

**The ABC's of Software Licensing and
Essential Elements for Software
Development or Licensing Agreements**

Prepared by

Raymond R. Bonnabeau¹
Hellmuth & Johnson, PLLC
8050 West 78th Street
Edina, Minnesota 55439
Telephone: 952-746-2121
E-mail: rbonnabeau@hjlawfirm.com
Web: hjlawfirm.com

¹ Mr. Bonnabeau is a partner at the law firm of Hellmuth & Johnson, PLLC and heads the firm's Information Technology (IT) and eCommerce Practice Group. He has extensive experience in drafting, reviewing and negotiating IT contracts, including numerous eight figure agreements. His practice has focused on representing healthcare, financial and insurance organizations in the procurement, development and implementation of IT. He also provides IT seminars and training sessions for clients. This article was originally presented as part of a continuing legal education seminar through All District Legal Education.

<u>Table of Contents</u>	<u>Page</u>
A. SOFTWARE LICENSING AGREEMENTS.....	1
I. INTRODUCTION.....	1
II. SOFTWARE	1
III. TYPES OF SOFTWARE LICENSES	1
IV. THE SOFTWARE LICENSE	2
V. ACCEPTANCE TESTING.....	4
VI. SOFTWARE WARRANTIES.....	6
VII. SOFTWARE SUPPORT.....	9
VIII. SOURCE CODE.....	11
B. SOFTWARE DEVELOPMENT AGREEMENTS.....	12
I. INTRODUCTION.....	12
II. SOFTWARE DESCRIPTION.....	12
III. OWNERSHIP	12
C. CONCLUSION	15

A.

SOFTWARE LICENSING AGREEMENTS

I. INTRODUCTION

Although there are a vast number of essential elements in all software licensing agreements, this section will only address the most rudimentary main essential elements within a “good” software licensing agreement. While a software vendor (“Vendor”) and a software user (“User”) may disagree upon what constitutes a “good” software license agreement, a mutual understanding of the business/technical objectives and a clear and concise software license agreement will benefit both the Vendor and User.

With the foregoing in mind, the main essential elements addressed within this section are:

- a. The terms of the software license;
- b. Acceptance testing of the software;
- c. Software warranties;
- d. Software support; and
- e. Source Code.

II. SOFTWARE

The right to use Software is typically given in the form of a license, rather than a sale. A sale would give the User all rights that are associated with ownership of the Software (e.g., sale or license of the Software to third parties, the right to copy and create derivative works). As Vendors desire to retain such rights in the Software, the Vendor conveys limited rights to the User in the form of a license.

III. TYPES OF SOFTWARE LICENSES

There are a variety of Software licenses available to Users. Software licenses may range from a license to use Software on a single computer, to a concurrent user basis at a single (or multiple) geographical site(s), or for use on a non-concurrent user basis at a single (or multiple) geographical location(s). To determine the appropriate license, one needs to determine who needs to use the Software and where such use is desired.

- a. **Single Computer License.** This is the most basic license as use of the Software will be limited to one computer (e.g., this is the typical license received when one purchases off the shelf software for personal use).
- b. **Concurrent User Licenses.**
 - i. Concurrent user licenses will allow a pre-determined number of

Users to simultaneously use the Software. Typically, the license fee will be based on the number of concurrent users, with a User having the right to purchase additional licenses should increased use be desired. Such licensed Software will often have a monitoring program which will limit use to the authorized number of Users. Therefore, additional Users will be denied access and be typically notified, by way of a screen prompt, that access was denied as the maximum number of authorized Users are presently using the Software.

ii. Software licensed on a concurrent user basis will often consist of the following two (2) software components: (a) a server component and, (b) a client component. The server component will be resident on a server, with the client component resident on remote computers (e.g., personal computers). Remote computers will be networked (or connected) to the server and the client component will enable a User to access and use the server software.

iii. Concurrent user licenses may limit access to one or more geographical site(s). For example, access may be limited to the authorized number of concurrent users located at the User's site.

c. **Non-Concurrent Licenses.** Non-concurrent user (non-single seat) licenses will enable any number of "permitted users" to use the Software. Typically, "permitted users" will be limited to employees, agents or contractors of the User. Should additional "permitted users" (e.g., non-employee healthcare providers), be desired by the User be sure to secure such use rights. Further, use of the Software may be limited to one or more geographical site(s).

