

Source-code discovery in patent infringement cases

No protective order can anticipate all contingencies, but an early and informed discussion can simplify the process.

BY ANDREA L. GOTHING AND PETER M. ROUTHIER

Many of today's consumer electronic devices contain embedded software. Frequently this software is automatically updated over the Internet to add new functions or fix bugs. Indeed, anyone with a smartphone is familiar with frequent "updates available" messages. While frequent software updates may make for more feature-rich and robust devices, it creates unique issues for patent litigants involved in source-code discovery.

On the one hand, a plaintiff patent holder frequently requires access to dozens of versions of source code in order to prove infringement. On the other hand, the defendant (or a third-party software provider) often has legitimate security concerns about disclosing its code. Unfortunately, these opposing interests create an environment for discovery disputes. Further complicating the situation is that source-code identification, production and review can be both time consuming and costly. Finding adequate time to identify, produce and review source code can be challenging when faced with discovery and expert deadlines.

Parties may avoid discovery disputes, however, with prompt agreement on two key aspects of

source-code discovery. First, the parties should negotiate a protective order that satisfies both the patent holder's need for access to source code as well as the producing party's need for security. Second, if multiple versions of software are at issue, the parties should attempt to agree on one or more representative versions. If an agreement cannot be made, then the plaintiff may find it necessary to seek source code for all accused versions.

As an initial matter, it is important to understand what source code is and why it is important to software patent cases. Source code is the human-readable — albeit oftentimes cryptic — set of instructions underlying software. It dictates the actions to be performed by a computer or embedded processor, but it does not do so directly. Instead, "source code" is typically compiled by a special program called a compiler into "object code." As a result, consumer electronic devices do not themselves contain source code — they contain object code.

Because humans cannot readily read object code, a plaintiff usually needs a device's source code to determine how it accomplishes its accused functions. But because even a single version of source code may comprise millions of lines, finding the accused functionality can be like looking for a needle in a haystack. See, e.g., *Lucent Techs. Inc. v.*

Gateway Inc., 580 F.3d 1301, 1332 (Fed. Cir. 2009) ("Outlook consists of millions of lines of code, only a tiny fraction of which encodes the [accused] feature"). Even with the aid of analysis tools, this review can take a substantial amount of time. Against this backdrop, patent holders facing even a moderately ambitious scheduling order must obtain access to source code as soon as possible.

PROTECTIVE ORDERS

Not surprisingly, defendants and third-party software providers are not eager to produce their confidential source code, which may be subject to trade secret protection. See *LinkCo Inc. v. Fujitsu Ltd.*, 230 F. Supp. 2d 492, 499 (S.D.N.Y. 2002). But being proprietary or confidential does not shield source code from production. When "source code is subject to very stringent protection," confidentiality concerns are often alleviated. See *Leader Techs. Inc. v. Facebook Inc.*, No. 08-862-JJF-LPS, slip op. at 7 (D. Del. Sept. 4, 2009). For instance, in the Facebook case, the U.S. District Court for the District of Delaware rejected Facebook's claim that it would be subject to "overwhelming" prejudice by virtue of having to produce the entirety of its source code. The court emphasized that the defendant's source code was available to only a few individuals on a non-networked, password-protected computer

terminal, with no printing allowed.

Production of source code under these circumstances is becoming the norm. In fact, the protective order in the Facebook case closely tracks the provisions of the District of Delaware's recently enacted Default Standard for Access to Source Code. Under the default standard, absent an agreement between the parties, source code must be produced on a password-protected, stand-alone computer located at an independent escrow agency. The parties must share the costs of production, and access to the code is limited to two outside lawyers and two experts. Notably, no printouts are allowed.

Although the default standard provides a good framework, it is not without traps for the unwary. Foremost among these is any deadline governing supplemental infringement contentions. In some cases patent holders have been ordered to supplement their infringement contentions with new information as soon as 30 days after gaining access to source code. See, e.g., *Uniloc USA Inc. v. Sony Corp. of Am.*, No. 6:10-cv-373-LED, slip op. at 6 (E.D. Texas May 20, 2011). Given the sheer volume of code, and the limited access to escrow, a 30-day deadline may be unfair in some circumstances. Therefore, before stipulating to an order tracking the default standard, the plaintiff should consider local rules or court-ordered deadlines that could necessitate greater access to the code. If greater access is unavailable, an agreement to extend the deadline for supplemental contentions may be needed.

Another potential trap is that the default standard does not allow source code to be printed without agreement of the producing party or order of the court. If this issue is not addressed early, the lack of printouts may prejudice a plaintiff's ability

to prepare for depositions, expert reports, motions and trial. Perhaps in view of this difficulty, the sample protective order provided by the U.S. District Court for the Eastern District of Texas allows for "a reasonable number of printouts," as well as for use of those printouts in depositions, reports and filings.

It may be difficult to provide more guidance than this at the outset, as what is "reasonable" will depend on the facts and the complexity of the case. But it is not unheard of for printouts to number in the thousands of pages. See, e.g., Plaintiff's Response to Defendants' Motion to Stay at 3 n.9, *JuxtaComm-Texas Software v. Axway*, No. 6:10-cv-11 (E.D. Texas May 23, 2011).

Finally, neither the default standard nor the sample protective order provides for electronic access to source code during a deposition. A complete electronic version of source code during fact and expert depositions is often more helpful for both parties than fumbling through thousands of pages of printouts — a fact that has been recognized by at least one court. *Nomadix Inc. v. Hewlett-Packard Co.*, No. 09-08441 (C.D. Calif. June 21, 2011) (noting that the deposition of an engineer familiar with the code "could not reasonably be accomplished by poking through reams of paper").

REPRESENTATIVE VERSIONS

Once a protective order is in place, the parties should discuss stipulating to representative versions of code. Not only do representative versions reduce the plaintiff's burden of proving infringement, they may reduce the defendant's burden of production. The defendant should be aware, however, that designating a version as representative may foreclose noninfringement positions buried in the differences between versions. Therefore, depending on the

complexity of the accused software, a stipulation regarding representative versions might require an initial production of one version. An initial production may enable the plaintiff to identify accused portions of code so that the defendant can determine whether that portion is representative of other versions.

If the parties cannot agree to representative versions, in most instances the plaintiff should seek all accused versions as well as all documents describing the differences between versions like change logs. If the defendant refuses to produce this information, the plaintiff should promptly seek relief from the court so as to avoid pitfalls associated with a lack of diligence or delay. See, e.g., *Sybase Inc. v. Vertica Sys. Inc.*, No. 08-cv-24 (E.D. Texas Nov. 30, 2009) (denying plaintiff's motion to supplement contentions due to a lack of diligence).

Ultimately, while no protective order can anticipate all contingencies, and while a stipulation to representative versions may never be reached, an early and informed discussion of these issues can simplify source code discovery for all involved.

Andrea L. Gothing is a partner, and Peter M. Routhier is an associate, in the Minneapolis office of Robins, Kaplan, Miller & Ciresi. They represent clients in complex patent infringement matters.