

THE SOURCE CODE ESCROW: A WORTHWHILE OR WORTHLESS INVESTMENT?

by
Walter D. Denson*

INTRODUCTION

Gameco Manufacturing Inc. (“Gameco”), a fast-growing home video game manufacturing company, has begun to expand its operations and requires a new computer system to keep pace with its expansive data processing needs. Gil Bates is the director of Gameco’s technology department. While attending a trade show in Chicago, Bates witnessed the demonstration of a new software technology developed by Softco Solutions (“Softco”). Though Softco has only been in business for two years, it is beginning to build a very strong reputation in the technology community for its efficient and innovative software products. Bates thinks this would be ideal for Gameco as it would meet all of its data processing needs, as well as increase efficiency and cut cost.

Negotiations between Gameco and Softco take place, and subsequently the two companies enter into a licensing agreement. Typical of most software licensing agreements, Softco delivers only the software’s object code and withholds the source code, the blueprint of the software. Softco is extremely reluctant to divulge the source code, as most software vendors are, for fear that its trade secrets will be misappropriated by the licensee or a third party. Such misappropriation will allow these other parties to copy the program and thereby benefit from the vendor’s innovations without having to incur any of the development costs of the vendor. Without access to the source code, however, Gameco is dependent upon Softco to correct any bugs or defects in the software, as well as perform any necessary maintenance. Consequently, Gameco, realizing the vulnerability of this position, requires Softco to enter into a “maintenance” agreement¹ wherein Softco agrees to update, debug, and maintain the software for the term of the contract.

Gameco is very enthusiastic. The demos were very impressive, and all of the bugs have apparently been eliminated. The software is rolled out company-wide, and over 10,000 employees have been trained on the system. Although the total cost of the software package is over \$2 million, Gameco is ecstatic when thinking about the competitive advantage it will gain in the market as a result of the time and money it will save through the use of the new technology.

* Second year student at Rutgers School of Law – Newark. Bachelor of Arts degree in economics from Syracuse University.

¹ Typically, the costs of most maintenance agreements are annual, and are based upon a percentage of the license fees (varying from 10% to 50%). It would be prudent for a licensee to negotiate for the maintenance agreement to last for the expected life of the software. See Peter Vogel, *System Acquisition: Protecting the User*, 547 PLI/PAT 1083, 1085 (1998).

One year later, bad news strikes – Softco is in severe financial turmoil and files for Chapter 11. Gameco’s counsel advises that the Bankruptcy Code protects Softco such that Gameco will be unable to compel performance of the maintenance agreement post petition. Gameco is panicked because all of its critical functions rely on the software. If the software crashes, or becomes outdated and unusable, without access to the source code, Gameco’s business could be seriously imperiled.

This hypothetical is an example of how a software vendor’s bankruptcy can substantially threaten a licensee’s business. Unfortunately, this is not an uncommon situation. In fact, this hypothetical is somewhat analogous to an all-too-real situation involving Amoco Oil Corporation.² The oil giant had acquired a recently developed software technology from a largely unproven software developer. Shortly after entering into the licensing agreement, the vendor went bankrupt. Amoco managed to avert disaster because it was prepared. Amoco was very impressed by the vendor’s software, however, it was leery of the fact that the vendor had only been in business for a short period of time. Therefore, as part of the licensing agreement, Amoco negotiated an agreement, wherein the vendor agreed to place its source code in a “source code escrow.” As stipulated in the agreement, the vendor’s bankruptcy was one of the conditions triggering the release of the source code. As such, Amoco was able to obtain the source code and maintain its software in-house.

The source code escrow is a species that evolved in an effort to strike a balance between the vendor’s interest in protecting the trade secret of its software, the source code, and a licensee’s interest in obtaining the source code should a vendor become defunct or bankrupt.³ Under this arrangement, the vendor typically deposits its source code with a third party, an escrow agent. The escrow agreement dictates that the vendor is required to update and maintain the source code on a periodic basis. The escrow agent is required to maintain the escrowed materials safely and confidentially, only surrendering the materials to the licensee upon the occurrence of a specified release event.⁴ The two most prevalent release events occur when: (1) A vendor ceases to support the code; or (2) A vendor files for bankruptcy.

In recent times, however, critics have questioned the utility of the source code escrow.⁵ These critics offer a number reasons why the source code escrow is a futile exercise in protecting a licensee from vendor bankruptcy, and warn of the repercussions of engaging in this type of practice. In this note, I will discuss some of the arguments both for and against the source code escrow. I will then offer some suggestions a licensee may wish to consider in order to ensure that the source code escrow provides adequate protection against vendor bankruptcy.

² See Technology Escrow Releases (Amoco) at http://www.ironmountain.com/resources/resource.asp?svc1_code=8&resource_key=379

³ See generally Jonathan Mezrich, *Source Code Escrow: An Exercise in Futility?*, 5 MARQ. INTELL. PROP. L. REV., 117-19 (2001).

⁴ See *id.*

⁵ See *id.* at 119-20.

WHAT ARE SOURCE AND OBJECT CODE?

The technical side of the computer world has a language all unto itself, replete with lingo and jargon that at times leave many of us lay people scratching our heads in bewilderment. One judge remarked during a breach of computer contract case that:

[I]n the computer age, lawyers and courts need no longer feel ashamed or even sensitive about the charge, often made, that they confuse the issue by resort to legal ‘jargon’, law Latin or Norman French. By comparison, the misnomers and industrial shorthand of the computer world make the most esoteric legal writing seem as clear and lucid as the Ten Commandments or the Gettysburg Address.⁶

Therefore, as an initial matter, I think it necessary to briefly describe what are source and object code since these terms are at the heart of the issue.