IV. THE SOFTWARE LICENSE:

a. **Terms.** There are a number of different terms used in describing the type of license granted by a Vendor. Below are some of the common license terms used in software license agreements, along with their common meaning.

For example, a Vendor could grant a User an "nonexclusive, nontransferable, paid-up and perpetual license" to use the Software.

1. **Nonexclusive.** The nonexclusive designation permits the Vendor to license the Software to other Users.

2. **Nontransferable.** The nontransferable designation limits the User's rights to transfer the license to other parties (i.e., the license is limited to the User only).

3. **Paid-up.** The paid-up provision is used to establish that the User is not required to pay additional "license fees" (beyond the initial payment amount) for license rights to the Software.

4. **Perpetual.** The perpetual designation means that the User's license rights to the Software is of an unlimited duration (e.g., no renewal).

b. **User:**

1. A User should be sure to allow all intended Users to use the Software. Should the User want affiliates and/or other persons and entities to use the Software, the User should be sure to clearly set forth such use rights in the Agreement. A User must also consider whether it wants to license the Software as a site license, or on a concurrent user basis, or simply on a single computer.

Vendor: Should carefully limit both the type of Users and the site's at, or from which, access to the Software may be made. In addition, should expressly prohibit any direct or indirect competitor of Vendor from accessing the Software].

c. **User:**

1. If a User desires to use the Software for an indefinite period of time and is paying one amount for such use rights, the User should obtain a paid-up and perpetual license to use the Software.

Vendor: Should attempt to condition the Software license to the payment and continuation of support (i.e., as long as the User is paying for support, the User's license to use the Software continues). If the Vendor desires not to condition the continuation of the license to support and the Software license is to be perpetual in nature (e.g., pay one license fee amount for perpetual use of the Software), the Vendor should insist that the User will obtain a paid-up and perpetual license *only after* payment of all license fees for the Software. In addition, the Vendor should insist that the license to the Software will terminate upon termination of the Agreement].

d. **User:**

1. If the Software license is granted only for use on a specific identified computer or server, the User should obtain the right to transfer the Software to compatible, upgraded or successor computers or servers without paying a fee (e.g., license, transfer or upgrade fee).

Vendor: If acceptable to Vendor, Vendor should make sure that the User is obligated to pay for any Vendor services provided to transfer such Software. In addition, Vendor should make sure the User is obligated to pay for any additional hardware/third party software necessary as a result of such a transfer].

V. ACCEPTANCE TESTING:

The acceptance provision is one of the most important provisions in a software license agreement. First, even with standard “off the shelf” Software, the User and Vendor will never know whether the Software will live up to the parties’ expectations until it is actually installed and used. Second, the need to develop sufficient and objective acceptance criteria will force User and the Vendor to come to a clear understanding of what the Software is supposed to do in User’s business environment. Third, from a User’s perspective, warranty provisions can be drafted to incorporate the acceptance criteria, thereby making them standards that the Vendor is committed to maintaining.

a. Acceptance Testing Criteria:

1. User:

- a. If the User has developed acceptance testing criteria (or other performance requirements), such criteria should be provided to the Vendor as soon as possible. User should request that the Vendor submit a written response detailing the Software’s capability to meet such performance criteria. Thereafter, the acceptance criteria and the Vendor’s response should be attached to the Agreement.
- b. If acceptance criteria has not been developed, User should consider developing (either individually or jointly with the Vendor) such criteria as soon as possible. Further, if the Vendor proposes to use its own testing criteria, User should make sure to obtain a copy of such criteria to determine its adequacy.
- c. User should make sure all agreed-upon acceptance testing criteria are attached to the Agreement and expressly incorporated into the Agreement.

[Vendor: Limit acceptance testing criteria to Vendor’s own documents (e.g., current user documentation). If additional acceptance testing criteria is required, be sure such is acceptable. Lastly, should the User want to use the Vendor’s response to proposal as performance criteria, be sure to review such response to avoid “puffing”].

b. Commencement of Acceptance Testing:

1. **User:**
 - a. Acceptance testing should commence *no earlier* than upon successful installation of the Software at User’s site. In addition, should third party interfaces and/or any other User components (e.g., hardware) need to be installed prior to acceptance testing, User should consider having acceptance testing commence after all of the foregoing are installed.
 - b. Acceptance testing should occur in a “production environment” within the User’s own internal environment on User’s own equipment.

