and used for smartphone offerings from Sony Ericsson, Samsung and others; however with respect to the Droid, Nexus One, and Droid Incredible, Motorola, Google, and HTC, respectively, have chosen to put a stake in the ground through use of the open source model for developers.

⁵ The Software Freedom Law Center filed several lawsuits in the Southern District of New York beginning in late 2007, for noncompliance with GPLv2 (by failing to distribute derivative software source code to downstream recipients.) Almost all of these cases have been settled based upon money damages, agreements to comply with GPLv2, notifying software recipients of the GPLv2 license terms, and requiring defendant to engage an open source compliance officer. See also *Jacobsen v. Katzer*, No. 2008-1001 (Fed. Cir. Aug. 13, 2008), where the Court of Appeals for the Federal Circuit held that that open source "Artistic License" created copyright conditions that could be enforced pursuant to the Copyright Act (including injunctive relief), not just a contractual obligation as between the licensor and licensee.

⁶ See generally, Frey Kelly L. Sr., Kelly L. Frey, II, and Courtney Smith, *Open-Source Software and IP, INTELLECTUAL PROPERTY DESKBOOK FOR THE BUSINESS LAWYER*, American Bar Association Section of Business Law (2d ed. 2009).

⁷ See www.opensource.org. The Open Source Initiative (OSI) is a nonprofit corporation formed to educate about and advocate for the benefits of open source.



⁸ See www.gnu.org/licenses/gpl-2.0.html. The General Public License (GPL) is the most widely used open source license and is controlled by the Free Software Foundation. Version 2.0 of the license was the cornerstone license for open source development until the release of Version 3.0 in 2007.

⁹ See www.apache.org/licenses/LICENSE-2.0. Oracle America, Inc. (who acquired Sun Microsystems, Inc. and Sun's Java platform in January 2010) filed a patent and copyright infringement action against Google, Inc. on Aug. 12, 2010, in the Northern District of California alleging that Google used certain Java-derived technologies within the Android operating system without the proper commercial license.

¹⁰ See www.apache.org.

¹¹ See <http://source.android.com/license/individual-contributor-license--android-open-source-project> (Individual Contributor License Grant) and <http://source.android.com/license/corporate-contributor-license--android-open-source-project> (Corporate Contributor License Grant).

*Co-Authored with Geoffrey P. Vickers

FOR MORE INFORMATION, CONTACT:



Kelly L. Frey, is a member in Dickinson Wright's Nashville office and can be reached at 615-620-1730 or kfrey@dickinsonwright.com.