

How Corporations Want to Buy Their Software.

Best Practices in Implementing Licensing

Dan Griffith

Motorola

Manager, Comprehensive Software Asset
Management team (CSAM)

What Do Corporations Want In Licensing

Software is a nebulous commodity.

You can't touch it nor can you put an asset tag on it.

Today it is difficult to measure utilization or verify a license file matches what was purchased.

What Do Corporations Want In Licensing

Corporations need to be able to calculate their return on software investments.

They need to be able to measure usage and justify new purchases.

They need software pools that meet peak and fluctuating engineering demands without buying “too much.”

They want software licensing that supports these needs.

What Do Corporations Want In Licensing

- **What do corporations want in software licensing?**
 - **Standard Behavior**
 - **Measurability**
 - **Flexibility**

Standard Behavior

- **Standard Behavior of license check-in and check-out.**
 - **Check-out time should be less than one second over a network with 1MB of bandwidth.**
 - **Licenses should be released and be checked back in once an engineer closes their session or if the session is idle, as defined in by the TIMEOUT option file feature**
 - **Primary features should be able to roll over to an alternate server to obtain a sub feature**
 - **Licenses should be backward compatible**

Standard Behavior – Check-Out & Check-In

- **Motorola has encountered products that take as long as 15 minutes to check-out over a fast network.**
 - Traced to legacy code in the binary attempting to checkout as many as 30 license features that no longer exist.
 - This not only wastes precious engineering time but also increases the quantity of licenses required during peak periods
- **Motorola has encountered products that do not support the FLEXIm TIMEOUT option**
 - Idle licenses remain checked out until the client closes the application ignoring any TIMEOUT setting in the option file.
 - This causes the purchase of more licenses than is required.

This is not what corporations want!

Standard Behavior – Sub-Feature Roll Over

- **Standard Behavior** in ability of product sub-features to be obtained from a server other than the server providing the primary license
 - **Example:** The fictitious product Moon ST costs \$250K per license and is shipped with two license features. Company WWT Design requires 50 licenses and splits these licenses across two servers for redundancy.

<u>Product</u>	<u>Feature</u>	<u>Server A</u>	<u>Server B</u>
Moon ST	Moon_ST	25 Licenses	25 Licenses
	Moon_inter	25 Licenses	25 Licenses

Standard Behavior – Sub-Feature Roll Over

Sub Feature Roll Over

<u>Product</u>	<u>Feature</u>	<u>Server A</u>	<u>Server B</u>
Moon ST	Moon_ST	25 Licenses	25 Licenses
	Moon_inter	25 Licenses	25 Licenses

Moon ST has a \$50 Timing option for which WWT Design only requires 14 licenses. They too are spread equally across both servers.

<u>Option</u>	<u>Feature</u>	<u>Server A</u>	<u>Server B</u>
Moon Timing	Moon_Time	7 Licenses	7 Licenses

Standard Behavior – Sub-Feature Roll Over

Sub Feature Roll Over

Product	Feature	Server A	Server B
Moon ST	Moon_ST	25 Licenses	25 Licenses
	Moon_inter	25 Licenses	25 Licenses
Moon Timing	Moon_Time	7 Licenses	7 Licenses

Due to a non-standard implementation of FLEXIm, Moon_ST must get its Moon_Time licenses from the same license server that Moon_ST was check out from.

If all 7 Moon_Time licenses on Server A are in use and an 8th Moon ST from Server A requires Moon_Time, that job will fail. Even though Moon_Timing licenses are available on Server B. This forces the customer to put 14 Moon Timing licenses on each server, double what is required by their engineers.

Standard Behavior – Sub-Feature Roll Over

Sub Feature Roll Over

Primary products must be enabled to call sub-products, sub-features, or associated products from any license server defined in the users license path.

Measurability

- **Corporations need to be able to measure their software usage and understand their usage requirements and ROI on these investments**
- **Lack of primary license features makes this difficult at best and often impossible.**
 - **Primary license feature is defined as a unique license feature that is checked out every time that product is checked out.**
 - **Primary license features must not be utilized by other products.**
 - **Primary license feature must clearly identify the product.**

Measurability – Primary License Features

In this example, usage of Design Force cannot be measured because its primary feature is also used for two other products.