When a software developer creates a new program, they do so using an alphanumeric, human readable programming language such as FORTRAN, PASCAL, Visual Basic, C, and C++.⁷ These programming languages are what are commonly known as the source code.⁸ The source code appears to the untrained eye as text,⁹ and describes in great detail the logic flow of a software program, which often includes the programmer’s narrative explanation of the various steps in the program.¹⁰ Often referred to as “high level language,” the source code “uses English words and symbols, and is relatively easy to learn and understand (e.g., ‘GO TO 40’ tells the computer to skip intervening steps and go to the step at line 40).”¹¹ One may think of source code as being somewhat analogous to a songwriter’s sheet music or score. The sheet music is the blueprint of the music, which includes the note arrangements and the songwriter’s directions enabling musicians to perform the work correctly. With the sheet music, a competent musician can copy and manipulate various parts of the music to develop a new, yet similar song. Similarly, a savvy programmer can manipulate and copy certain aspects of the source code to develop another software product, which would likely be in direct competition with the original program.

⁶ District Judge Edenfield presiding in the case of *Honeywell, Inc. v. Lithonia Lighting, Inc.*, 317 F.Supp 406, 408 (N.D. Ga. 1970).

⁷ See *Johnson Controls Inc. v. Phoenix Control Sys., Inc.* 886 F.2d 1173, 1175 and n.2 (9th Cir. 1989).

⁸ See *id.*; see generally *SAS Inst., Inc. v. S&H Computer Sys., Inc.*, 605 F.Supp. 816, 818 (M.D. Tenn. 1985)(defining source code as “the series of instructions to the computer for carrying out the various tasks which are performed by the program, expressed in a programming language which is easily comprehensible to appropriately trained human beings”).

⁹ See Andy Johnson-Laird, *The Discovery of Computer Software in Patent Litigation*, SC 16 ALI-ABA 209, 213 (1997).

¹⁰ David M. Carrick and Francis J. St. Hilaire, *Source Code Escrow*, Akins, MacAulay & Thorvaldson at <http://www.aikins.com/newsletters/fall96/8.htm> (Fall 1996).

¹¹ See *Apple Computer, Inc., v. Franklin Computer Corp.*, 714 F.2d 1240, 1243 (3d Cir. 1983).

Computers do not read or act directly on source code.¹² As such, it is necessary to translate source code into a language that the computer can read and understand.¹³ This translation is accomplished by using a computer program called an “assembler” or “compiler” program.¹⁴

The result of the translation of the source code to machine readable code is called the object code.¹⁵ Often referred to as “machine language,” the object code is “a binary language using two symbols, 0 and 1, to indicate an open or closed switch (e.g. ‘01101001’ means, to the Apple [computer], add two numbers and save the result).”¹⁶ These two numbers are arranged in various combinations which specifically represent the alphanumeric characters of the source code.¹⁷ The central processing unit (CPU) is only able to comprehend and follow this type of instruction.¹⁸

The object code is “devoid of much of the information [contained] in the original source code.”¹⁹ All of the programmer’s comments have been removed,²⁰ thus preventing would-be copiers from being able to read the programmer’s mind. In addition, “[t]he [symbolic constant and variable names] are normally removed from production of object code that is sent out into the market.”²¹ As such, there is very little comprehensible material present in the object code that will allow for reverse engineering²² of the software. It is for these reasons that the object code, as opposed to the source code, is delivered to software licensees.

THE SOURCE CODE ESCROW ARRANGEMENT

¹² See *Midway Mfg. Co. v. Strohon*, 564 F.Supp 741, 750 (N.D. Ill. 1983).

¹³ See *id.*

¹⁴ See *Apple Computer, Inc.*, 714 F.2d at 1243; see also *Sega Enter. Ltd. v. Accolade, Inc.*, 977 F.2d 1510 n.2 (9th Cir. 1992).

¹⁵ See *Carrick and St. Hilaire*, *supra* note 4.

¹⁶ See *Apple Computer, Inc.*, 714 F.2d at 1243.

¹⁷ See *Sega Enter. Ltd.*, 977 F.2d at n.2

¹⁸ See *id.*

¹⁹ See *Johnson-Laird*, *supra* note 3 at 218.

²⁰ See *id.*

²¹ See *id.*

²² Reverse engineering is the process by which the object code is translated into source code. This process is accomplished by the use of a “disassembler” or “decompiler” program. These programs read the electronic signals for “0” and “1” that are constantly produced while the program is being run, and stores the resulting object code in the computer’s memory, and translating this object code into source code. The disassembler and decompiler programs are commercially available and are widely used in the software industry. See *Sega Enter. Ltd.*, 977 F.2d at n.2, see also *Sony Computer Enter., Inc. v. Connectix*, 203 F.3d 596, 600 (9th Cir. 2000)(detailing the reverse engineering process).

Most source code escrow agency companies advertise via the Internet²³ and in technology trade journals. One commentator estimates that approximately “80% of all Fortune 1,000 firms have at least one software package on deposit with an escrow agent.”²⁴ Under the typical software escrow arrangement, the escrow serves as an independent repository for a vendor’s source code and any related documentation for a time generally equal to the term of the underlying software license.²⁵ The vendor is generally required to make periodic deposits of updated, enhanced, and modified editions of the source code, or other variants of the licensed software.²⁶ While providing an archival storage place for source code, the escrow also provides protection for the vendor’s trade secrets.²⁷ The escrow agent is prohibited from divulging the trade secrets of the source code “until it receives notice of an event requiring release, or until it returns the source code to the vendor upon the termination of the escrow.”²⁸ In order to obtain the source code, a licensee must notify the escrow agent and the vendor that an event requiring the release of the source code has taken place.²⁹ Such an event might be the inability or refusal of the vendor to maintain or update the software,³⁰ or the insolvency of the vendor.³¹ When an escrow agent receives such notice, “the agent is required to notify the vendor that it will release the source code to the licensee unless the vendor objects in writing within a specified period of time.”³² If the vendor objects, “the escrow agent is directed to hold the source code until ordered to release it by a court or by the vendor.”³³ However, if the vendor does not object, “the escrow agent is required to release a copy of the source code to the requesting licensee, retaining a copy for itself and for other participating

