Software Licensing And Royalty Rates

By Dwight Olson

I have been invited many times to talk about royalties and royalty rates in the software licensing industry at LES meeting workshops, but I have not commented on the various software types and license contexts of commercialized software technology that are associated with royalty rates. Royalty rates associated with software licensing are generally found in the various public databases in the cloud (such as EDGAR[1] at the Securities and Exchange Commission), as well as in surveys done by LES societies such as the LES (USA & Canada) royalty rate healthcare survey. In writing this article I drew on my past experience in software product commercialization, heading up the world's largest source code escrow service.[2] In that capacity, I had to read many hundreds of escrow agreements and their associated software licensing agreements. With the aid of ktMINE, I considered thousands of software agreements from their database of over 15,200 global royalty rate agreements. Of the ktMINE agreements that include software, the results were narrowed to 1,464 software agreements that were appropriate for this article. I reviewed nearly 10 percent of these results in detail. This research was conducted to answer the following question: Do contexts, industry sector or types of software impact royalty rates?

The following slides are from a 2012 LES workshop presentation of licensing and royalty rates. But what do these four slides really mean? Is it the industry that causes software royalty rates to vary?

■ Dwight Olson,
V3Data,
LESI IP Valuation Committee
Director and Past Chair,
San Diego, CA, USA
E-mail: dcolson@ix.netcom.com

1. Since 1934, the SEC has required disclosure in forms and documents. In 1984, EDGAR began collecting electronic documents.

2. Data Securities International held the source code associated with software licensing. See *Wikipedia*.

Regarding Software Investment

It is in the United States government's National Income and Product Accounts (NIPA) technical paper[3] where I first found any recognition of business and government expenditures (acquisitions) for computer software as investments. Prior to the year 2000, business expenditures for software were classified as inputs to production, and government expenditures for software were classified as government consumption expenditures. The NIPA paper also highlighted some other interesting points. From the NIPA paper:

> **"Three types of software are newly treated as investment. These are:**
>
> (1) Prepackaged software;
>
> (2) Custom software; and
>
> (3) Own-account software.
>
> *Prepackaged software is software intended for non-specialized uses and is sold or licensed in standardized form…Prepackaged software is available in a number of forms, ranging from pre-wrapped boxes to electronic files that may be downloaded from the Internet.…Custom software is software tailored to the specifications of a business enterprise or government unit. It may include new computer programs as well as programs incorporating preexisting or standardized modules…Own-account software consists of in-house expenditures for new or significantly-enhanced software created by business enterprises or government units for their own use.[4]*

Software Investment Growth

"Software investment by business and government increased rapidly from very small amounts in the late 1950's to $1 billion in 1966. It continued to grow rapidly, to more than $10 billion beginning in 1979,[5] and to $159 billion in 1998."[6]

Regarding Software Licenses

A software license is a legal instrument (usually by way of contract law, *with or without printed material*) governing the use of software. Software licenses can generally be fit into the following two (2) categories:

- proprietary and
- free/open source.[7]

A typical proprietary software license grants a user permission to use the software by the payment of a royalty fee.

Regarding Proprietary Software License Contexts

In my years of software product commercialization, I became familiar with five (5) primary contexts for proprietary software licenses. These are:

1. End-user license (licenses Individual, business, or enterprise [seat, company] for use of the software) using a standard license agreement for all licensees, typically in the form of a "shrink wrap" license.
2. Internet licenses (old shrink wrap, click through, cloud, *etc.*, and similar to end-user license for use of product).
3. Software patent or trademark license for the IP rights use.
4. VAR[8], OEM[9], Distributor[10] license for distributing the software in some bundled or other direct-controlled form.
5. Custom software license for use of newly coded or modified software in some form, which usually includes support or IP rights issues and can include 1–4 above for use. A typical custom agreement will deal with the unique issues between licensor and licensee and the "unique" own-developed software.

---

7. I discuss open source valuation in *les Nouvelle* June 2010, "Software Valuation of Open Source Software (OSS)—It's Not All Free Beer!"

8. Value Added Reseller—a company that adds features or services to an existing product, then resells it (usually to end-users) as an integrated product or complete "turn-key" solution. This practice occurs commonly in the electronics industry, where, for example, a VAR might bundle a software application with supplied hardware. See *Wikipedia*.

