

It's a nightmarish scenario. Let's say that you're the head of Information Technology (IT) for a hotel chain. You pay a software development company \$500,000 to create new software for your worldwide reservations service. Then they go bankrupt or for whatever reason refuse to support the software. If you didn't consider access to the source code in your agreement with the developer, you may find that you're unemployed.

Essentially, "source code" is computer programming that humans can read. "Object code" is programming that only your computer can read. Typically, as a user, you only have access to the object code.

Software developers consider the source code to be their most valuable trade secret. With the source code, a competitor could create a competing work without incurring all of the development costs of the original developer. Source code is the lifeblood of the software development business.

From your perspective as the head of IT for that hotel chain, you need the source code to continue the evolutionary development of your software or to fix bugs if your original developer disappears on you. If they can't or won't help, you'll need to hire another company to do the work. This other company's efforts will be severely hampered if they don't have the original source code.

Back to the Beginning

If we go back to the beginning of this transaction between you and the developer, we see that you had competing interests. You needed software developed to run reservations. You felt that for \$500,000, you should own all rights to the final product. The developer wanted to be able to license use of the software to competing hotels and said that "ownership" would cost you far more than \$500,000. So, you agreed on a perpetual, non-exclusive license. You could use it forever, but they could license the final product to others. (Presumably for more than \$500,000 since you were the guinea pig.)

Typically, when you get a license, you only get the object code. Remember that the source code is the last thing the developer wants to give you. Your lawyer blew it if he didn't work out a deal concerning access to source code before you paid your \$500,000.

Typical Arrangement

Clearly, you needed a compromise. You had your very real need to access the source code in case the developer went out of business or filed for bankruptcy, or the developer simply breached their support contract with you. Just as legitimately, the developer needed to keep the source code secret -- locked up in a vault.

The solution was a source code escrow. Source code escrow is when you deposit the source code and other information with a neutral third party (escrow agent). The "other information" is miscellaneous material that a trained computer programmer would need to work with the source code. This might include things like technical manuals, maintenance tools, proprietary utilities and contact information for key technical employees.

The escrow agent's job is to keep a copy of the source code secure and safe. You only get the source code if the developer defaults on support, goes out of business, is involved in a merger or acquisition, or any other predetermined "release condition."

The idea is that if a "release condition" occurs, the escrow agent gives you the proprietary information. This strikes a balance between the parties. The developer can maintain strict control of its company's greatest assets by not giving away its intellectual property, while continuing to satisfy your needs. At the same time, you can protect your investment in the technology and ensure proper support from the developer or access to the source code in case the developer fails. It's a fair deal that no developer should refuse. If you fail to get a software escrow, you and your lawyer should be shot.

Setting Up the Escrow

There are many variations in the way you can establish an escrow. My scenario has been \$500,000 software. This calls for a major league escrow arrangement.

Still, your development project may only be \$25,000, not \$500,000. This might call for a minor league arrangement. Somewhere in between is a continuum of increased diligence and cost. You'll need some expert legal counsel to guide you through all the options in between.

Each escrow will have its own unique twists. First, let's talk about the simpler "minor league" escrow.

The minor league escrow arrangement is appropriate when the size of your investment is smaller. The problem with what I describe as the "major league" arrangement is that it can cost many thousands of dollars. You may not be able to justify this investment for software that costs you only \$25,000.

A typical simple arrangement has the developer depositing the source code and related materials with a bank or attorney who puts the material in a vault. While this arrangement may be better than nothing, the key words are "may be." There are many problems with this simple arrangement. What's on the CD-ROM in that vault? Is it what it purports to be? Is it really everything that a trained computer programmer would need to work with the source code? You may find that your banker escrow agent is lost when you need the source code released from escrow. Bankers are in the banking business not the escrow business.

As your investment in software increases in size, you might want to consider an escrow with a professional software escrow company. You never want your lawyer or bank holding your escrow. They are just not equipped to do this right.

A professional escrow company also can offer you technical verification services. This is an essential service. Unfortunately, it's not always affordable for smaller investments in software. It's absolutely required if your investment is large.

Typically, an escrow company will offer three levels of verification with varying costs. Typical is something like this.

With Level I Verification, they "ensure that the media deposit in your escrow account is readable and complete. They test the deposit to verify that the source code files are identical to the files that the developer warrants are included in the deposit."

With Level II Verification, they "confirm that the source code will convert into object code, and that the complete, deposited source code files are the same as their counterparts residing on your hardware."

With Level III, they "confirm that the deposited source code, when compiled, will process data exactly as the licensed program does."

How far you go with your escrow will mostly depend on how much you invest in the software. The more you invest in the software, the more you should invest in the escrow.

Software escrow is an essential insurance policy. Still, you must remember that unless you take Level III to the max, you'll never be sure of what you have in escrow. When your developer goes bankrupt is not the time to find out that the CD-ROM in escrow is blank.

Mark Grossman's "TechLaw" column appears in numerous publications.

If you would like a consultation with Mark Grossman regarding a legal matter, please call or send him an email. Mark will set up an appointment for a meeting, or a telephone consultation if you are not in South Florida.

Mark Grossman has extensive experience as a speaker as well. If you would like him to speak before your group or corporate meeting, please let us know.

You can find a TechLaw archive at: www.DewittGrossman.com

If you have any comments, please send them to MGrossman@DewittGrossman.com.

I welcome your comments and questions, but due to the volume of email I get, I cannot promise that every comment or question will be answered.

Disclaimer: The advice given in the TechLaw column should not be considered legal advice. This newsletter only provides general educational information. You must never rely upon the advice given here. Your individual situation may not fit the generalizations discussed. Only your attorney can evaluate your individual situation and give you advice. Except as provided below, you may feel free to forward, distribute and copy the TechLaw column if you distribute and copy it without any changes and you include all headers and other identifying information. You may not copy it to a website.

Copyright 2005 Mark Grossman All Rights Reserved