²³ See DSI Technology Escrow Services at <http://www.dsiescrow.com>; EscrowTech at <http://www.escrowtech.com>; The Source Code Escrow Company at <http://www.source-code-escrow.com>; Guard-IT Corporation Escrow Services at <http://www.guard-it.com>; Lincoln-Parry SoftEscrow at <http://www.softescrow.com>; Software Escrow Corporation at <http://www.softwareescrowcorp.com>; Fidex Americas Corporation at <http://www.fidex.com>; Desaware at <http://www.desaware.com/SourceCodeEscrowL3.htm>; The Archive Source Code Escrow at <http://thearchive.com/copy/archive.html>; InnovaSafe, Inc. Technology Protection Services at <http://www.innovasafe.com>; Zurich Depository Corporation at <http://www.zurichdepository.com/4.htm>; Trexcel Corporation at <http://www.trexcel.com>.

There are also companies that draft software source code escrow agreements which one may download any of these agreements from the Internet. See e.g. LegalPoint at <http://www.smartagreements.com/cgi-local/SoftCart.exe/tec/lp51.htm>.

²⁴ See Gary H. Anthes, *The Dangers Behind Software Escrow*, Computerworld (Dec. 21, 1998) at http://www.computerworld.com/cwi/story/0,1199,NAV47_ST033324,00.html.

²⁵ See Thomas M.S. Hemnes and Susan Barbieri Montgomery, *The Bankruptcy Code, The Copyright Act, and Transactions in Computer Software*, PLI order No. G4-3818, July 21, 1988, available at WL 255 PLI/PAT 657, at *686.

²⁶ See *id.* at n.4 (“Exactly what further deposits are required depends on the terms of the underlying license. If, for example, the license does not cover further editions of the licensed programs, then the source code for such further editions would not need to be covered by the escrow agreement”).

²⁷ See Edwin H. Taylor, *Protecting Computer Trade Secrets*, PLI Order No. G4-3790, May 1, 1986, available at WL 224 PLI/PAT 797, at *804.

²⁸ See Hemnes and Montgomery, *supra* note 16.

²⁹ See *Id.* at 686-87.

³⁰ See *Id.* at 687.

³¹ See Taylor, *supra* note 17.

³² See Hemnes and Montgomery, *supra* note 15 at 687.

³³ See *Id.*

licensees.”³⁴ Some escrow agreements provide for arbitration when disputes arise over whether an event requiring release has taken place.³⁵

Once the source code is released to a licensee, generally, the escrow agreement provides the licensee with the “rights and obligations respecting the source code similar to those which apply to the object code.”³⁶ These rights include the licensee’s right to “use the source code to maintain licensed programs, subject to the obligations not to disclose the source code to third persons and to return or destroy the source code upon the termination of the license.”³⁷

To avoid possible lengthy and expensive litigation, the escrow agreement should be meticulously drafted to effectively notify the parties when a release event occurs.³⁸

IS THE SOURCE CODE ESCROW A FUTILE INVESTMENT?

One commentator argues that the “path to salvation via [source code] escrow is so full of pitfalls that ... many companies enjoy little real protection.”³⁹ Also, some may feel that the source code escrow enjoys widespread success only because escrow agency companies play on the fears of businesses who rely on vendors for complex and unique applications.⁴⁰

Specifically, it is argued that the utility of source code escrows is dubious when considering: (1) The adequate protections already afforded by Section 365(n) of the Code; (2) The problem of a “learning curve” should a licensee obtain the source code; (3) The infrequent releases of escrow materials; and (4) The possibility of an obstinate vendor who refuses to approve the release of the source code to the licensee.⁴¹

HAVE THE 1988 AMENDMENTS TO SECTION 365 OBVIATED THE NEED FOR SOURCE CODE ESCROWS?

The utility of the source code escrow as a source of protection against vendor/depositor bankruptcy has always been questioned, even prior to the 1988 amendments. It was during that period when trustees and debtors-in-possession began determining that licensing and escrow agreements were executory contracts, and unilaterally rejecting those contracts pursuant to Section 365(a) of the Bankruptcy Code (“Code”).⁴²

³⁴ *See Id.*

³⁵ *See Id.*

³⁶ *See Id.*

³⁷ *See Id.*

³⁸ *See Taylor, supra* note 18 at 805.

³⁹ *See Anthes, supra* note 24.

⁴⁰ One company states on its website that “Without a technology escrow agreement, a licensee’s investment in technology is unprotected. Recovering the source code of that technology by other means, such as through the court system, can take years – and by then it’s too late to save the company’s operations and resources that were tied to that technology”; *See* www.dsiescrow.com/techescrow/index.html.

⁴¹ *See Mezrich, supra* note 3 at 120.

⁴² A practice that largely began following the controversial decision reached in *Lubrizol Enter. v. Richmond Metal Finishers, Inc.*, 756 F.2d 1043 (4th Cir. 1985), *cert. denied*, 475 U.S. 1057, 106 S.Ct. 1285, 89 L. Ed. 2d. 592 (1986), discussed *infra* at note 50.