Vendor: Acceptance testing should occur as early as possible (e.g., upon delivery of the Software). If delivery of the Software is not acceptable to the User, be sure to backend the date upon which acceptance testing will commence (e.g., testing will commence upon the earlier of: (i) completion of the tasks set forth in the project plan which are a pre-requisite to the commencement of acceptance testing; or (ii) thirty (30) days after installation of the Software. Such a provision places the burden of non-Vendor obligations (e.g., the installation of User hardware) onto the User and acceptance testing would not be delayed due to such occurrences. In addition, Vendor should expressly exclude responsibility for non-performance issues caused by non-Vendor deliverables (e.g., third party hardware)].

c. **Conformance to Acceptance Testing Criteria:**

1. **User:**
 - a. User should insist on “strict” conformance with the acceptance testing criteria. As a fallback, User could agree that the Software needs to “perform in all material respects” to the acceptance testing criteria.

Vendor: Resist “strict” conformance, as such could be used by the User to cite a “slight” non-conformance issue as a basis for non-acceptance. Insist that the Software should “substantially” conform to the acceptance testing criteria].

d. **Testing Period/Remedy:**

1. **User:**

- a. Provide for an adequate number of days to test the Software.
- b. Provide for the possibility that the Software will not pass acceptance testing (e.g., allow Vendor a certain number of days (“cure period”) to correct the non-conformance *at no cost to User*).
- c. If a cure period is acceptable to User, User must have a period of time to re-test the Software after expiration of such period. Be sure to expressly state when User’s re-testing period will commence and end.
- d. Provide User with the right to terminate the Agreement in the event the Software does not pass acceptance testing during the re-testing period.
- e. If User terminates the Agreement, Vendor should give User a full refund of all sums paid to Vendor.

[Vendor: Insist on a limited testing period (e.g, the Software shall be deemed to have met the acceptance testing criteria and be deemed accepted by User upon the Software “substantially” performing in accordance with the acceptance testing criteria). Provide for an adequate “cure period” in the event of non-conformance. Be sure to exclude non-performance issues caused by non-Vendor deliverables (e.g., User’s hardware). In addition, consider limiting refund to “license fees” paid to avoid express contractual obligation to refund implementation/training and other fees].

VI. SOFTWARE WARRANTIES

A Vendor should attempt to cast a software license in the form of a commercial transaction (e.g., the moment you drive it off the lot it is yours with all faults). However, a User should view a software license agreement as an investment (e.g., that the User may be paying for the Software with the expectation that the User will be able to use such Software for many months, if not years, to come). In either event, warranty provisions should be carefully drafted and reviewed by both Vendors and Users.

- a. **Warranties:** Generally, software warranties are comprised of performance warranties (i.e., Software will perform in accordance with certain performance criteria) and non-performance warranties (e.g., Vendor owns the Software; no disabling code; scanning for viruses, etc.).

1. **User:**

- a. Vendor should warrant that it owns the Software or, to the extent it does not own the Software, it has all rights necessary to grant to User the license under the Agreement.

[Vendor: this warranty is usually not an issue, however, the Vendor should limit this warranty as follows: “that it owns the Software or, to the extent it does not own the Software, it has all rights necessary to grant to User the license to the Software for use within the terms of this Agreement].

- b. Vendor should warrant that the Software will not contain any disabling code (e.g, a software mechanism which will automatically disable the Software upon the occurrence of a certain event or upon passage of a certain pre-determined time period).

[Vendor: this warranty often is not an issue, however, the Vendor should limit the scope of this warranty as follows: “to the best of Vendor’s knowledge” the Software will not contain any disabling code].

- c. Vendor should warrant that it has used its best efforts to scan for viruses within the Software.

[Vendor: Resist “best efforts” language and insist, at a minimum, to a “reasonable efforts” standard. In addition, consider proposing that the Vendor has used commercially reasonable efforts to scan for viruses using commercially available virus software].

- d. Vendor should warrant that the Software is Date Compliant.²

[Vendor: If a Date Compliance warranty is acceptable, be sure to exclude non-conformance issues caused by non-Vendor deliverables, input errors, etc.].

