Source Code Escrow: More Than Meets the Eye?

AUGUST 3, 2011

by John Pavolotsky

Before joining my current firm, I worked in an open source company for 2+ years. Once in a while, customers asked for a source code escrow. The immediate response was, there's no need. The source is shipped with the executable (object). By way of further explanation, the software is written in PHP, so the source and executable are one and the same. End of discussion.

Fast forward a few years, and back to the (unchanged) world of source code escrows. While perhaps not the most scintillating of topics, there's still quite a bit to sink your teeth into:

1) Is "source code" defined sufficiently broadly to allow the customer (licensee) to take full advantage of it? This is particularly relevant if the software is hosted in a Cloud environment.

2) Is there a self-escrow (meaning that the escrow will be escrowed with the customer), or will the software be escrowed with a third party escrow company? If the latter, who pays?

3) Is there a meaningful opportunity to be trained on the source before release is triggered? As a licensee, there's a need to assume that licensor's lead engineer will not be available if and when there is a release.

4) Do the release conditions make sense? Here are a few that come to mind:

a) Licensor breaches [any provision of] the license agreement;

b) Licensor fails to function as a going concern or operate in the ordinary course;

c) Licensor is subject to voluntary/involuntary bankruptcy;

d) Licensor is insolvent (under the balance sheet test or by failure to pay debts when they become) or becomes the subject of any insolvency, receivership or assignment for the benefit of creditors proceeding;

e) Licensor defaults on a loan from a third party;

f) A change of control occurs;

g) Licensor terminates the license agreement prior to the applicable termination date.

Note that most of these are negotiable. For example, there is no reason why a change of control, by itself, should affect licensor's ability to maintain or support the software. The same logic applies to a loan default, and so on. For the bankruptcy condition, should the trigger be a Chapter 7 proceeding and/or a liquidating Chapter 11 proceeding? Should the trigger in a Chapter 11 proceeding be a rejection by the debtor (licensor) of the license agreement? Put otherwise, even in a bankruptcy, if the debtor is performing the license agreement, what difference should it be to the licensee?

5) What is the scope of the source code license? Is it to modify the software to in effect provide the support and maintenance services that should have been provided by the licensor, or is it broader, allowing the licensee to modify or enhance the software to, e.g., remain current with technological innovations? When does the license issue? On the effective date of the license agreement (coupled with a covenant not to exercise the rights under the license until there is a release condition) or when there in fact is a release condition?

6) Who owns the modifications to the source, if there has been a release?

7) When does the source code return to escrow?

What are the obligations (confidentiality, etc.) of licensee once there has been a release?

 Of course, there's more, but that can be be the subject of a future post. Comments welcomed.

John Pavolotsky
Of Counsel
Greenberg Traurig, LLP.
pavolotskyj@gtlaw.com

John Pavolotsky's practice focuses on technology transactions and other intellectual property matters. Prior to joining GT, John was Corporate Counsel of SugarCRM Inc. and General Counsel of Fourth Dimension Software. Prior to earning his law degree, John worked as a Research Associate at the University of California, San Francisco. John's industry experience includes travel automation, customer relationship management, commercial open source, and pharmaceutical research.