The Executory Contract Dilemma for Licensees

Section 365(a) of the Code governs the disposition of executory contracts.⁴³ This section was designed and it was Congress' intent to allow a debtor to maximize its assets by minimizing its burdensome liabilities.⁴⁴ As such, a debtor-in- possession or trustee may elect to either retain or reject an executory contract, in an effort to maximize the bankrupt's assets.⁴⁵

Neither the 1978 Bankruptcy Code nor its predecessor define what is an executory contract.⁴⁶ Congress recognized the difficulty in adopting a hard and fast definition of an executory contract.⁴⁷ However, Congress stated that the definition "generally includes contracts on which performance remains due to some extent on both sides."⁴⁸ Professor Vern Countryman, a very influential individual in this area, defined an executory contract as "a contract under which the obligation of both the bankrupt and the other party to the contract are so far unperformed that the failure of either to complete performance would constitute a material breach excusing the performance of the other."⁴⁹ The Supreme Court has found contracts to be "executory" where "performance remains due to some extent on both sides."⁵⁰

The concept of rejection is fairly simple. Rejection in this context is similar to an anticipatory breach of a contract, where the debtor notifies the other contracting party and the bankruptcy court that it will not continue to perform its contractual obligations.⁵¹ What is significant about rejection is not the liability that may be imposed upon a debtor, but rather the

⁴³ See 11 U.S.C. § 365 (2001). Subsection (a) provides "Except as provided in sections 765 and 766 of this title and in subsections (b), (c), and (d) of this section, the trustee, subject to the court's approval, may assume or reject any executory contract or unexpired lease of the debtor.

⁴⁴ See John P. Musone, *Crystallizing The Intellectual Property Licenses In Bankruptcy Act: A Proposed Solution To Achieve Congress' Intent*, 13 EMORY BANKR. DEV. J. 509, n.3 (1997)(stating that Congress' intent in enacting section 365 "is to relieve the debtor from burdensome liabilities and to allow it to take the fullest advantage of its assets. This is accomplished by providing the debtor with an option to convert otherwise out of pocket expenses into damage claims").

⁴⁵ See 11 U.S.C. § 365(a).

⁴⁶ See David S. Kupetz, *Intellectual Property Issues In Chapter 11 Bankruptcy Reorganization Cases*, 9 J. of BANKR. and L. PRAC. 425, 426 (2000).

⁴⁷ In the legislative history, Congress recognized that there "is no precise definition of what contracts are executory." See H.R Rep. No. 95-595, at 347 (1977).

⁴⁸ See *id.*

⁴⁹ See Michael T. Andrew, *Executory Contracts In Bankruptcy: Understanding 'Rejection'*, 59 U. COLO. L. REV. 845, 849 (1988)(quoting Professor Vern Countryman's definition of an 'executory contract' that has "come to be viewed by many as embodying, in a simple formula, the entirety of fundamental executory contracts doctrine").

⁵⁰ See *N.L.R.B. v. Bildisco and Bildisco*, 465 U.S. 513, 523, 104 S. Ct. 1188, 1194 n.6 (1984).

⁵¹ See Daniel T. Brooks, *Intellectual Property Bankruptcy Protection Act of 1988*, 272 PLI/PAT 575, 607 (1988).

limits it places on the other contracting party's damage remedies.⁵² Unless the other contracting party perfects a security interest in the executory contract, that party will only have an unsecured claim against the debtor's estate.⁵³ In fact, where an executory contract is converted into an unsecured claim, the other contracting party may not have an effective remedy at law.⁵⁴

Despite the power to reject executory contracts, the trustee or debtor-in-possession does not enjoy unfettered discretion to do so.⁵⁵ They must show an exercise of sound "business judgment"⁵⁶ to gain approval from the bankruptcy court to reject an executory contract.⁵⁷ The court may deny a trustee's or debtor-in-possession's request to reject an agreement where these parties have shown "bad faith or gross abuse of discretion."⁵⁸ Bad faith or gross abuse is generally demonstrated where the rejection will inhere little or no benefit to the debtor's estate while causing devastating effects upon the business of the other contract party.⁵⁹

The *Lubrizol* Controversy

The power to reject intellectual property licenses was assumed by trustees as a result of the controversial decision reached by the Court of Appeals for the Fourth Circuit in *Lubrizol Enterprises v. Richmond Metal Finishers, Inc.*⁶⁰ In *Lubrizol*, the defendant Richmond Metal Finishers ("RMF") granted Lubrizol Enterprises ("Lubrizol") an nonexclusive license to use a metal-coating process technology.⁶¹ Pursuant to the licensing agreement, RMF was required: (1) to notify Lubrizol of any patent infringement suit and to defend such suit; (2) to notify Lubrizol of any other licensing of the process and to reduce Lubrizol's royalty fees if the other licensee secured a lower royalty, also known as the "most favored licensee" clause;⁶² and (3) to indemnify Lubrizol for losses arising out of any misrepresentation or breach of warranty by RMF.⁶³ Lubrizol was required: (1) to account for and pay RMF royalties for the use of the process; and (2) to pay off other existing debts.⁶⁴ After filing for Chapter 11 protection, RMF

⁵² *See id.*

⁵³ *See id.*

⁵⁴ *See id.*

⁵⁵ *See id.* at 608.

⁵⁶ *See Group of Instit. Investors v. Chicago, Milwaukee, St. Paul and Pac. R.R.*, 318 U.S. 523, 550, 63 S. Ct. 727, 742, 87 L.Ed. 959 (1943) (applying the business judgment standard to a bankrupt's decision whether to affirm or reject a lease).

⁵⁷ *See id.* (citing *In re Meehan*, 46 B.R. 96 (Bankr. E.D.N.Y. 1985)(denying seller's petition upon finding that the seller attempted to avoid performance of a contract for the sale of a house by filing petition and rejecting the contract)).

⁵⁸ *See id.*

⁵⁹ *See id.* at 609.

⁶⁰ 756 F.2d 1043 (4th Cir. 1985); *see also In re Select-A-Seat Corp. v. Silver*, 625 F.2d 290 (9th Cir. 1980)(finding a software license to be executory where the licensee was obligated to pay the bankrupt vendor royalties from use of the software and the bankrupt vendor was obligated not to sell the software to a third party).