9. Original Equipment Manufacturer—a manufacturer of products or components that are purchased by another company and retailed under that purchasing company's brand name. While the term was used in the early 1960s and 70s in the U.S. to refer to value-added resellers,[3] OEM is currently defined by IBM to refer to "a manufacturer of equipment that may be marketed by another manufacturer." See *Wikipedia.*

10. Distributor, such Software Distributor (SD) such as the Hewlett-Packard company's name for their HP-UX software package management system. SD provides a set of tools for creating packages that will install software on a system running the HP-UX operating system. The packages can be grouped together into a software repository called a depot. Apple's iTunes for software applications is another example.

---

3. "Recognition of Business and Government Expenditures for Software as Investment: Methodology and Quantitative Impacts," 1959-98.

4. *Ibid.*

5. It was in 1976 that IBM unbundled software from hardware and I personally consider this to be the beginning of the software licensing revolution.

6. "Recognition of Business and Government Expenditures for Software as Investment: Methodology and Quantitative Impacts," 1959-98.

The three types of commercialized software associated with the license contexts that I am most familiar with are shown in the table below:

(1) Prepackaged software–1, 2, 4,

(2) Custom software–3, 5,

(3) Own-account software–5, if licensed out for other than internal use.

Software Type	Prepackaged	Custom	Own (out)
License Context			
1. End User	X		
2. Internet	X		
3. IP Only		X	
4. VAR, OEM Distributor	X		
5. Custom		X	X

The X's in the chart reflect where a license context makes sense regarding a software type. For example "Own software" is not licensed with a standard off-the-shelf prepackaged license agreement, but rather is licensed using a custom printed agreement. If there is no X, then the context and type together do not necessarily make sense, or I have not found any examples that matched them.

Regarding Software Intellectual Property

A software license will typically make reference to the software's Intellectual Property (IP) and IP rights embedded in or that travel with the software. *Under United States copyright law all software is copyright protected.* Typically software owners use IP protections to provide for the following, as shown in the table below:

1. Copyright–protects delivery content.

2. Trademark–protects brand.

3. Trade Secret/know-how, *etc.*–protects non-distributed components (*i.e.*, source code).

4. Patent–protects invention, methods, process, and design.

Software Components As Intellectual Property[11]				
Software Component	**Copyright**	**Trademarks**	**Patents**	**Trade Secrets**
GUIs	√	√		√
Source Code	√	√	√	√
Object Code	√			
Business Process	√	√	√	√
Data	√	√		√
Table Structures	√			√
Documentation Internal Design And External Use	√	√		√

Regarding Software Royalty Rate & Royalty Rate Assessment (RRA)

RRA is a practical tool to gauge the impact of a *royalty* fee commitment in a license on the business interests of the contracting parties. The terms 'royalty', 'royalty rate' and 'royalties' are generally used interchangeably. However, in this article I use royalty (royalties) to reference the "use" license subscription/transaction fee and royalty rate to reference the percentage of revenue paid by the licensee to the licensor (such as for the sublicensing and distribution of a licensed software product). Firms that license proprietary software products have spent large sums of money to develop software components (see above), and then from the components prepare the software product deliverables (executables and user documentation usually identified by a software product *trademark)* that are then licensed for revenue via royalties. In the case of sublicensing rights in the license, the royalty rate charged can be a percentage of the software product use fee and/or a royalty rate (percentage) of the total revenue generated (gross or net depending on the situation).

Market Size for Software Licensing (USA)

With information from the United States Bureau of Economic Analysis (BEA), I used the slide below in an LES Brazil 2011 Annual Meeting presentation[12] to provide a basis for understanding the size of the market for software licensing. Please note that the data is for the year 2000. However, as Carol Moylan stated in the BEA report, "BEA cannot use a demand approach to estimating software, primarily due to a lack of consistent source data." She went on to state that "(t)he Annual Capital Expenditure Survey (ACES) provides capital spending by industry."[13] According to this data, *businesses in 2012 accrued eighty seven (87) billion dollars on capitalizing software and spent/licensed over two hundred forty eight (248) billion dollars on software.*

Regarding Royalty Rates for Prepackaged Software

I have found the royalty rate for prepackaged (off-the-shelf,

---

11. Table is from "Software and Valuation," *les Nouvelles,* Dec. 2008, Dwight Olson, CLP, Denny Kolb.

12. 2012 Annual LES Brazil meeting. Workshop on software licensing pitfalls. Dwight Olson, CLP.