- e. Vendor should warrant that the Software will not infringe on any patent, copyright, trade secret, trademark or any other third party proprietary rights.

² Although the new millenium arrived worldwide some years ago, Users should still insist on including a Date Compliance warranty provision within software licensing agreements. Most (if not all) vendors will assert that a user’s date compliance concerns should not exist as we are well past the year 2000. However, the passage of time will not, in and of itself, render a software application able to correctly and accurately calculate dates among and between different centuries.

[Vendor: Should resist this warranty and insist that the User’s infringement concerns are addressed under the Indemnification provision within the Agreement. This will reduce the likelihood of the User having both an indemnification and warranty claim].

- f. Vendor should warrant that the Software will conform to the descriptions, standards and performance criteria (including the acceptance testing criteria) contained in the Agreement.

[Vendor: Should resist “strict” conformance and insist on “substantial” conformance. In addition, Vendor should limit performance criteria to its current documentation (e.g, user manuals, etc.)].

- g. Vendor should warrant that the Software will conform to all state and federal laws and regulations to enable the User to use the Software as set forth in the Agreement.

[Vendor: At a minimum, resist the inclusion of “state” laws and specify which federal laws apply. In addition, exclude non-conformance issues caused by non-Vendor deliverables and consider inserting language allowing the Vendor to “pass back” to the User (on an equitable basis) costs incurred by the Vendor in complying with such federal laws].

b. Warranty Commencement/Warranty Duration:

1. **User:**

- a. All performance warranties (i.e., VI.f. above) should commence upon User’s acceptance of the Software.
- b. All non-performance warranties should commence upon execution of the Agreement and continue thereafter throughout the duration of the Agreement.
- c. Initially, attempt to have all performance warranties continue throughout the duration of the Agreement as long as User is receiving support from the Vendor. Should the Vendor fail to agree to such a provision, make sure to expressly set forth the number of days such performance warranty is in effect.

[Vendor: All warranties should commence as soon as possible (e.g., upon delivery of the Software) and be

limited for a time certain (e.g., 90 days after delivery of the Software). In addition, include language that the Vendor will not be responsible for warranty non-conformances caused by non-Vendor deliverables or otherwise not due to Vendor (e.g., modification of the Software by User)].

c. **Warranty Breach:**

1. **User:**

- a. Provide Vendor with a set number of days to correct any warranty non-conformance *at no cost to User* (such period should commence upon Vendor receiving User's notice of non-conformance).
- b. In the event Vendor is unable to correct the non-conformance, User should receive a pro rata refund of all sums paid to Vendor under the Agreement based on a five (5) year straight-line depreciation calculated from the date of User's acceptance of the Software. **Please note:** The above guideline suggests a five (5) year depreciated refund based on the assumption that the Software will have a five (5) year useful life. Should the Software have a shorter/longer estimated useful life, the provision should be accordingly modified].

Vendor: Again exclude warranty non-conformances caused by non-Vendor deliverables or otherwise not due to Vendor (e.g., modification of the Software by User). Resist "refund" language or, at a minimum, reduce the length of the useful life for the Software and limit refund to "license fees" paid].

VII. SOFTWARE SUPPORT

Often, Software support provisions are given less consideration than necessary by Users. As support provisions may provide a User with its only leverage to obtain post-warranty fixes. Conversely, support provisions provide a Vendor with an opportunity to limit its exposure and liability under a software license agreement.

a. **User:**

1. Vendor should provide User with all updates, upgrades, releases, enhancements, modifications and versions ("Releases") of the Software at no additional cost to User. In the event the Vendor will not agree to provide User with

subsequent Releases at no cost to User, User should obtain a price discount for such Releases.

[Vendor: Vendor should, at a minimum, exclude what it may consider as “new products” from the definition of Releases. In addition, Vendor should include language obligating the User to pay for any services performed by Vendor in the installation and implementation of Releases].

2. Support services should be clearly set forth. For example, support services could include:
 - a. 24 x 7 x 365 toll-free phone support;
 - b. On site support within a certain period of time after a request for on-site support service; and
 - c. Support response times (e.g., User to receive a callback from Vendor within X amount of time after a report of a “critical” error, etc).