⁶¹ *See id.* at 1045.

⁶² *See Stuart Moskowitz, Intellectual Property Licenses In Bankruptcy: New 'Veto Power' For Licensees Under Section 365(n)*, 44 BUS. LAW, 771 n.37 (1989)(recognizing that "[a]mong contract specialists and attorneys familiar with such terms and conditions, this is known as a 'most favored [nations] licensee' clause").

⁶³ *See Lubrizol*, 756 F.2d at 1045.

⁶⁴ *See id.*

sought, pursuant to section 365(a), to reject this license as an executory contract “in order to facilitate sale or licensing of the technology unhindered by restrictive provisions in the Lubrizol agreement.”⁶⁵ The court acknowledged that it was a well established principle in the law that a contract was not considered executory by the mere fact that one party was obligated to make payments to the other party.⁶⁶ However, the court found that Lubrizol’s other non-monetary obligations were sufficient to find the contract executory.⁶⁷ Surprisingly, the court permitted the rejection of the intellectual property license notwithstanding the fact that such a result would cause serious financial burdens upon Lubrizol.⁶⁸ Because many companies must rely on the vendors to perform the software maintenance functions, rejection of the license could be disastrous for a licensee’s business.⁶⁹ “The *Lubrizol* decision was significant because it not only served to relieve the estate of future performance obligations under the license, but it also effected a complete rescission of the technology transfer.”⁷⁰ (emphasis added). Fearing the impact of the *Lubrizol* decision, the intellectual property community immediately began “[l]obbying efforts to statutorily overturn the Lubrizol decision ... realiz[ing] that it was powerless to either retain the benefit of its bargains or recoup its investment costs when confronted with licensor bankruptcy proceedings.”⁷¹

The Impact of the 1988 Amendments

⁶⁵ *See id.*

⁶⁶ *See id.* at 1046 (citing *In re Smith Jones, Inc.*, 26 Bankr. 289, 292 (Bankr. D. Minn. 1982)).

⁶⁷ *See id.* (stating that the contract would not be considered executory had Lubrizol only owed RMF “nothing more than a duty to make fixed payments or cancel specified indebtedness under the agreement.” However, the court reasoned that Lubrizol’s duty to “deliver written quarterly sales reports and keep books of account subject to inspection by an independent Certified Public Accountant ... [went] beyond a mere debt, or promise to pay money, and was at the critical time executory).

⁶⁸ *See id.* at 1048.

⁶⁹ As stated *infra*, this reliance is due in part because the vendor withholds the source code needed to perform such functions. The following are two examples of software licensees being placed in precarious positions because of vendor bankruptcy:

Case 1: NEC, a Tennessee digital imaging company, licensed software for \$1 million from a vendor that filed for bankruptcy. The software was an integral component of one of NEC’s vital processes. As told by NEC’s vice president, “... without the software, we would not be able to get into it and make changes.... If we did not get the code out of escrow, we could not continue to operate for very long and we would have lost our investment in the software.” *See dsiescrow.com/whatsnew/cip18.html*.

Case 2: Viasoft Inc., a software developer, whose products included third party technology embedded for resale, entered into a licensing agreement, where if one of its suppliers went out of business, Viasoft would be responsible for supporting that technology of its customers. Subsequently, one of the vendors Viasoft was carrying went out of business. After DSI, the escrow company, was unable to locate the vendor, the source code was released to Viasoft. As told by one of Viasoft’s corporate contract administrators, “Had we needed that information for a current product and not been able to find and get cooperation from them, we would have had a serious problem.” *See dsiescrow.com/whatsnew/cip18.html*.

⁷⁰ *See Moskowitz, supra* note 52 at 780.

⁷¹ *See Musone, supra* note 34 at 512.

Because of the far-reaching implications spawned in the wake of *Lubrizol* and its progeny,⁷² on October 18, 1998, President Regan signed into law Public Law 100-506, commonly known as the Intellectual Property Bankruptcy Protection Act of 1987 (“The Act”).⁷³ As stated in the Senate Report on Public Law 100-506, the 1988 amendments to section 365 were:

To make clear that the rights of an intellectual property⁷⁴ licensee to use the licensed property cannot be unilaterally cut off as a result of the rejection of the license pursuant to Section 365 in the event of the licensor’s bankruptcy. Certain recent court decisions interpreting Section 365 have imposed a burden on American technological development that was never intended by Congress in enacting Section 365. The adoption of this bill will immediately remove that burden and its attendant threat to the development of American Technology and will further clarify that Congress never intended for Section 365 to be so applied.⁷⁵

In short, the Act was enacted to give protection to licensees of intellectual property from the rejection of their licenses by a bankrupt vendor. Moreover, “Congress was concerned that the response to the *Lubrizol* line of cases has been to force parties to rely on sales/assignments of technology rather than licenses, with a consequent chilling effect on new technological development.”⁷⁶

Subsection 365(n) of the Bankruptcy Code codified the Act.⁷⁷ With these new amendments, a licensee of intellectual property has two immediate choices should a bankrupt vendor choose to reject the license as an executory contract. First, the licensee may choose to accept termination, and treat the rejection as a breach of contract.⁷⁸ If the licensee should choose to accept this option, “the licensee disavows any right to the continued use of the licensed

⁷² See e.g., *In re Laser Disc Computer Sys.*, No. 87-0042-L, slip op. (Bankr. D. Mass. Oct. 8, 1985)(citing *Lubrizol*, the court granted a debtor/licensor’s application to reject an exclusive technology license with Aimcorp, its licensee, for rights in a laser jukebox technology).

⁷³ See Moskowitz, *supra* note 52 at 771.