13. Estimation of Software in the U.S. National Accounts: New Developments by Carol Moylan Bureau of Economic Analysis U. S. Department of Commerce.

downloadable, *etc.*) software with contexts 1, 2, and 4 software components are typically set by competition in the application market, and are directly associated with the software product's use (execution of the application or software product to perform a task or process). The royalty rate may be multiplied by the number of software users at the enterprise (or limited to physical computer equipment at the enterprise). I have also seen royalties for a single user run from free[14] (owner has an open source business model rather than an ROI of licensing), to tens of dollars (such as Microsoft Word), to tens of hundreds of dollars (for a specific-purpose standard software application such as a data controller) to tens of thousands of dollars (for software that controls a paper cutter in the cardboard box industry).

For prepackaged software, it made sense to correlate royalties that are software application specific to a niche market within an industry, relying heavily on the features and functions of the software product to provide some rationale for variances of the royalties. In the VAR/OEM/Distribution license or sublicense rights license section of a custom agreement, many prepackaged software (sublicensing rights) licenses have royalty rates associated with the licensee's revenue for payment back to the licensor. I have found wide variances in these royalty rates, as might be expected, which could be due to unique services being provided or to the level of licensing experience of the licensor or licensee in the market niche. Further investigation should be made to determine if royalty rates associated with prepackaged software for use in different industries do correlate in OEM/VAR/distribution licenses.

Many licenses had tables with varying rates, where a smaller royalty rate was associated with a volume commitment. There may have been other business considerations that impacted the rate as well. Such business considerations could be for first line of support for the "end-user" in situations where the initial support for a sub-license is handled by the OEM, VAR or distributor. That is, the licensee is responsible for phone calls, emails, on-line support, *etc.* for the sub-licensee.

In my investigation for the OEM/VAR of prepackaged software, there were few licenses where the license permitted the VAR or OEM to modify the licensed software. This may be due to the desire of the licensor/owner to maintain some form of digital rights management (DRM) of the software, and to control the payment and collection of royalties. In these licenses where modifications were permitted, it appears as though a small set of source code was provided for modifications, and the modifications were limited to special fixed functions unique to the OEM/VAR equipment (I concluded this as the agreements contained source code escrow clauses). In other license agreements, it appears that the licensor made such modifications and provided a new version of the software product to the licensee for use or other revenue generation/extraction. These kinds of agreements usually had a significant up-front fee, which typically is used for commercializing the modified software components with ultimate delivery of a new version of the software product to the licensee.

To close my comments of prepackaged software, it is in my opinion that the software licenses for OEM and VAR are more closely aligned to the issues for the software integrator, or even the software distributor, than to the tangible hardware intricacies of the OEM or VAR. Why do I think this? Bill Elkington, an LES member, sheds some light:

> *"In product company after product company, the value of the product is in the software that is at its heart. In company after company, application-specific integrated circuits are being morphed into "prepackaged" software that runs on general purpose processors. In company after company, technology trends and market forces are forcing the product software to be abstracted from the company's product hardware. The hardware, for the most part, in embedded products is being commoditized. The differentiation is in the software.*[15]

Regarding Own-Account Software

In my software history, the type of software known as "own-account software" came into existence when an enterprise determined that it must computerize or automate some processes. For example, Bank of

---

14. "Software Valuation of Open Source Software (OSS)–It's Not All Free Beer," *les Nouvelles,* June 2010, Dwight Olson, CLP.

15. Bill Elkington, *Intellectual Asset Management,* May/June 2013 article.

America most likely developed its own on-line banking applications in order to be competitive and to provide its clients with immediate access to banking functions. In my history as an escrow agent that verified source code in escrow, I found most, if not all, complex systems software components had significant third-party software, as well as open software intermixed with its own developed software components. I believe most own-account software never made it to the commercial licensing market (apart from possible cloud transactional usage) due to the many proprietary, commercialization, and IP rights issues involved in delivering a "standard, off-the-shelf" software product. That is not to say that own-software applications are not licensed within the enterprise family, say to a subsidiary enterprise or alliance. However, such licenses' royalties are set internally by the enterprise or alliance, and are difficult to find or get access to.

In my analysis, I found varying royalties and royalty rates tied to use, distribution, digital rights management, training, support, and installation. My research indicates that some royalties were associated with an actual transaction to provide a service to a client in addition to the provision of the software. It would appear in today's environment of cloud computing that installable prepackaged software may be in turmoil. An example of this is the client-side software in the form of a mobile app that accesses a cloud service. I am not saying we are seeing the end of installable prepackaged software products (downloaded or loaded off a CD into computers and laptops), but rather software companies may be migrating applications to the cloud for "transactional use" royalty generation. For example, a software company here in San Diego that I am familiar with, in the "medical pathology market," does transactional-based royalty fee collection rather than pursuing an annual royalty licensing fee. Based upon discussions with the owner, I computed the corresponding average annual royalty rate to be close to 1 percent of the annual revenue for an independent pathologist.