[Vendor: Should attempt to have support obligations written as support “objectives or goals”. For example, “It is the Vendor’s goal to provide User with on-site support within X amount of time after receiving a report of a critical error”)].

3. Support should commence upon either (i) the expiration of the performance warranty period, or (ii) upon acceptance of the Software. If support is to commence upon acceptance of the Software, such support should initially be provided at no cost to User [as User should not have to pay the Vendor to correct warranty non-conformances]. A User should resist provisions tying the User’s license rights to the payment of support.

[Vendor: As support is a revenue stream for a Vendor, support should commence as soon as possible (e.g., upon installation of the Software). In addition, consider tying the User’s right to use the Software to the continuation of support services (e.g., as long as the User pays for Support the User retain it’s license to use the Software)].

4. Vendor should warrant that all support services will be performed in a good and workmanlike manner consistent with acceptable industry practices. Vendor should provide all support necessary to continue the warranties under the Agreement at no additional cost to User.

[Vendor: Resist a “support warranty” as such may be used by the User as a basis of a warranty claim. This is less of an issue if the support obligations are a part of separate agreement].

5. A Vendor’s support obligations should be part of the software license agreement. As a result, a Vendor failure to perform such support obligations would likely be a breach of the Agreement and afford the User with remedies under the software license agreement.

[Vendor: Insist on separating Vendor’s support obligations (and the User’s remedies) from those contained within the software license agreement. This can be done by executing a separate support agreement. Attempt to limit Vendor’s support obligations to “repair or replacement” and limit the User’s remedies to “repair or replacement” and/or “to support fees paid”. By doing so, the Vendor may eliminate it’s liability under the software license agreement in the event the Vendor breaches the support agreement].

VIII. SOURCE CODE

a. General: The term “object code” refers to the “machine readable” version of the Software program. Whereas, the term “source code” refers to the “human readable” version of the Software program (i.e., software code written in programming language by a programmer). The “source code” version of the Software program would provide the underlying programming structure of the Software to a User and consequently, Vendors should be reluctant to make available such to Users. However, in the event a Vendor is unable to perform its obligations under the Agreement, the User may want the option to continue to use the Software. However, such continued use may require that User have the Software source code on site.

b. User:

1. User should clearly set forth the terms governing delivery of Software source code (e.g., source code release events, use rights, etc.).
2. **Release events:** Delivery should occur in the event the Vendor or its successor corporation which assumes its obligations under the Agreement (i) ceases to transact business or (ii) maintain its computer software research, development, and support services at levels

sufficient to meet its obligations and responsibilities under the Agreement on an ongoing basis.

[Vendor: Should release source code only in the event Vendor ceases to transact business].

3. User should secure the right to for it, or a third party, to use, copy, modify, maintain and enhance the source code, documents and descriptions for User's [and, if applicable, User's Affiliates] internal use only.

[Vendor: Should not allow User to disclose the source code to any person or entity which directly or indirectly competes with Vendor. In addition, Vendor should insist that the source code of the Software is subject to the confidentiality provisions within the Agreement].

B.

SOFTWARE DEVELOPMENT AGREEMENTS

I. INTRODUCTION

As with software license agreements, there are a number of essential elements within software development agreements. However, for the purposes of discussion, the two most essential elements (i.e., description of what is to be developed and ownership) are addressed within this section.

II. SOFTWARE DESCRIPTION

a. Clearly defining the transaction is the single most important element of any software development arrangement ("Development Agreement"). The parties' relative duties and expectations should be set forth in a detailed Statement of Work ("Statement of Work") or a change order that alters an existing Statement of Work.

b. Services and deliverables should be clearly spelled out in a Statement of Work that is attached and expressly incorporated in the Development Agreement by reference. Depending upon the nature of the engagement, the Statement of Work should contain development/performance milestones, detailed descriptions and specifications of the services and/or deliverables, acceptance criteria, payment schedules, and rate schedules.