⁷⁴ The Bankruptcy Code Section 101 (35A)(2001) defines intellectual property as any of the following:

- (a) trade secret;
- (b) invention, process, design, or plant protected under title 25;
- (c) patent application;
- (d) plant variety;
- (e) work of authorship protected under title 17; or
- (f) mask work protected under chapter 9 of title 17; to the extent protected by applicable nonbankruptcy law.

Note that under the Bankruptcy Code, trademarks are not considered to be intellectual property.

⁷⁵ S. REP. No. 100-505, at 1–2 (1988), *reprinted in* 1989 U.S. CONG. & AD. NEWS 3200.

⁷⁶ See *id.* at 3-4.

⁷⁷ See Musone, *supra* note 34 at 512.

⁷⁸ See 11 U.S.C. § 365(n)(1)(A) (stating that a licensee may elect to treat such contract as terminated by such rejection if such rejection by the trustee amounts to such a breach as would entitle the licensee to treat such contract as terminated by virtue of its own terms, applicable nonbankruptcy law, or an agreement made by the licensee with another entity).

intellectual property.”⁷⁹ Or alternatively, the licensee may elect to enforce the license for the contract term and renewal.⁸⁰ Consequently, while the licensee may continue to exercise its present and renewal licensing rights, the licensee must also continue to pay all royalties to the vendor,⁸¹ and waive any right to either a setoff⁸² or an administrative expense claim.⁸³

If the licensee elects to retain and enforce its rights under the licensing agreement, and notifies the trustee in writing, the trustee must provide the licensee with the intellectual property, including any “embodiments” and “supplementary agreements,” and must not interfere with this right.⁸⁴ The legislative history notes that the reference in subsection (n)(3)(A) to “any agreement supplementary to such contract” encompasses, *inter alia*, source code escrow agreements.⁸⁵ To be safe, the escrow agreement should explicitly state that the escrow agreement is an “agreement supplementary to” the licensing agreement as provided by Section 365(n).⁸⁶

Further, the legislative history notes that “embodiments” include, *inter alia*: (1) a prototype incorporating the intellectual property that had been prepared but not delivered at the

⁷⁹ See Musone, *supra* note 32 at 513.

⁸⁰ See 11 U.S.C. § 365(n)(1)(B) stating:

[T]he licensee may elect to retain its rights (including a right to enforce any exclusivity provision of such contract, but excluding any other right under applicable nonbankruptcy law to specific performance of such contract) under such contract and under any agreement supplementary to such contract, to such intellectual property (including any embodiment of such intellectual property to the extent protected by applicable nonbankruptcy law), as such rights existed immediately before the case commenced, for -

- (i) the duration of such contract, and
- (ii) any period for which such contract may be extended by the licensee as of right under applicable nonbankruptcy law.

⁸¹ See 11 U.S.C.A. § 365(n)(2)(B) (2001)(stating “the licensee shall make all royalty payments due under such contract for the duration of such contract and for any period described in paragraph (1)(B) of this subsection for which the licensee extends such contract”).

⁸² See *id.* at § 365(n)(2)(C)(i)(stating “the licensee shall be deemed to waive any right of setoff it may have with respect to such contact under this title or applicable nonbankruptcy law”).

⁸³ See *id.* at § 365(n)(2)(C)(ii)(stating “the licensee shall be deemed to waive any claim allowable under section 503(b) of this title arising from the performance of such contract”).

⁸⁴ See 11 U.S.C.A. § 365(n)(3)(A) and (B) stating:

If a licensee elects to retain its rights, as described in paragraph (1)(B) of this subsection, then on the written request of the licensee the trustee shall (A) to the extent provided in such contact, or any agreement supplementary to such contract, provide the licensee any intellectual property (including such embodiment) held by the trustee, and (B) not interfere with the rights of the licensee as provided in such contact, or any agreement supplementary to such contract, to such intellectual property (including such embodiment) including any right to obtain such intellectual property (or such embodiment) from another entity.

⁸⁵ See Senate Report, *supra* note 64 at 9.

⁸⁶ See Marc S. Friedman and Helene Hirsch Wingens, *Averting Disaster When Software Supplier Goes Bankrupt*, 8 COMPUTER L. 13, 14 (1991).

time of the petition; (2) genetic materials needed to produce certain biotechnological products; and (3) computer source code.⁸⁷

The legislative history also states that the licensee “is entitled to any judicial relief necessary to enforce” its rights.⁸⁸ However, the licensee is prohibited from compelling post-petition specific performance.⁸⁹ Legislative history indicates that subsections (n)(3) and (n)(4) are only intended to permit access to the intellectual property as it exists on the day of the petition.⁹⁰ These sections do not in any way compel future affirmative performance of development, maintenance, or other on-going undertakings by the trustee.⁹¹

There is one caveat that appears in the code that all licensees should be wary of. As the legislative history notes, “critical to any right of a licensee to obtain such embodiments ... is the pre-petition agreement of the parties that the licensee have access to such material and the physical existence of such material on the day of the bankruptcy filing.”⁹² In other words, for a licensee to obtain the intellectual property, *i.e.*, the source code, the licensee must incorporate a clause in the licensing agreement which provides that both parties agree that the licensee shall obtain the source code in the event of, *inter alia*, vendor bankruptcy.⁹³

As is shown, Section 365(n) provides significant protection to intellectual property licensees. This section gives licensees a choice to reject or retain the benefits of its bargain in the event of vendor bankruptcy. With these changes, a trustee or debtor-in-possession cannot simply reject the intellectual property license and preclude a licensee from gaining access to the source code. Now that this barrier to access is removed, source code escrow may appear to be an unnecessary practice.

CAN A LICENSEE ACCLIMATE ITSELF TO THE SOFTWARE QUICKLY ENOUGH TO AVERT BUSINESS INTERRUPTION?

