

## ***The solution to software licensing and maintenance for the oil and gas industry—only pay for what you use when you use it***

STUART A. JACKSON, S2S Systems, LLC, Missouri City, Texas, U.S.

Why should you pay for software when it isn't being used or for software functionality that isn't needed? This seems obvious but, until recently, a software client had little or no alternative. The only option was to purchase around-the-clock access to unneeded capabilities along with the needed ones, even though only a fraction of the software functionality is used a fraction of the time it is licensed.

However, now that most of us are connected via the Internet and internal intranets, a new concept, dynamic user locked software licensing (DULSL), is changing the situation. DULSL provides the client with the means to pay only for the functionality in the software they require and only for the time they actually use it. DULSL provides software licenses that are activated when the client starts the application and stops when the client exits the application. They can be transferred and installed on any machine the client wishes to work on.

This type of licensing should also hasten how a software evolves because developers can see which applications are used most frequently or for longer periods of time and, thus, they will likely concentrate on improving those more profitable features.

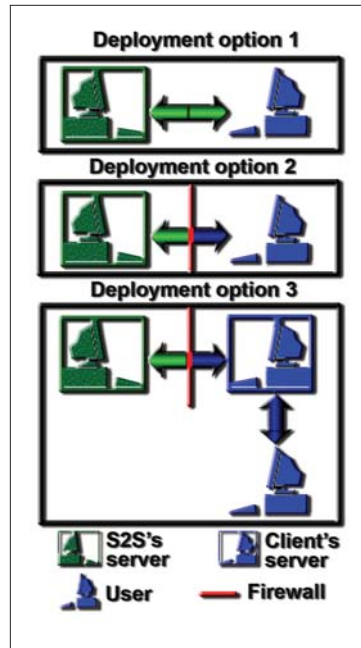
**Background.** The justification for charging clients for applications they never use or capabilities they do use when they aren't using them seems ridiculous, but it's the norm with virtually every software package.

The direct and indirect costs associated with traditional software licensing are put into relief by the following questions:

- How much time is spent maintaining and integrating different and incompatible licensing schemes?
- How much of the maintenance and support payments for a given software application are applied to desired enhancements?
- How often does the software vendor introduce new technology that isn't included under the software support agreement and therefore requires another capital purchase?
- What percentage of the software application is used?
- What percentage of the time an application is licensed for is the software actual used?

DULSL licenses, on the other hand, are workflow-derived and maintained over the Internet or a local intranet. Therefore, DULSL requires the application software to be web-deployable, workflow-derived, and multivendor capable. Each of these descriptors is fundamental to the complete solution.

The software needs to be web-deployed so that usage sta-



tistics can be accumulated, licenses can be transferred between hardware platforms, the software can be instantly upgraded as new and required functions or bug fixes become available, and required software capabilities can be provided on demand.

The software needs to be workflow-derived or, in other words, built and downloaded to the client as the client defines the workflow. Indirectly this process defines the required functionality because it eliminates all unneeded capabilities. A by-product of workflow-derived applications is that the application user interface is not cluttered with buttons and controls for functionality that is not required, thus making the user more efficient and increasing productivity while reducing licensing costs.

Finally, the software needs to be multi-vendor capable to allow seamless integration of proprietary software by software from third-party vendors.

**How it works.** DULSL requires that the client's hardware be initially connected to the vendor's deployment server for license verification and software definition and download. This connection can be through an internally maintained intranet server, or directly through the Internet to the vendor server. Once the software's function has been defined, it will operate whether connected to the server or not.

The software license is broken into four components: the calendar life of the license, the maximum number of hours of use allowed for the license, the calendar life of the ticket, and the maximum number of hours of use allowed for the ticket.

The license life and usage generally have upper bounds that the user will never exceed, and as long as invoices are paid under the terms of the license agreement the license will remain active.

These two licensing parameters control software access when the system running the software is connected to the server that deployed and licensed it. In a different scenario, the ticket life and usage parameters are used to maintain license control when the system running the software is not connected to the license server that deployed the software.

Tickets control how long in clock/calendar time and how many hours the client is allowed to operate disconnected from the license server. Another way to look at tickets is that they manage how long the software vendor will allow a particular client to operate the software without reporting back usage statistics for billing. A ticket is issued to a client the first time the software is downloaded and executed and remains on their system and accumulates usage statistics until it expires and a new ticket is issued. New tickets are issued by the license server and therefore require the client system to be connected to the license server.

