

EFFECTIVE CONTRACTING IN SOFTWARE LICENSING

Elizabeth A. (Betsy) Kelly*

Computers have become an integral part of today's business environment. As businesses struggle to compete not only locally and internationally, but also on the world wide web, computers are being relied upon more and more for providing a competitive edge through speed, accuracy and delivery of services. Consequently, computer programs, or software, are being developed to address virtually every aspect of a business's activities.

A computer software supply transaction generally involves two parties, the supplier of the software and the user of the software. Ideally, the terms of the entire business transaction for the supply of software will be covered in a software license agreement. This requires that the parties contemplate not only the major commercial aspects of the transaction, but also the legal issues that impact on these commercial aspects. Achieving a commercially reasonable balance between the parties' rights and obligations requires effective contracting skills. Legal counsel familiar with the intellectual property rights and the commercial issues associated with the software license can contribute greatly to the process of effective contracting.

The benefit of effective contracting in software licensing from a user's perspective, is that it ensures that a user will receive sufficiently appropriate computer software installed and functioning in accordance with the user's needs, and will also anticipate the potential problems, as well as prescribe a mechanism for resolving such potential problems, in the event that the software doesn't function accordingly. For a supplier, effective contracting prescribes clear parameters for its obligations to its customer, the user. If a contract is drafted with commercially reasonable terms, then there is a greater chance that each of the parties will be able to live up to their obligations under the contract and enjoy a successful, mutually beneficial business relationship. This paper will address, in practical terms, the legal issues that impact upon the software supply business transaction.

Product Description

Fundamental to the software license is the description of the software to be supplied to

* Associate lawyer with Patterson Palmer Hunt Murphy, Saint John, New Brunswick specializing in I.T. and commerce law.

the user. This software may have already been developed by the supplier and has been perhaps licensed to other users or, it may have been customized specifically for the user by the supplier. In some instances, a supplier will develop a new software product especially for the user which often necessitates negotiating a different contract, namely a software development agreement. For the purposes of this paper, I will contemplate a simple software license agreement where the supplier is not required to make any customizations to its software at all and where it provides a set of specifications setting out the functionality of the software as part of the product description. Such specifications can provide a measurable set of criteria upon which acceptance testing may be based. Additionally, a supplier may also prescribe minimal hardware and software components necessary for the operation of its software in accordance with its product description but which is purchased or licensed by the user independently of the supplier's software.

Grant of License

Generally, a supplier will retain legal title to its software and grant to the user merely a right to use the software, often referred to as a "grant of license". A supplier will be inclined to give a user a restrictive grant of license thus retaining more rights to itself so that it can freely market the software to other users. A user is well advised to review the grant of license to ensure that it has a sufficiently broad scope of rights to allow it to use the software in a meaningful way in accordance with its objectives. In instances where software has been customized for a user, a user may be able to negotiate an exclusive right to use the software, giving it a competitive advantage in the marketplace. Otherwise, a supplier will generally grant to the user a non-exclusive license which may be limited to a use within a specified territory, thus retaining the right to license the software to other users.

Third Party License Restrictions

Some suppliers of software incorporate third party software into the software platform which they are licensing. The licenses between the third party(s) and the supplier are sometimes referred to as "upstream licenses". A supplier is well advised to review its upstream licenses to ensure that it has the right to incorporate the various upstream software applications into its own software together with the corresponding right to sublicense to the user. Often such upstream licenses which do allow sublicensing will prescribe minimum terms and conditions by which an end user must abide. Such minimum terms and conditions generally include restriction on use, a prohibition preventing title to the upstream software from passing to other parties, a disclaimer of any liability for damages arising from the use of the upstream software, as well as a prohibition against any reverse engineering or decompilation of the software. In some instances, an upstream licensor may require the supplier to stipulate that it is a third

party beneficiary under the sublicense agreement. By so doing, the upstream licensor gains the benefit of privity of contract allowing it to seek a remedy against an infringing user where it has infringed the intellectual property rights of the upstream licensor.

Payment and Successful Installation

Payment for a grant of license for use of software may be by lump sum payment, royalty, or both. A prudent user will ensure that payment is tied to successful installation of the software so that it operates in accordance with the product description. Acceptance criteria will set the parameters of what constitutes successful installation and operation. In many instances a payment scheme will contemplate a lump sum payment to provide some security for the supplier upon signing, with the balance of the payment tied to a series of successful acceptance tests conducted during key installation points.

Part of the installation and implementation of the software may involve training the user's employees to use the software in an effective manner. A well drafted contract will set out the terms of the training, the scope of the training, as well as training location and key personnel to be trained.