One of the issues confronting a licensee in the event of vendor bankruptcy, is that even should a licensee manage to procure the release of the source code, the licensee must hire programmers to quickly get up to speed and make whatever modifications are necessary to keep the software functioning.⁹⁴ In theory:

It is possible to maintain a product which was developed using high-level procedural code, using the executable [source]code – if enough very clever people can be found and paid for to decompile large quantities of machine code, and to

⁸⁷ See Senate Report, *supra* note 64 at 10.

⁸⁸ See *id.*

⁸⁹ See *id.* (clarifying the language in (n)(1)(B) by noting that the right to enforce specific performance of the contract is not one of the rights a licensee retains).

⁹⁰ See *id.*

⁹¹ See *id.*

⁹² See *id.* at 11.

⁹³ See Friedman and Vogel, *supra* note 81 (providing a sample clause stating in pertinent part, “Licensor acknowledges that if Licensor as a debtor-in-possession or a trustee in bankruptcy in a case under the Bankruptcy code rejects this Agreement or any agreement supplementary hereto, Licensee may elect to retain its rights under this Agreement or any agreement supplementary hereto as provided in Section 365(n) of the Bankruptcy code...”).

⁹⁴ See *id.* at 12.

understand and modify the resulting code without the benefit of comments, meaningful labels, or meaningful datanames. For a crucial small program, such an undertaking is feasible, but for the hundreds of thousands of lines that make up a complex commercial application, it isn't.⁹⁵

In addition, "the learning curve for a complicated software application is steep and may result in the costs comparable to purchasing a whole new system or application."⁹⁶ As one commentator observed, "software may comprise millions of lines of code. The time taken by a programmer to become sufficiently familiar with the code in order to use it to maintain and develop the software may, in terms of cost for the licensee, outweigh the economies of simply acquiring a new package."⁹⁷

One company suggests that the one way to prevent receiving outdated escrow materials is to "include in the terms or the escrow agreement that the deposit must be updated quarterly or whenever a new version of the escrowed technology is released."⁹⁸ In addition, as explained below, verification services can help prevent receiving outdated source code.

Documentation is also extremely valuable. Without adequate documentation, the nightmare of trying to decompile another developer's source code will become worse.⁹⁹

DOES THE INFREQUENT RELEASE OF SOURCE CODE MAKE THE ESCROW A WORTHWHILE INVESTMENT?

There is no real dispute over the fact that escrow releases are a relatively rare occurrence.¹⁰⁰ One commentator points out that on its web pages, "one company cites a paltry 0.5% release rate, indicating that it releases just one out of every two hundred deposits.¹⁰¹ Consequently, this commentator feels that a licensee may be better to forego the escrow

⁹⁵ See Mezrich, *supra* note 37 at 117 n.4 (commenting on the difficulties licensees may encounter in trying to decompile the source code should they manage to secure its release from escrow).

⁹⁶ *See id.*

⁹⁷ *See id.*

⁹⁸ See Assessing the Myths and Realities of Software Escrow Protection, Case In Point, A Free Publication From DSI Technology Escrow Services.

⁹⁹ See Anthes, *supra* note 15 (highlighting remarks by Scott Heintzeman, Vice President of Knowledge Technologies at Carlson Hospitality Group Inc., the Minneapolis parent company of Radisson Hotels Worldwide, who stated that "If you don't have good documentation, the software might be worthless to you.... How is the software structured? What is the installation process? How is data configured?").

¹⁰⁰ See Mezrich, *supra* note 82 at 121.

¹⁰¹ *See id.* The company cited was DSI Technology Escrow Services. However, Andrew Moore of DSI, argues that comparing the number of escrow releases to the number of escrow deposits simply does not tell the whole escrow story. He suggests that one should think of the "paltry 0.5% release rate" in a slightly different way. He states that "technology escrow is, in essence, 'software insurance'. Moore analogizes that escrow insurance is like a homeowner's fire insurance. "There are not too many homeowners who do not have fire insurance. If typically one out of every 200 homes burned to the ground, I think we'd all see our fire insurance premiums increase substantially. The reality is that fire insurance claims occur far less often (on a percentage basis) than do software escrow releases, and yet people usually buy it."

arrangement before imposing the costs on the vendor, which in turn translates into a higher product price.¹⁰²

THE OBSTINATE VENDOR PROBLEM

Generally, most escrow agreements require that the vendor consent to the release of their source code. This may prove to be a huge stumbling block and as a consequence, “the user-company could face months or years in court while its mission-critical application falls into disrepair.”¹⁰³

This was the dilemma that Radisson Hotels Worldwide (“Radisson”) faced. Radisson had contracted with a vendor to obtain its mission critical reservation system software.¹⁰⁴ Pursuant to a subsequent escrow agreement, the vendor deposited the source code with an escrow agent. The agreement stated that if the vendor did not support the source code, the source code would be released to Radisson where it would maintain the software in-house. When Radisson learned that the vendor was going out of business, it contacted the escrow agent seeking to invoke its contractual right to obtain the source code. However, typical of most escrow agreements, the agreement provided that the vendor had to give prior approval before release of the source code. Not too surprisingly, the vendor refused to give its approval. Fortunately, Radisson was able to persuade the vendor to release the software with the threat of a personal fraud suit against an officer of the vendor’s company. Sadly enough, dealings between vendor and licensee can be quite messy and unpleasant, especially in cases where the vendor refuses to release the source code.

BRIDGING THE GAP

The 1988 Amendments do not obviate the need for the source code escrow. The source code escrow and the Amendments are necessary compliments, wherein the source code escrow fills in gaps left open by the Code. While the Code provides a way for the licensee to recover the source code from a bankrupt vendor, the vital piece it does not provide for is a verification process of the usability of the escrowed source code.¹⁰⁵ It is this gap a well-crafted source code

¹⁰² See Mezrich, *supra* note 82 at 121.

