



Dead Code, the Law, and Unintended Consequences

By Michael Barr

What is Dead Code?

Dead code is source code that is not executed in the final system.

Dead code comes in two forms. First, there is dead code that is commented out or removed via `#ifdef`'s. That dead code has no corresponding form in the binary. Other form of dead code is present in the binary but cannot be or is never invoked.

In either case, dead code is a vestige or unnecessary part of the product.

The Legal Consequences of Dead Code

Dead code has been frequently encountered when conducting source code reviews for litigation. The presence of dead code can have unintended legal consequences and several scenarios have been observed in which dead code heightened the probability of a loss in court.

Scenario 1: Dead code implements part (or all) of the algorithm.

When a patent infringement suit is brought against an electronics or software company, one or more versions of the involved product's relevant source code must be produced to the plaintiff's legal team.

Both the patent owner's and the plaintiff's expert(s) will examine the source code(s) to identify portions of the code that implement each part of the algorithm. If one of the identified parts is implemented in dead code that becomes part of the binary, the product may still infringe an asserted claim of the patent—even if it is never invoked.

Tip: It is left to the legal teams to determine if dead code does legally infringe. However, it is

possible that neither side's software expert(s) will notice it is dead. It is also possible that the judge or jury won't be convinced by a dead code defense.

Scenario 2: Dead code leads to overly-complex and vulnerable code.

In cases such as a product liability suit involving injury or death, a software expert examining the source code may use poor code quality as a basis of their opinion. This scenario can occur when the reviewed source code is overly-complex and riddled with commented out code and/or preprocessing directives.

Source code that is hard to read is harder to maintain. Code that is hard to read and maintain is also more likely to contain bugs. In such a scenario, the defendant's engineering team may appear to be sloppy or incompetent to the jury.

Tip: Overly-complex code also increases the cost of source code reviews – as both side's experts will need to spend more time examining the code to understand it fully.

In a source code copyright (or copyleft) suit the mere presence of another party's source code may be sufficient to prove infringement—even if it is isn't actually built into the binary.

Tip: Litigators should be aware that developers of electronics and software products are at risk of code containing files or functions of open source software that, by their mere existence in the source code, attaches an open source license to all of the proprietary code.