Maintenance and Technical Support

Many software platforms require ongoing maintenance and technical support to ensure the continuing functionality of the software because minor defects, and even major defects, may arise which impair its functionality and ultimately impacting on the user's business. In recognition of this practical reality with software, many users secure ongoing maintenance contracts with either the supplier or perhaps an independent third party maintenance provider. Either way, the user is wise to ensure that the software license provides for maintenance one way or the other. Maintenance and technical support services generally require a right of entry onto the user's premises and, in some instances, also a right to gain access to the user's systems remotely by a modem. A supplier or a maintenance provider will want to ensure that the user designates certain key employees as contact people and that such contact people are adequately trained and familiar with the software such that they are able to diagnose problems as being either major or minor, and also that they are able to work with the supplier or maintenance provider so that such problems may be pinpointed and fixed readily implemented.

Generally, a well drafted maintenance contract will set out parameters defining the priority of such errors as well as prescribe what efforts will be made to rectify the errors in accordance with a set time table. It is prudent for a supplier or maintenance provider to ensure that the obligation to provide services relates only to the software supplied and

that, where the error actually lies elsewhere (*i.e.*, in the user's other software or with its hardware) then the supplier has a right to either discontinue providing the particular support service or perhaps a right of repayment for its time and effort to remedy a problem which really was not associated with its software.

Source Code Escrow

Where maintenance and technical support services are necessary for the ongoing successful operation of the software, a user is wise to ensure that it has the benefit of a source code escrow agreement. Such an agreement requires the supplier to deposit the source code of the software with an independent third party escrow agent. The "source code" is the human readable version of the software, as opposed to the machine readable version that is licensed to the user, known as the "object code". The escrow agent holds the source code in trust for the benefit of the user in the event that the supplier has breached a fundamental term of the contract and /or has failed to provide maintenance services necessary for the ongoing operation of the software. Upon the happening of such an event, the escrow agent will release the source code to the user upon the terms and conditions set out in the escrow agreement. These give the user the limited right to use the source code for the purpose of carrying on the necessary maintenance services to keep the software operational. Common triggering events for the release of the source code are insolvency of the supplier, material breach of the terms of the license, or some other event which would have the effect of denying the user the necessary maintenance services.

Warranties and Indemnities

A prudent user will require a supplier to warrant that it has the full rights and powers to enter into the license agreement with the user. This requires the supplier to warrant that, not only does it have all necessary rights pursuant to its upstream licenses, but also that it has all necessary rights with respect to its own software which it has incorporated into the licensed platform. In order for a supplier to give such a warranty, it must ensure that it has obtained all necessary intellectual property rights from its upstream licensors, its employees and any third party contractors which may have contributed to the development of the software. As a general principal, employees are deemed to have developed the software for the benefit of their employer. However, a supplier is recommended to ensure that its employees have signed employment contracts which clearly specify that all work developed belongs to the employer/ supplier.

Difficulties arise, and ownership rights become blurred, when employees come from competing companies and apply knowledge of their former employer's product to the development of their new employer's product, or where employees develop software products on their own time independent of their employment with the supplier.

With respect to third party contractors who are not employed by the supplier, but rather have contracted their services to the supplier, a development contract is essential to ensure that the supplier obtains the necessary intellectual property rights to allow it to market the software and accordingly enter into license agreements with its users. Notwithstanding such due diligence measures, a supplier may want to limit the warranty it gives by qualifying it "to be to the best of its knowledge and belief", in recognition of the fact that it doesn't have full control over its employee's or contractor's actions.

A prudent user will also require that the supplier warrant that the software is Year 2000 compliant, that is, that it will continue to operate beyond the year 2000. Since "compliant" does not have any legal meaning, nor any practical meaning within the realm of software licensing, a supplier is recommended to warrant that its system will comply with a specified standard, such as the DISC PD 2000-1 standard published by the British Standards Institution.

In contracts of this nature, it is most commercially reasonable for the party best able to control the risk to accept the risk. This is most apparent when dealing with the indemnity issue where software supplied could infringe the intellectual property rights of another party. Generally, the supplier is in the best position to provide such an indemnity where it has control over the content of the software, either through its upstream licenses or through its own development efforts. The scope of an intellectual property indemnity may be very broad, and thus favour a user, or may be very limited, and thus favour a supplier. An example of a limited indemnity would be one where the supplier only agrees to defend and indemnify the user against claims that the software infringes the copyright of another party within the territory or jurisdiction in which the user is operating the software. By limiting the indemnity to only copyright infringements, the supplier is best able to control the scope of its liability because copyright infringement requires the allegor to prove, not only that the software is significantly similar to the infringed software, but that the supplier actually copied the infringed software. This is very difficult to prove and therefore many software suppliers are comfortable with giving copyright indemnities.

