HOW TO

Improve Your Company's Form Software License Agreement

A ten-part article published by Koley Jessen to help software licensors improve their form software license agreements.

KOLEY JESSEN

ATTORNEYS

TABLE OF CONTENTS

PART 1: LIMITATION OF LIABILITY

KEY TAKEAWAYS

1. A form software license agreement should contain a waiver of consequential damages provision and an aggregate liability cap provision in favor of the licensor.
   These two provisions are often referred to collectively as "limitation of liability"

2. Licensors should be very careful when agreeing that certain damages and liabilities are exceptions to the limitation of liability provisions and subject to unlimited liability.

3. If a licensor agrees to allow the limitation of liability provisions to limit the customer's liability ("make them mutual"), the licensor should add certain additional exceptions to the limitation of liability provisions that are subject to unlimited liability.

Contracts are, in essence, a set of legally enforceable promises. If a party breaks one of its contractual promises, the other party may be able to obtain remedies (often in the form of money damages) from the party that broke its promise. Notwithstanding the foregoing, even if a licensor breaks one of its contractual promises, it is customary and "market" for a licensor to disclaim certain types of damages and to limit its aggregate liability under the software license agreement. The primary rationale behind these provisions is that no one commercial contract is worth "betting the company" by exposing the licensor to unlimited liability.

As a starting point, a licensor's form software license agreement should contain a waiver of "consequential damages" provision in favor of the licensor. This provision says that the licensor will not be liable for any consequential, incidental, and other indirect or punitive damages—often referred to collectively as "consequential damages"—resulting from the licensor's breach of the software license agreement. Broadly put, consequential damages are damages suffered by a party that flow indirectly from the other party's breach of contract. A waiver of consequential damages provision can provide a lot of protection to the licensor because it expressly says that many potential damages are not recoverable by the customer if the licensor breaches the software license agreement.

The form software license agreement should also contain an aggregate liability cap provision in favor of the licensor. This provision says that the licensor's total aggregate liability for all claims relating to the software license agreement will not exceed a certain amount (the "liability cap"). The liability cap figure is typically a function of fees received by the licensor under the software license agreement, i.e., 12 months' fees. Although the liability cap concept seems simple, the language can have a significant amount of nuance. For example, a licensor should ensure that its liability cap is an aggregate cap that limits the licensor's aggregate liability in connection with any and all claims relating to the software license agreement, rather than a "claim by claim" liability cap or some other variation that only caps a subset of the licensor's liability. The licensor should also consider its insurance coverage when determining acceptable liability caps under software license agreements.

We can use an example to demonstrate the criticality of these provisions for licensors: let's assume a licensor licenses a logistics software tool to airlines that is mission critical for airlines to schedule their flights across the globe. In this example, such licensor licenses the software to an airline for $500,000 per year. Let's further assume that the software delivered to the airline contains a bug in breach of the software license agreement, and the airline is forced to halt operations for one day. Finally, we will assume the airline incurs $10 million in damages due to the software bug – costs to fix the bug as quickly as possible, costs to perform the same business function via an alternate method, lost profits due to flight cancellations, costs to perform a press campaign to fight bad publicity, costs to respond to regulatory investigations, etc.

Without the standard limitation of liability provisions, it is possible that the licensor could be liable for all $10 million. With properly worded aggregate liability cap provision, it is possible that the licensor would only be liable in amounts that would in no event exceed $500,000. Moreover, with a properly worded waiver of consequential damages provision, the licensor's liability could be even lower because many of the damages, costs, and expenses suffered by the airline would be deemed consequential damages.

Customers will oftentimes edit limitation of liability provisions so that certain damages and liabilities are exceptions to the limitation of liability provisions, meaning that the licensor will not get the benefit of the liability cap and the wavier of consequential damages provision if it breaches the software license agreement. Although the current trend for technology providers is to avoid unlimited liability more aggressively than in the past, it is still common for licensors to accept unlimited liability for the following:

(1) fraud and willful misconduct;

(2) breach of confidentiality obligations; and

(3) indemnification obligations for third-party claims that allege the software infringes third-party intellectual property rights.