Regarding Royalty Rates on Custom Software

In my opinion, custom software will always require custom licensing. A custom license is used to address all of the unique issues and situations that need to be addressed for the deliverable software components and/or software products to be used. Some unique situations include:

1) Patents or trademarks are to be licensed outside the scope of "internal use" with the included software product, and may be used in the case of a patent to build a different commercial product.

2) Proprietary know-how or trade secrets are to be licensed, such as the source code, CAD drawings, client databases, *etc*.

3) Installation, training, support issues can be quite complex. For example, an Internet download and install may not be possible due to the need to integrate a variety of third-party software products from different licensors at installation time.

4) On-going support and maintenance need to be determined on an *ad-hoc* basis. For example, problems in the software may require considerable modifications, as well a possible re-install of the software by the licensor.

Regarding Software Patent-Only Licenses and Royalty Rates

In my research I did not come across any licenses that contained only software patent rights usage. They all had associated background software, technology or know-how in the license. I found the typical grant language in a majority of the licenses that contained patent rights to contain this or similar language:

1. Grant the right to use licensed patents and licensed methods in the fields of use;

2. Grant the right to make and sell licensed products; and,

3. Grant the right to reproduce, prepare derivative works, distribute copies to the public, and display publicly the licensed software.[16]

In Gordon Smith's book, "Valuation of Intellectual Property and Intangible Assets, Third Edition," he discusses the concept that proprietary technology or trade secrets and know-how is often more valuable to an enterprise than its patents. He references Karl Jorda, David Rines Professor of Intellectual Property Law and Industrial Innovation at Franklin Pierce, who states, "Patents are but the tips of icebergs in a sea of trade secrets. Over 90 percent of all new technology is covered by trade secrets and over 80 percent of all license and transfer agreements cover proprietary know-how or hybrid agreements relating to patent and trade secrets." Much of those trade secrets are in software processes, products and programs, electronic databases maintained by software, and trade secrets embodied in software.[17]

Software Licensing and Royalty Rates– A Closer Look

I used ktMINE to extract 2,336 license agreements that contained references to software from their database of over 15,200 global royalty rate agreements.
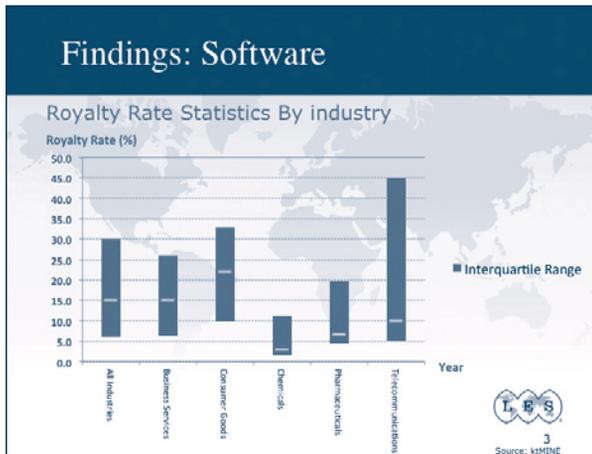
---

16. "In the license from the University" for any application, such grant to be effective on the date that The U.S. Department of Energy grants The Regents of the University of California's request for permission to assert copyright in that software.

17. "Leveraging Software via the Capital Markets," Dwight Olson and David Drews, *les Nouvelles*, September 2008.

Of these agreements that include software intangibles, I narrowed the results to 1,464 agreements that were appropriate for this article by excluding those that were part of joint development, cross license, franchise, or asset purchase agreements. I personally reviewed nearly 10 percent of the results in detail.

In the chart below, I condensed all software types into "Custom," as I found no "off-the-shelf" or "own" software. There were some that had similar language to these other types of agreements in some sections, but differed significantly in the sections related to sublicensing and distribution of the software. I found no software patent-only agreements. In order to provide additional insight into the above royalty rate charts, I have provided the following notes for each industry.

