

Source Code Repositories: Authenticating Production of Source Code

[Josh Siegel](#) | Senior Manager & Forensic Examiner

The reliable record of changes maintained by source code repositories makes them among best evidence an expert can be provided for the purpose of authenticating source code production.

A source code repository tracks the development of a program by maintaining native source code files that can be examined as they existed throughout the history of the program's development. Each change made to a source code file can be recorded by the repository, and it is difficult to alter the data within a repository without leaving traces of the alteration.

For example, source code repositories often tie a unique, sequential ID number to each update of code. A gap in the sequence of code updates may indicate that the repository has been altered.

For example, source code repositories often tie a unique, sequential ID number to each update of code. A gap in the sequence of code updates may indicate that the repository has been altered. Similarly, if a program was purportedly developed over the course of several years, but all of the code contained in a produced repository was added to the repository on the same day, the produced repository is probably not the repository used during the development of the software.

When an expert lacks access to a source code repository, he or she can still potentially authenticate produced source code if provided with individual files in native format. Although files produced outside of a repository are more easily altered, files in native format still contain metadata that may allow an expert to authenticate evidence. Metadata is data about data, and experts review metadata that records such information as the date of creation and last date of modification for computer files produced as evidence. For example, if all three of the following conditions were true, they would be strong indicators that produced code is authentic:

- The party producing the code states that the year of completion for a version of code is 2005.
- The produced code contains only files with last modified dates prior to 2005.
- There are no obvious omissions from the produced code.

However, even where code is produced in native format as in the example above, individual source code files may be drawn from several different versions of a program. Access to a source code repository allows an expert to verify whether produced code constitutes a true and accurate copy of a version of a program as it existed at a certain time, or if the produced code was reconstructed from several different versions. Additionally, source code in native format but produced outside of a repository can omit files containing evidence of copying. Access to the repository allows an expert to evaluate the completeness of a production of source code.

Because converting a file out of its native format may alter or delete data that might have been used to authenticate the file, source code produced in a non-native file format is the most difficult to verify.

Unless provided with more information, an expert may be unable to authenticate code produced in static formats such as paper printouts or text images (e.g., PDFs). For example, when filing a copyright, only the first 25 and last 25 pages of a program must be submitted to the copyright office. When this “deposit copy” of a program contains the program in its entirety, an expert can compare it to produced code for the purposes of authenticating the produced code. However, this method is limited to small programs and cannot rule out the possibilities that the copyright filing itself contains errors or that the documentation submitted to the copyright office is a reconstruction.

When this “deposit copy” of a program contains the program in its entirety, an expert can compare it to produced code for the purposes of authenticating the produced code.

For all of these reasons, the source code repository is instrumental for verifying the completeness, authenticity, and validity of a source code production.

Read the first installment: [Source Code Repositories: What is a Source Code Repository?](#)

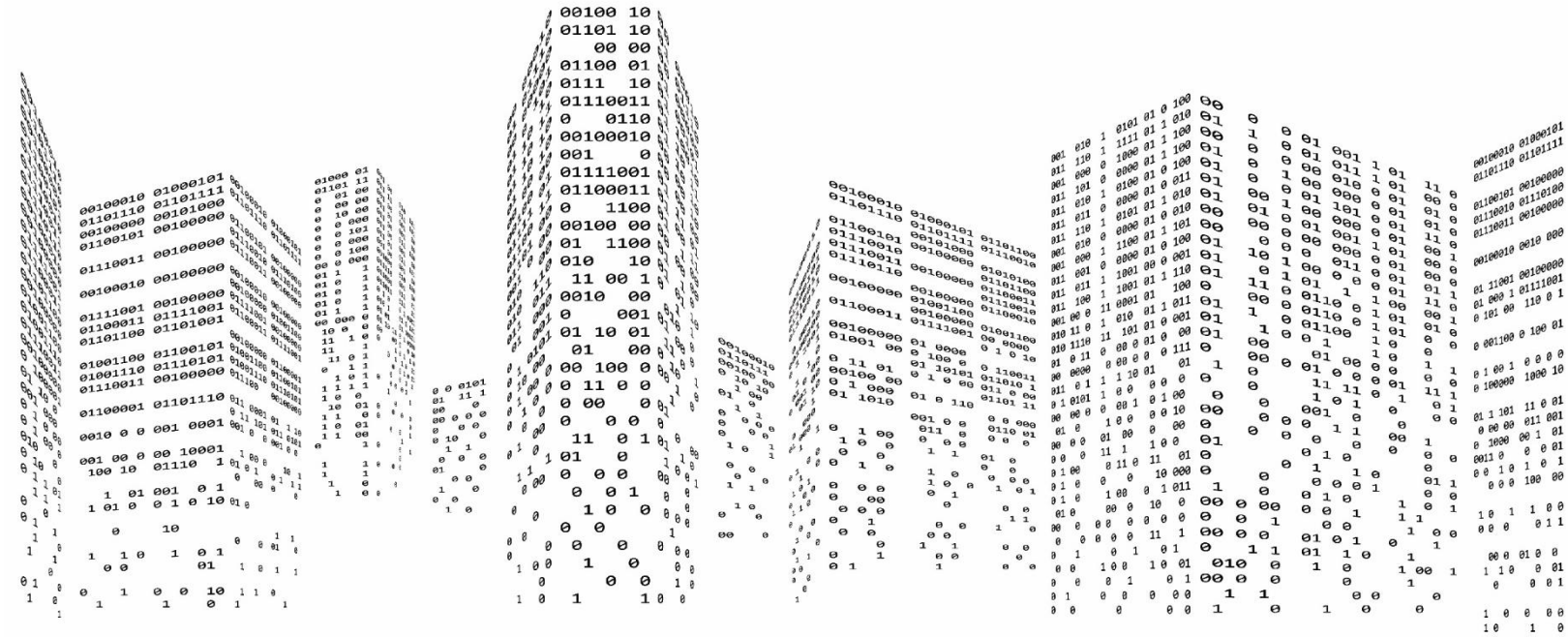
Read the second installment: [Source Code Repositories: Reviewing the Right Version of a Program](#)

Josh Siegel

Senior Consultant & Forensic Examiner



Josh Siegel has substantial experience analyzing copyright, patent, and trade secret claims related software and information technology. Josh performs functional testing, analyzes defect systems and metadata, examines source code in intellectual property disputes, acquires and analyzes data in digital forensics, and finally integrates that data into written reports and testimony.



If you are an attorney in need of an intellectual property expert, we invite you to consider [DisputeSoft](http://DisputeSoft.com).

Contact Information

Jeff Parmet, Managing Partner

301.251.6182

jparmet@disputesoft.com

12505 Park Potomac Ave. | Suite 475 | Potomac, MD | 20854