¹⁰³ See *Id.*

¹⁰⁴ See Anthes, *supra* note 24.

¹⁰⁵ For example, DSI offers the following three-tier program for technical verification:

Level I – The purpose is to document a description of the hardware and software environments need to (a) read the computer media, (b) maintain the source code, and (c) compile the source code. The areas documented are the computer hardware and third party software.

Level II – The purpose is to ensure that the computer media is readable, accessible in the defined Level I environment, and not encrypted. The areas that are documented are the computer hardware and software environments of Level I and are used to read the media and produce a file-directory listing, tools used to create backups, with all passwords or keys.

Level III – The purpose is to ensure that the deposited materials can be compiled. The areas documented are the hardware and software environments identified from Level I and are used to prepare a copy of the executable program. In addition, DSI documents the listing of object and executable files, their sized and create date, and any of the variances or exceptions to the object code building process. Each of these three levels has an additional fee amount attached that increases with each level.

escrow agreement can fill. This process works to allay the fears of licensees that the source code is being consistently maintained during the contractual period, and is instantly usable and not useless upon surrender.¹⁰⁶ As one commentator remarked, “[t]he last thing you want to find is that tape you’re pulling out of the deposit isn’t the software but is a copy of the Rolling Stones.”¹⁰⁷

In addition, some escrow agents have suggested that the source code is also a valuable marketing tool for vendors.¹⁰⁸

CONCLUSION

The source code escrow does indeed offer a significant amount of protection for software licensees against vendor bankruptcy, even after the 1988 amendments to the Code. As noted prior, because of the gaps left open in the Code, the amendments fall short of providing a software licensee with adequate protection. However, the escrow does not in and of itself provide an adequate level of protection sought by licensees. Nevertheless, this practice can be an extremely effective source of protection for licensees when used in conjunction with a meticulously drafted licensing and escrow agreement, and in lieu of the 1988 amendments. There are certain points a licensee should be sure to include in its licensing and escrow agreements to maximize the protection of the escrow.

First, the licensee should negotiate a clause that gives the licensee the right to decide when an event triggering release has occurred. In addition, the licensee should include terms in the escrow agreement that clearly spell out which events are “triggering events” for release of the source code. One of the problems with the source code escrow is that vendors and licensees often disagree on whether a release event has occurred because of ambiguity in the agreement.

See dsiescrow.com/services/verif.html.

Another escrow agency, Trexcel Corporation, offers the following two-tier verification process:

Basic Verification – When the Trexcel receives that materials for escrow, the materials are physically inspected to ensure that the physical description and labels correspond with the description in the agreement.

Custom Critical – Trexcel physically examines the contents similar to the Basic Verification. After the examination, the source code is installed on a secure computer system configured specifically to permit the full compilation of the source code into object code. The Object Code is then sent to the Software Developer/depositor for verification. Upon successful verification, the copy is sent to the escrow beneficiary. *See* <http://www.trexcel.com/Services.htm>.

¹⁰⁶ Tom Morehouse, president of SourceFile, a California escrow company, noted that “an independent audit showed that 80% of all the escrowed software he was getting had defects that rendered it unusable.” *See* Anthes, *supra* note [15]. In addition, in the Radisson Hotels example, the software that had been placed in escrow could not have been used to book guests at Radisson’s 500 hotels. The vice president of Radisson’s parent company stated that “[a]s soon as we got suspicious, we quickly called for audit and found out there were many pieces of code and documentation missing.” *See id.*

¹⁰⁷ Remarked by David Weidenfeld, chief technology counsel of McDonald’s Corp. Further, Mr. Weidenfeld stated that “[t]he escrow agreement has not value whatsoever if you are not going to audit the deposits on a regular basis. *See* Anthes, *supra* note 15.

¹⁰⁸ “Vendors also gain a competitive edge by using escrow as a value-added benefit when marketing their technology.” *See* Guard-IT Corp., *Escrow Services: Protect the Developer’s Proprietary Assets and their Licensee’s Investment*, at <http://www.guard-it.com/Services.html>.

Therefore, typical of most agreements, both parties seek arbitration, which can cause enough of a delay to cause the licensee's software to fall into disrepair and become unusable.

Second, the licensee should seek to include release terms that do not require prior approval by the vendor. This may be a tough sell, because vendors always want to maintain control over their source code. Nevertheless, a licensee should try to negotiate this point vigorously. As we saw in the Radisson example, a vendor may refuse to give its approval for whatever reason, and at times, no reason at all. In reality, a vendor may never give its approval for release of its source code. This again, could cause an excessive delay, whereby placing the licensee's business in jeopardy.

Third, the licensee should always deal with reputable, proven escrow agents who have adequate verification process in place. The verification function is extremely important because it, for the most part, gives assurances to the licensee that the source code is being well-maintained, and will be instantly usable upon the release of the code. The last thing a licensee wants to happen is to wait until the source code is released to find out whether or not the vendor had been performing its contractual duties of updating and maintaining the software. This is just too risky a venture. In addition, a licensee should go a step further and have an independent third party, one other than the escrow agent, to perform periodic audits on the source code. This is a way of having two separate bodies verifying the source code for authenticity. This may be overkill, however, when thinking of the risks involved, it reminds me of the old adage, "better safe than sorry."

Lastly, the licensee should negotiate to have the software developer's names and addresses to be included in the escrow account. Thus, if the vendor company goes bankrupt, to prevent any "learning curve" issues on getting up to speed on the software, the licensee could seek to hire these individuals to aid the licensee in updating the software.

This is by no means an exhaustive list of tips for a licensee in negotiating a software licensing agreement. However, these points should be addressed to ensure that the source code escrow does indeed live up to its billing as a worthwhile investment.