

Software Escrow: Protecting business critical software in the Middle East

Sana Saleem - Associate - Digital and Data
- Dubai International Financial Centre

April 2017

Software escrow might be requested by a licensee as a means to mitigate its risk during the period the critical business software is being developed for the licensee and the period during which the licensee requires the use of the business critical software. Generally, under a software escrow arrangement, the escrow agent is required to release the software source code to the licensee in specific scenarios (e.g. if the licensor files for bankruptcy). Access to the source code and associated documentation allows the licensee (through its programmers) to maintain the software, modify it (for example, to fix bugs or errors) or to enhance it (for example by building new functionality).

Considering software escrow

Software escrow is a useful means of minimising the risk of using third party supplied business critical software. It is an essential part of business continuity and disaster recovery planning. It serves to secure the storage of business critical source code and provides protection whilst the software is being developed, and for as long as the software is in use by the customer in the operation of its business.

Software escrow should always be considered by the licensee of any third party software, particularly custom developed business critical software. Some factors to take into account as a licensee when deciding whether to request the licensor to place the software in escrow, include:

- the importance of the software in the operation of the business;
- the potential impact to the business and customers (such as financial or reputational loss) in the event the software is corrupted or otherwise ceases to function as intended;
- whether the software is difficult or costly to replace if the solution is no longer supported by the vendor (whether due to insolvency or otherwise); and
- the organisation's corporate risk management strategy.

Software escrow agreement

A software escrow agreement is typically a three party contract that governs the procedures and terms of the escrow process between the licensor, licensee, and escrow agent. The agreement normally outlines the procedures for the deposit of the source code and handling of the source code by the licensor and escrow agent, including what will be deposited (updates, customisations, etc.), how often the deposits are to occur, how the escrow agent is to receive the source code, and where and how it is to be stored.

The most heavily negotiated provisions in a software escrow agreement are those relating to the events that trigger the release of the source code. In essence, the aim is to ensure that the customer is able to obtain access to the software in the event the licensor is unable to continue maintaining and supporting the software. The trigger events are negotiable and typically include:

- bankruptcy of the licensor;
- the discontinuance by the licensor of its business;
- the dismissal of substantially all of the employees of the licensor / service provider, or the employees

- that develop and/or provide maintenance for the licensed software;
- failure by the licensor to comply with the licensing agreement (such as failing to provide required support services);
- the discontinuation of support by the licensor for the type or version of software licensed to the licensee; and
- the occurrence of a change in control event in relation to the licensor / service provider.

Software escrow and the cloud

Under a traditional software escrow arrangement, if the supplier were to become bankrupt, for example, the customer might be able to access the source code and hand it over to a new service provider for the purposes of maintaining the software so that the customer could continue to utilise the software

A software escrow arrangement in the context of a cloud offering is slightly more complicated. Firstly, the supplier may not want to enter into escrow arrangements with all of its customers (particularly those who are paying bargain prices). Secondly, if the parties were to opt for a public cloud SaaS offering, then it is very likely that a standard version of the software will be offered to all customers. Even if the software were to be bespoke, due to the very nature of cloud services there is likely to be very little software installed on the customer's equipment to begin with as the software would be accessed via a web browser. Finally, if the source code were to be made available to the customer, on its own, it would not be sufficient to allow for continuation of the SaaS. At a minimum, access to the following would also be required to ensure continuation of service:

- documentation in relation to the source code and access to critical systems and portals;
- information regarding network configuration and topology; and
- knowledge of administrative processes and procedures.

The good news is that cloud software suppliers are becoming more open to entering into escrow agreements in order to reassure customers of the vendor's commitment to best practice around service continuity. Separately, an increase in the use of the cloud is resulting in an evolution in escrow services where the parameters of escrow services are changing to account for SaaS offerings. Escrow is increasingly being used to provide continuity for SaaS; however, it must be carried out properly. An effective cloud escrow service would, for example, provide for verification and integrity tests on each deposit to ensure it is accessible, virus free, and consists of the correct type of material, insists upon regular deposits of code as the supplier updates and maintains the software, and generally ensures that the service delivered is tailored to the cloud's peculiarities and is as watertight as possible.

Al-Tamimi & Company's [Technology, Media & Telecommunications team](#) regularly advises on technology related issues including software escrow. For further information please contact Sana Saleem (s.saleem@tamimi.com) or Nick O'Connell (n.oconnell@tamimi.com).