EXPLAINING THE ALMOST UNEXPLAINABLE: SOURCE CODE EVIDENCE AT TRIAL

Part one of this two-part series explores early discovery considerations with source code evidence and their effect on trial.

By David A. Prange and Benjamen C. Linden, Robins Kaplan | May 04, 2020

The past decade has seen a proliferation of products and services that use computer networks, the Internet of Things, or the Industrial Internet of Things. This technological evolution has resulted in increased complexity in litigation involving computer-based products and services. In many types of litigation, such as claims for patent infringement, copyright infringement, trade secret misappropriation, or software licensing disputes, computer source code is at the forefront of the case.

The use of source code evidence at trial, however, introduces unique challenges; principally, the evidence is very diffcult to understand for the average fact finder. This two-part series discusses some strategies litigators should consider to effectively present source code evidence beginning with discovery and through trial.

BACKGROUND

Generally, source code refers to human-readable programmed instructions that specify how computing functionality of a product is logically organized and operates. Each computing function—for example, pressing a button on a smartphone screen or printing a document—has associated source code written by a programmer in a particular programming language. This source code may ultimately take the form of software, firmware, or be implemented as hardware. Software or firmware source code is generally converted (i.e., compiled) into binary form before it can be run on a microprocessor. Hardware source code is similarly converted (i.e., synthesized) into a description of an electronic circuit.

During discovery, it is not uncommon to have source code produced on a secure computer available only to outside counsel and retained experts. Protective orders governing the production of source code largely prohibit dissemination outside this secure environment. Physical copies of the source code are almost always severely limited to a small number of printouts that are far less than the entire source code production.

In view of these restrictions, presentation of this evidence to the fact finder is a signifcant challenge. Source code is not readily understandable to the average juror. Unlike other documents, like contracts or emails, the average juror rarely has the background to make liability determinations based on their own understanding of the source code. Still, planning and early strategic consideration of these difficulties can be employed to make presentation more efficient and understandable to the fact finder. In this first part of the series, pretrial strategies such as negotiation of protective order provisions and securing the proper foundation for source code evidence are discussed.

PROTECTIVE ORDER CONSIDERATIONS

The protective order presents an early opportunity to shape the use of source code at trial. In large part, protective orders are entered by stipulation of the parties and address the handling of confidential information, including source code, during discovery. A stipulated protective order, however, can have little effect on the actual treatment of source code evidence at trial. (For example, see the N.D. California Model Stipulated Protective Order (https://www.cand.uscourts.gov/forms/model-protective-orders/) for Litigation Involving Patents, Highly Sensitive Confidential Information and/or Trade Secrets: “Any use of Protected Material at trial shall be governed by a separate agreement or order”).

Still, if source code evidence may be used at trial, a party should consider this during negotiation of the protective order to ensure that the option is not later foreclosed. For example, considerations should include specifying (1) procedures for putting parties (including third parties) on notice of the use of their source code at trial; (2) the number of extra copies of printed code permitted for trial; and (3) the handling and destruction of these extra copies. Parties should also consider specifying the use of watermarked or colored paper for source code printouts to allow for better tracking, and hence protection, of the material. These provisions can help avoid trial surprises such as objections from an opposing party or third party on the use of information considered highly sensitive.

ESTABLISHING SOURCE CODE ADMISSIBILITY THROUGH DISCOVERY

Before the fact finder may consider source code evidence, it must be admitted into evidence. In that regard, source code evidence is not unlike other evidence admitted at trial. Ideally, admissibility of source code can be established through deposition of a knowledgeable witness with first-hand knowledge of the source code. Alternatively, admissibility may be achieved through declarations that meet the requirements of FRE 902(11) ("Certified Domestic Records of a Regularly Conducted Activity").

A declaration under FRE 902(11) should address that the source code is a business record under FRE 803(6) (A)-(C). The declaration, from a knowledgeable person, should clearly address when and how the source code was created, how the source code was maintained and documented, and the extent to which source code development is part of the regular practice of the business. Procedurally, a party intending to rely on the declaration must provide reasonable written notice to the adverse party of this intent. The party must also make the declaration and source code evidence available for inspection in advance of trial.

Failure to follow the requirements of FRE 902(11) could result in exclusion of the evidence or worse. For example, in *Wi-Lan Inc. v. Sharp Elecs. Corp.* (Feb. 14, 2019, D. Del.), the trial court found that a declaration did not meet the requirements of FRE 902(11) because the declarant offered only conclusory statements that were facially inconsistent with comments and metadata in the underlying source code. As a result, the source code was inadmissible and summary judgment granted for failure of proof. Accordingly, it is important for a litigant to start thinking early about how the source code evidence will ultimately be admitted at trial.

CONCLUSION

If source code evidence is needed at trial, there are easy early steps that can be taken to secure proper foundation for its admissibility. With the proper evidentiary record, the next task is preparing for and presenting the source code evidence in court. The second piece of this two-part series will address that topic, including selecting the proper witness to testify, sealing the courtroom at trial, and the logistics of presentation in court.

David Prange is a partner at Robins Kaplan LLP and leads the trade secrets subpractice. His practice focuses in complex business litigation with an emphasis on intellectual property, including patents, trade secrets, trademarks, and licensing disputes in federal and state courts across the United States. He has litigated and tried multiple cases to successful verdicts involving source code evidence. DPrange@robinskaplan.com (mailto:DPrange@robinskaplan.com)

Benjamen Linden is an associate at Robins Kaplan LLP. His practice is focused on high-tech litigation, including patent infringement, trade secret, licensing, and outsourcing disputes. He has experience presenting and managing the development of source code evidence through all stages of litigation up to and through trial. BLinden@robinskaplan.com (mailto:BLinden@robinskaplan.com)