However, where the software may incorporate patentable subject matter, a user will want to ensure that the supplier will also indemnify it against any claims for patent infringement. Such claims may arise when a third party alleges that the user has infringed its patent by virtue of using the software supplied by the supplier. Patent infringement claims are more difficult to control because, unlike copyright infringement claims, independent creation is not a defense. That is, a supplier may independently develop its own software and perhaps even file for a patent, however, it is possible that a third party may have also created sufficiently similar software for which it may also have filed a patent application in advance of the supplier's. Because of the delayed publication process within the patent application process, a supplier may not know of a third party patent which could defeat its own in the face of an infringement claim.

Because it is very difficult for a supplier to control the risk of patent infringement, it is often not prudent for a supplier to provide patent indemnities.

Also before providing a patent indemnity, a supplier must review its upstream licenses to determine whether or not it has been provided indemnities for either patent, copyright, or other intellectual property rights infringements. It would be especially imprudent for a supplier to provide a patent indemnity where an upstream license only has a limited copyright indemnity because it denies the supplier recourse against its upstream licensor for damages suffered as a result of a patent infringement which may be tied to the upstream software.

Just as a user will want to ensure that it has adequate coverage from a supplier for infringement claims, a supplier may also require a user to indemnify it against any infringement claims arising out of the user's use of the software. For instance, a supplier may be subject to an infringement claim where its software is used in combination with the user's other software which infringes a third party's intellectual property rights. A commercially reasonable software license will ensure that in such an event, the user will indemnify the supplier for infringement claims where use of the supplier's software alone would not have resulted in the infringement claim.

Limitation of Liability

Just as warranty and indemnity clauses have become a standard component of many software licenses, so have limitation of liabilities clauses. A prudent supplier will ensure that the license agreement limits its liability to a prescribed amount or to the fees paid under the agreement. Liability for indirect and inconsequential damages, such as loss of profits and data, whether incurred by the user or some other third party, are often excluded. For a user to insist upon unlimited liability would be commercially unreasonable. However, it may not be commercially unreasonable for a buyer to insist that the limitation of liability terms be made mutual for each party, depending upon the nature of the software and its intended use.

Confidentiality

Because the negotiation of a software license can be very detailed and protracted, and may require each party to divulge confidential information, it is often prudent for the parties to execute a mutual non-disclosure agreement. This protects and preserves each party's rights to its confidential information. Confidential information disclosed by a user may include its business plans and operating systems, while a supplier's confidential information may include its software's functional specifications, and perhaps its development strategies for future releases of the software. Generally, non-disclosure agreements prescribe a broad definition for each party's confidential

information and then go on to exclude from that broad definition information which is already in the public domain, information already known by the other party, information which was released to the public by a third party, or information which is disclosed pursuant to a court order. This helps to get around the difficulty associated with actually attaching a definition of each party's confidential information as such a definition can sometimes disclose more information than what the party wishes to disclose. Non-disclosure agreements can also operate to protect a party to whom information has been disclosed by setting out parameters for use of the confidential information thus providing some certainty. Generally, the confidentiality requirement is limited to a certain period of time, such as two years, five years, or even ten years (which in the case of software is like infinity given its rapid rate of change).

Governing Law and Arbitration

The law governing the transaction embodied in the software license must be specified. This is particularly important where the parties conduct their respective businesses in different jurisdictions. Because the court process is reputed to be slow, many software licenses are drafted to require the parties to submit their disputes to arbitration where the arbitrator, or at least one of a panel of arbitrators, has some expertise in the area of software licensing. An additional benefit of arbitration is that, unlike court proceedings, it can be confidential. However, the requirement to submit to arbitration in a software license should not deprive the other party of seeking injunctive relief through the courts and this should be specified in the license.

Conclusion

A well drafted software license will encompass the entire transaction in writing for the supply of software and any related services. Both parties should ensure that an adequate period of time is allotted to allow the user to review any standard form contract which a supplier may provide and/or for the parties to negotiate an equitable and commercially reasonable set of terms to be embodied in the software license. This process can be facilitated by legal counsel who are familiar with the intellectual property issues and the commercial issues associated with software transactions. Counsel can provide an assessment of the risks associated with certain terms and conditions, while also providing creative, workable solutions where the proposed terms are not acceptable for a party. This allows a deal to happen, not break down. Understanding the risks and knowing the product and corresponding services to be delivered provides the parties with a greater chance of successfully implementing their business transaction. This makes the time and effort spent on negotiations well worth the investment.