

EDAC License Management and Antipiracy (EDAC-LMA) Position on External License Reclaiming

10/24/2013

Summary statement: Any attempt to externally reclaim/harvest licenses via individual process suspension or similar methods is very risky and is generally unsupported by EDA vendors.

Details: A product's licensing algorithms are designed around the contractual usage of the product. An external license reclaim tool may circumvent that licensing behavior, causing violations of the customer's usage agreement and creating significant business risk for the EDA vendor. Furthermore, there are many scenarios that would fail, possibly with data corruption or loss. These failures will be difficult to identify and diagnose when the user calls the EDA vendor's customer support organization -- they will appear to be application bugs. EDAC-LMA member companies specifically do not support using this technique, except where the EDA vendors have implemented it themselves in a manner that will be safe, effective, and in accordance with the product's business model.

Here are some of the specific technical issues:

- Many EDA products have complex architectures with multiple dynamic processes, open files, open sockets for inter-process communication, processes on different computers, etc.). It would be difficult or impossible for a customer to know which processes to safely suspend. Also, there would be no way to close open files, etc. before suspension, which would leave the application in a very precarious state. This scenario is worse than a computer's "sleep" mode, since non-suspended processes and OS services would continue to act on the application's data and program state. Of course, these factors present risk of data corruption or loss.
- When un-suspending an application, a license reclaim tool would have no way of discerning the required process-wakeup sequencing and timing for multiple interrelated application processes.
- EDA products have various exception-handling behavior. If an external license reclaim tool causes a product to unexpectedly lose its license, there is no guarantee the application will handle that fact gracefully, possibly resulting in data loss.
- A product may already include idle detection for license reclaiming, and external process suspension could interfere with that.
- A product may disable TIMEOUT for a reason, and any external process attempting to force licenses to be released anyway could trigger data loss.
- In some products, the FlexNet-licensed executable can be "idle" for many hours while a separate compute engine is busy. The compute engine contains no FlexNet code and has a proprietary secure connection to the FlexNet-licensed parent. If an external license reclaim tool suspends the parent, it could result in data loss.

- A license reclaim tool will have trouble accurately determining if an application is idle. Only EDA vendors fully know what "idle" looks like internally for a particular product. Attempts to use mouse clicks, CPU utilization, memory footprint, etc. may not accurately model the different product usage patterns, possibly causing an application to be suspended while it is actually still in use.
- Duplicate grouping and license packaging adds additional complexity and risk. There is no way for an external license reclaim tool to know when multiple instances of an application are sharing the same license.
- Users sometimes mix node-locked uncounted licenses with licenses from a license server. An external license reclaim tool, unaware of product licensing details, could suspend an application that is currently using a node-locked uncounted license, when it should be left alone.
- Many EDA products change their behavior based on the license features that were checked out. If an application is suspended and then restored, it is possible the product will get different license features than initially checked out, resulting in unexpectedly different application behavior.
- Even if some testing is done and a license reclaim tool seems to work, there is a high probability that testing did not cover all cases that could be encountered under actual usage. The result could be a false sense of trust in the technique, risking data loss at times when it would be especially unfortunate.