III. OWNERSHIP

Numerous issues arise in the context of ownership of the Software developer's work. Without express terms in the Development Agreement relating to ownership, unintended results are likely, especially with respect to intellectual property rights.

a. **Copyright Ownership.** Copyrightable work created by an employee or Software developer is governed by the “work made for hire” doctrine. Under the Copyright Act, a work made for hire is defined as: (1) a work prepared by an employee within the course and scope of his or her employment; or (2) a work specially ordered or commissioned for use as a contribution to a collective work, a part of a motion picture or audiovisual work, a translation, a supplementary work, a compilation, an instructional text, a test, answer material for a test, or an atlas, *so long as the parties expressly agree in writing that the work constitutes a work made for hire.* 17 U.S.C. § 101. The employer or commissioning party is deemed the author of copyrightable material that qualifies as a work made for hire. 17 U.S.C. § 201(b). In the absence of an express agreement, the Software developer will own the intellectual property rights in any copyrightable work developed, even though the User ordered and paid the Software developer to develop such work. Kirk v. Harter, 199 F.3d 1005, 1008 (8th Cir. 1999).

b. **Rights Included In Ownership of Copyright.** Ownership of the copyright in a work includes the right to reproduce the work, make derivative works based on the original work (new version of a computer program), the right to distribute copies of the work, and the right to perform and display the work publicly. 17 U.S.C. § 106.

c. **Conveyance Issues.**

i. **Property Rights vs. Copyright.** Property rights are separate from those rights conferred by federal copyright law. 17 U.S.C. § 202. Accordingly, the Development Agreement should separately address ownership of property rights and ownership of copyrights and other proprietary rights.

ii. **Conveyance of Specially Commissioned Works.** When the Software developer is an independent contractor, simply agreeing that the developer’s work is a “work made for hire” may not actually transfer the copyright. Copyrightable work designated as a “work made for hire” will be deemed as such only if it falls into one of the nine statutory categories identified above. In the case of custom programs and particularly with respect to derivative works (new versions), the “work made for hire” designation may not apply. In order to successfully transfer the copyright to the Developer’s work in that case, the Development Agreement must include an express written assignment of the Developer’s copyright in the work. Such an assignment is subject to termination in the future by the author or the author’s heirs under 17 U.S.C. § 203 (permits termination of a grant of rights 35 years after the transfer).

iii. **Developer’s Interests.** In some cases the Software Developer may want to own the work developed under the Statement of Work. Should that be acceptable to the User, the User should, at a minimum, obtain a paid-up and perpetual license to use the deliverable.

d. **Checklist.**

- i. The parties should agree who owns the rights, title, and interest in the physical deliverables (reports, programs, manuals, tapes, listings etc.), if any and such interest should be expressly assigned to that party.
- ii. The parties should agree who owns the intellectual property rights in the deliverables and intangibles (ideas, concepts, information, inventions, discoveries, creations, specifications, technical data etc.).
- iii. If the parties agree that the User will obtain the copyrights to any of the developer's work, those works should be deemed "works made for hire" and the copyright should be expressly assigned to the User. The developer should agree to execute and sign any and all applications, assignments, or other instruments that the User deems necessary to obtain such rights in the United States and foreign countries.

e. Assumptions.

1. Developer is customizing existing Core Technology
2. Developer owns Core Technology
3. Specifications for customizations owned by User
4. Customizations are exclusive to User

Contract language specifies that all customizations constitute a Work Made for Hire and User owns all rights, title and interest in such customizations. Developer can not use such customizations or general knowledge gained through the development of such customizations for any derivative works

f. Assumptions.

1. Developer is customizing existing Core Technology
1. Developer owns Core Technology
2. Specifications for customizations co-developed

Contract language specifies that all customizations are owned by Developer. Developer may use such customizations or general knowledge gained through the development of such customizations for any derivative

works. User entitled to front-end discount or royalty payments from subsequent sales. User obtains a nonexclusive, paid-up and perpetual license to use the customizations.

g. Assumptions.

1. Developer is customizing existing Core Technology
2. User owns Core Technology
3. Specifications for customizations owned by User
4. Customizations are exclusive to User

Contract language specifies that all customizations constitute a Work Made for Hire and User owns all rights, title and interest in such customizations. Developer can not use such customizations for any derivative works. Developer may use general knowledge gained through the development of such customizations for any derivative works.

C.

CONCLUSION

Although the essential elements of software licensing agreements and development agreements addressed in this article are not intended to be exhaustive, such should provide the reader with guidance and information necessary to better understand such issues and how they may be used to make a better agreement for you and your client.