Product: Design Force

Design-Force (sub license feature)

Compiler-Force (*primary license feature*)

DF-mgt (sub license feature)

Product: Design Force Ultra

Design-Force (sub license feature)

Compiler-Force (*primary feature of Design Force used as sub feature*)

DF-mgt (sub license feature)

DF-Ultra Force (*primary license feature*)

Product: SOC Force

Design-Force (sub license feature)

Compiler-Force (*primary feature of Design Force used as sub feature*)

DF-mgt (sub license feature)

SOC-opt (*primary license feature*)

SOC-Interface (sub license feature)

Measurability – Primary License Features

Other examples of product to feature mapping issues

Product	Feature	Issue
Turboplace	83421	Primary name feature does not identify product. Use of numbers provides little information.
5960	ViewEdit	Primary name feature does not identify product
AI_Enterprise	200 aimgt env_ai	Use of numbers for sub-features is confusing Primary feature name does not identify product
Stargate	gate_si sun_ex	No primary feature exists Either feature might be checked out when the product is invoked depending on function requested

Primary feature shown in bold.

Measurability – Primary License Features

- 1) **Every Product must have a primary feature.**
- 2) **The primary feature must be checked out every time the product is run, and must remain checked out the entire time the product is in use.**
- 3) **The primary feature name must be unique and cannot be shared with or utilized in other products. During the creation of new products or the bundling of products, the primary feature name of a previous product must not be utilized in the licensing of the new product.**
- 4) **The primary feature name must clearly identify the product it enables**

Measurability – Primary License Features

- **Rick Kline of Macrovision and I have worked together to developed standard naming convention for license features.**
- **The standard is simple:**
 - **ApplicationName.FeatureName**
 - **Example for a product's primary feature:**
 - flexbill.primary
 - **Example for a product's sub feature:**
 - flexbill.reporting
 - flexbill.decryption
 - **Example for features that are common to several products:**
 - common.log_viewer
 - common.export

Flexibility

- **Corporations want flexibility**
 - **Flexible license server configurations**
 - Single stand alone license servers
 - Three server quorums
 - Multiple Independent License Servers
 - **License Pool Remixing, License pool buffers**
 - Requires the ability to measure usage **(Primary Features!)**
 - **Pay Per Use and Overage Licenses**
 - Requires the ability to measure usage **(Primary Features!)**
 - **Access To New Technology**

Flexibility – Pay Per Use

- *Pay Per Use, Overage Models, or Software On Demand* has seen limited success because usage is difficult to measure and utilization is hard to define
- Lack of primary license features is standing in the way of tomorrow's licensing models
- Corporations want flexible models which provide access to peak license quantities & new tech without over buying.
 - This requires the ability to measure utilization.
 - This requires primary license features
 - Today's lose-lose scenario is:
 - Corporations purchase less than peak requirements which impact engineering. Limited new technology access.
 - **Software vendors lose out on revenue for peak demand and suffer slow adoption of new technology**

Summary

- **What do corporations want in licensing?**
 - **Standard Behavior**
 - Quick check-in and check-out of licenses
 - Server to server roll-over for sub features
 - Backward compatibility
 - **Measurability**
 - Unique identifiable primary license features for every product
 - **Flexibility**
 - Pool Remixing
 - Pool Buffers
 - Method of accessing licenses for peak demand and new technology

Summary

As the customer, corporations want to purchase software in a way that fits how they use it.

The business models of the software vendors need to more closely reflect the needs of their customers for a win-win scenario.

Credits

Motorola's Comprehensive Software Asset Management team (CSAM) manages software from 31 vendors, utilizing 20 central license servers at 3 US locations serving EDA licenses to over 15,000 designers globally at Motorola design centers in India, Israel, Germany, England, Russia, China, Australia, Japan, Ireland, Brazil, Singapore, and the United States

Special thanks to Ray Embry who leads our CSAM application development team, Jason Wiseman who leads our license support team, and Debbie Simpson, Jessie Burciaga, Andy Siebert, Dave Lopez, Alan Zelhart, Syed Imam, Hung-Phu Tran, and Michael Battle who make it all happen.