Licensors of software typically utilize software license agreements providing for their ownership of the licensed software and related IP, as well as restrictions barring licensees from reverse engineering the code at issue. The scope of protection, of course, depends on the final language of the licensing agreement and disputes can arise when licensees decide to develop similar software in-house, or with a third party. Indeed, a recent case, *Ford Motor Co. v. Versata Software Inc.*, No. 15-10628 (E.D. Mich. Sept. 7, 2018), tackled some of these issues.

In this case, a Michigan district court found that broad and general software ownership provisions extended to trade secrets embedded in the software at issue, but that a fairly expansive definition of “reverse engineering” might not be broad enough to cover certain activities associated with developing similar software. Ultimately issues of fact precluded many of the grants of summary judgment that both parties sought, but the case offers interesting insights with respect to software licensing practices on at least those two issues.
According to the opinion, Versata Software Inc. (“Versata”) licensed automotive configuration software to Ford Motor Co. (“Ford”) for more than two decades. In 2014, however, Ford decided to replace Versata’s software with software it was developing internally and therefore elected not to renew its licensing agreement with Versata. Predictably, this led to an IP dispute between Ford and Versata. Both parties sought summary judgment on their causes of action, and, in a mixed ruling, the court found issues of material fact prevented judgment on most – but not all – claims.

Ford sought a declaratory judgment seeking, among other things, a ruling that it had not infringed any trade secrets owned by Versata. Versata filed counterclaims alleging, among other things, misappropriation of such trade secrets. In resolving the question, the court looked to specific contract provisions and concluded that when Ford agreed to the last agreement between the parties in 2004, it “irrevocably acknowledged” that “[s]ubject to the licenses granted [therein], Ford has no ownership interest in the Software....” Based on this “unambiguous acknowledgment,” the court ruled that Ford cannot now claim an ownership interest in the trade secrets embedded in the Versata software. The court rejected Ford’s argument that even if it disclaimed ownership in the software, it might still claim an ownership in the relevant trade secrets, stating it was unpersuaded that the trade secrets were distinct from the software that Ford “irrevocably” acknowledged it did not own.

While finding in favor of Versata on the trade secret ownership issue, the court declined to rule as a matter of law that Ford breached the agreement by allegedly performing certain acts that Versata considered “reverse engineering” of its software. Versata sought summary judgment alleging Ford “reverse engineered” the Versata software when Ford inputted identical data into both the Versata software and its internally developed software and then “studied” the “outputs” from Versata’s software to ensure its own software produced similar results.

The court examined the definition of reverse engineering in the Agreement. Section 1.7 of the Agreement defines “reverse engineering” as follows:

Disassembling, Decompiling, and reverse engineering include[s], without limitation, “(i) converting the Software from a machine-readable form into a human-readable form;” [.....] (iii) examining the machine-readable object code that controls the Software’s operation and creating the original source code or any approximation thereof by, for example, studying the Software’s behavior in response to a variety of inputs;” [and] (iv) performing any other activity related to the Software that could be construed to be reverse engineering, disassembling, or decompiling.

The court found that provision of the Agreement ambiguous:

“In particular, it is not clear whether Section 1.7 prohibits any and all “studying [of] the Software’s behavior in response to a variety of inputs,” or whether Section 1.7 prohibits only such studying that is part of an effort to “examin[e] the machine-readable object code that controls the Software’s operation and create[e] the original source code or any approximation thereof....”
As a result of the court’s limited ruling, a resolution of the broader IP and contractual claims surrounding this software licensing dispute will have to wait for trial.

There are at least two lessons to learn from this opinion:

1. This case reinforces the basic point that broad IP ownership language is helpful for software licensors. However, in some situations – and particularly where software is being developed by the licensor at the request of, and with input from, a future licensee – that licensee may want to try to retain ownership or some rights to trade secrets being conveyed to the software developer (e.g., the form of the custom requirements for the software). In this case, the court was unpersuaded that the trade secrets were distinct from the software. Had the license agreement included language that allowed the licensee to use concepts, methodologies or know-how embodied in the software, the outcome might have been different.

2. Certain “boilerplate-ish” terms of software licensing agreements are due for a revisit. For example, in this case, the court found that a traditional definition of reverse engineering was ambiguous in light of certain practices with .jar files and PDO (PHP Data Object) repositories. Licensors and licensees alike should look at their forms and older agreements, keeping in mind how software and tools and practices have evolved and continue to evolve at a fast pace.

Regardless of the result, this case – which concerns a contract written ten years before the litigation commenced – demonstrates how fast technology can outdistance the ability of license drafters to provide for all of the expected issues that might arise in a software licensing arrangement.

© 2021 Proskauer Rose LLP.

National Law Review, Volume VIII, Number 281

Source URL: https://www.natlawreview.com/article/common-software-licensing-language-issue-ip-dispute