Findings: Software

Royalty Rate Statistics By industry

Royalty Rate (%)

■ Interquartile Range

Source: ktMINE

Types And Contexts

Industries	Note and RR%
Business Services	1. 2%-70%
Consumer Goods	2. 2%-60%
Chemicals	3. 1%-25%
Pharmaceuticals	4. 2%-20%
Telecommunication	5. 1%-80%

Notes

1–There were 939 agreements for business services extracted by ktMINE of which I reviewed 82. The 82 had royalty rates that ranged from 2 percent to 70 percent of revenue for sublicense, service and/or transaction fees. 26 of the 82 had established specific fees per transaction and not a royalty rate. One even had a $1,000,000 up front fee. I did not attempt to correlate the royalty rates or provide a mean as I could not determine any consistency in the agreements such as a royalty rate for a patent license.

2–There were 613 agreements for consumer goods extracted by ktMINE of which I reviewed 62. The 62 had royalty rates from 2 percent to 70 percent of revenue for sublicense, service and/or transaction fees. Of the 62, 12 agreements had only royalty transaction fees.

3–There were 22 agreements for chemicals extracted by ktMINE of which I reviewed 15. The 15 had royalty rates from 1 percent to 25 percent of revenue for sublicense, service or transaction fees.

4–There were 57 agreements for pharmaceuticals extracted by ktMINE of which I reviewed 13. The 13 had royalty rates from 2 percent to 20 percent of revenue for sublicense, service or transaction fees. There were upfront payments ranging from $750,000 to $2,500,000 that dealt with modifications to the software.

5–There were 205 agreements for telecommunications extracted by ktMINE of which I reviewed 29. The 29 had royalties from 1 percent to 80 percent of revenue for sublicense, service or transaction fees. There were upfront payments ranging from $350,000 to $3,000,000 for initial delivery and/or modifications to the software. One required a retainer fee of $30,000 per month plus royalties.

Conclusion

In my analysis, I could not determine that industry sector or types of software were drivers for determining a royalty rate; and I could not determine that the mean (yellow bar) was informative for software. However, I did find certain licensing contexts to be a primary driver for determining royalty rates. For example, agreements that contained provisions for sublicensing of the software, the royalty rate was as typical at 70 to 80 percent of the sublicensed revenue.

I believe a closer analysis could be done in the future to determine if other drivers exist. For example, if the software is to be used in a VAR or OEM situation, would such context impact a royalty rate? That is, would software co-mingled with other components such as hardware or software impact a royalty rate? Or would the license royalty of the software product be the primary determinant for the royalty rate?

I also extracted some "License Grant" language[18] examples to give some idea of the variety of software licenses I encountered. I have masked the identity of the software and parties involved:

1. Grant the right to use the Product (Software and Object Code versions of other Integrated Software programs provided by AAAA International, Ltd. which are used in conjunction with the Object Code versions of BBBB(TM), an interactive computer software and hardware for use in the storage and analysis of research data in a collaborative environment, including, without limitation, all modifications and enhancements thereto including BBBB(TM) BIOINFORMATICS, the Hardware, and the Documentation) at the Designated Location(s) for its internal Crop Science research and development purposes for the Term.

2. Grant the right to make, have made, use, sell, offer for sale and import products and to practice any method under the Intellectual Property ("Intellectual Property" refers to, collectively, all trademarks, trade secrets, know-how, proprietary information and other intellectual property of XXX Holdings, Inc. and YYY Technologies, Incorporated currently developed by XXX Holdings, Inc. and/or YYY Technologies, Incorporated, including, without limitation, in each instance, all specifications, engineering drawings, schematics, bills of materials, software source and object code and algorithms, wiring diagrams, test procedures, assembly drawings, artwork, and other documents or files related to the Technology) in the Markets...

3. Grant the right to use the Technology (XXXX technology, an end-to-end system of media management and collaboration tools), designed to be built into a variety of video applications and supports commercial media management functions from creation through distribution, which is protected by USA patents and incorporates the Object Code and Documentation for the Technology) and Intellectual Property Rights thereto and to convey, develop, market, license, and/or sell the Derivative Work which utilizes the Technology in the territory.

4. Grant the right to use the handwriting recognition technology to be used in Wireless Mobile Phone products developed by ZZZ Telecom Co. Ltd. itself and sold with ZZZ Telecom Co. Ltd.'s proprietary brand name "QQQQ."

5. Grant the right to reproduce, perform, display, transmit and distribute the Licensed Content (content and images and code) on the Licensee Site and Related Media (mobile sites, mobile applications (apps), widgets, gadgets, RSS feeds, e-mail newsletters and other content delivery media) intended solely for use by End Users in the Territory.

6. Grant the right to use CCCCS software, software used for the creation of custom graphs for scientific and engineering applications, in object code form only, to develop GGGGG Applications. ∎

---

18. From ktMINE,v which are compiled from SEC, SEDAR and other public sites.