A license is validated on the license server and a ticket is

validated on the client's system. If the client is connected to the license server, the ticket will be automatically refreshed every time its life or usage limit is exceeded. If the client is not connected to the server, the ticket will continue to accumulate usage statistics until the next time the client's system is connected to the server, at which time the usage information will be recorded on the license server and the ticket refreshed. Assuming that the client is paying software invoices on agreed-upon terms, the four licensing parameters would be set so that the client would never encounter any licensing issues whether running connected to the license server or not.

**Technical and cost benefits.** DULSL's technical benefits include multivendor solutions, a built-in "survival of the fittest" culling mechanism, and the ability to transfer a license to the hardware required in the workflow.

Assuming that clients require DULSL from their software vendors, the latter will need to produce modular and connectable software components. Modularity is required so that only the client-requested capabilities can be provided instead of the entire application. A by-product of modularity would be well-defined communications and interaction between the software components which would, in turn, allow other software vendors to easily integrate their capabilities. In an ideal world, this would result in a greater multivendor suite of application functionality with the client being able to select the technology that is most applicable to their current problem.

Whether a single vendor or a multivendor solution, the client will control where software resources are dedicated by painlessly providing the software vendor with usage statistics regarding what is used and how much it is used. Those software capabilities that are not used by the majority of clients should get the least development attention and, therefore, client usage patterns will dictate where the software vendors focus their efforts.

DULSL allows the client to transfer a single license to the hardware platform best suited for their current problem. Whether working alone on a desktop, going on a business trip with a laptop, or giving a management presentation in the visualization theater, the client has full control concerning where (i.e., what hardware) the software is available simply by electing to transfer the license and then starting the software on the desired machine. Even if the software has never been run on the desired platform, since it is Web-deployed and multi-platform capable, the installation on the desired machine requires little or no IT resources. DULSL's biggest cost benefit is the ability to only license the software functionality required and only pay for it when it is used. Additionally, clients may benefit through:

- Focused workflow derived applications that increase productivity by avoiding the clutter of unwanted functionality.
- Reduced IT costs since deployment, installation, and support are now the software vendor's responsibility.
- Reduced costs and workflow cycle times because, instead of being constrained to the capabilities of currently licensed software, explorationists can integrate multivendor capabilities to solve problems.
- Reducing costs associated with transferring licenses between desired hardware platforms and the integration of diverse licensing schemes.

**Impediments.** The main impediment to DULSL will be the reluctance of traditional software vendors to only charge clients for the software that they find useful and only charge

them for it when it is used. This licensing approach will initially result in a reduction of revenues for those vendors, but over time will provide a "survival of the fittest" culling of unneeded, unwanted and therefore unlicensed software capabilities. This will reduce and refocus the vendor resources to the benefit of the client and, ultimately, the vendor's bottom line. Furthermore, since the user is required to periodically connect to the vendor server, vendors will be able to detect any attempts to circumvent the licensing structure or reverse engineer the application components. When irregularities are detected, licenses would be automatically disabled, thus reducing software piracy and again increasing software vendor revenues.

A second impediment to DULSL is the reluctance of clients to allow usage monitoring which is needed for the vendor to accurately invoice the client. On the surface this may seem insurmountable for those clients that have a problem with usage monitoring; however, there are a couple of viable solutions. Figure 1 shows three deployment options:

- The client connects directly to vendor server with no firewall protection.
- The client connects to vendor server with firewall protection.
- The client periodically connects to vendor server to update software and maintains a mirror server inside their firewall.

Implementation of the third option, which addresses client concerns regarding usage monitoring, requires the client to purchase and install a vendor server inside a firewall, and either contract the vendor or utilize newly-freed IT resources to maintain it. This configuration allows all licensing and usage information to remain inside the company's firewall. Under this option, the user would disconnect the internal server from an intranet and connect it to the vendor server periodically to download the latest software updates and optionally upload usage statistics. Once the client has verified the latest software is bug- and virus-free, they then disconnect from the vendor server and reconnect their server to the intranet. This deployment can be further adapted to eliminate the usage statistics upload entirely by taking a more traditional licensing approach based on a maximum number of hours of expected use. Under this approach, usage monitoring would be completely disabled and the client would only connect to the vendor server when software updates were available.

**Conclusions.** Implications of DULSL include:

- Only purchase functionality that is needed.
- Only pay for software access when it is being used.
- Install and transfer licenses between hardware platforms and users with a single button press.
- Utilize fully integrated multivendor workflows.
- Seamlessly integrate proprietary capabilities into multivendor workflows.
- Dynamically construct applications based on workflow requirements. [TJE](#)

*Acknowledgments: The author thanks Bonnie Barton and her family for their assistance in editing this article.*

*Corresponding author: Stuart A. Jackson 1-281-684-0576 or [stuart@s2ssystems.com](mailto:stuart@s2ssystems.com)*