Any additional exceptions to the limitation of liability provisions may be outside of "market" terms. In our example above, if the licensor agreed to broad exceptions to the limitation of liability provisions, it could be "on the hook" for most or all of the $10 million in damages suffered by the airline. Also, when the licensor agrees to unlimited liability for breach of confidentiality obligations, it should ensure that it does not agree to unlimited liability for breach of any data privacy or security terms.

Sometimes, customers will edit limitation of liability provisions to benefit the customer as well. The limitation of liability provisions will say that the customer will not be liable for any consequential, incidental, or other indirect damages resulting from the customer's breach of the software license agreement and that customer's total aggregate liability for all claims relating to the software license agreement will not exceed the liability cap. Generally speaking, licensors are willing to make the limitation of liability provisions mutual if the following additional exceptions are added:

(1) customer's obligations to pay license fees;

(2) customer's use or disclosure of the software outside the scope of the license grant; and

(3) customer's indemnification obligations (if any).

PART 2: ASSIGNMENT

KEY TAKEAWAYS

A form software license agreement should:

1. prohibit the customer from assigning the contract to any third party without the licensor's prior written consent;

2. expressly state that all assignments in violation of the assignment clause are null and void; and

3. address the licensor's rights if the customer undergoes a change of control.

Assignment provisions govern a party's ability to transfer some or all of its rights, obligations, and liabilities under a contract to a third party.
As a general rule, if the contract does not include an assignment provision or otherwise address a party's right to assign the contract, default law permits a party to assign the contract to a third party without the consent of the other contracting party. This general rule is not always true for intellectual property licenses, but licensors should take the safe approach and work under the assumption that default law (absent contract terms to the contrary) will permit the customer to assign the contract without the licensor's consent.
As a good starting point in their form software license agreement, licensors should prohibit the customer from assigning the contract to any third party without the licensor's consent.

The simple example we will use to illustrate the importance of addressing assignment rights in a software license agreement is a licensor granting an enterprise software license to a small company ("SmallCo"), and SmallCo subsequently being acquired by a large multi-national company ("BigCo"). In one version of the example, SmallCo is acquired by BigCo via an asset sale. In another version of the example, SmallCo is acquired by BigCo via a stock sale. In the last version of the example, SmallCo is acquired by BigCo via a merger.

There are many different reasons why the licensor may want to prohibit SmallCo from assigning the software license agreement to BigCo and having BigCo using the software as the new counterparty to the agreement. BigCo may be a competitor or potential competitor of the licensor. Also, the licensor may miss out on additional revenue because BigCo would now have an enterprise license in exchange for SmallCo's presumably lower enterprise license fees. Furthermore, the licensor may have difficulty complying with its other obligations in the software license agreement (ex. maintenance and support) if the other party is BigCo and not SmallCo.

If the licensor has a standard assignment provision in its form software license agreement with the "null and void" language referenced above, the licensor has largely protected itself from SmallCo assigning the software license agreement to BigCo in an asset sale scenario. An asset sale would require BigCo to become the other party to the software license agreement by SmallCo assigning the agreement to BigCo, but the software license agreement expressly says that SmallCo does not have the right to do that. Moreover, the "null and void" language says that SmallCo does not have the legal power to do that. So, if SmallCo purported to assign the software license agreement in violation of the assignment provision, the purported assignment would not be effective – BigCo would not be a party to the software license agreement and would not have new rights under the software license agreement. Also, many software license agreements would provide a termination right to the licensor if the customer purports to assign the software license agreement in violation of its terms.

In the stock sale example, it is unlikely that SmallCo would need to assign or transfer the software license agreement in any way. Rather, the ownership of SmallCo changes. The parties to the software license agreement remain unchanged in the stock sale example, meaning the parties to the agreement remain licensor and SmallCo. As a result, the standard assignment provision would likely not protect the licensor from a situation where SmallCo is acquired by BigCo in a stock sale. That language says SmallCo has no right or power to transfer the software license agreement to another party, and SmallCo has not done that in this stock sale example.

The stock sale scenario may be less concerning for the licensor because the other party to the software license agreement is still SmallCo (not BigCo). So, depending on how the enterprise license is precisely worded, expanded software usage rights to all of BigCo may, or may not, be a concern. However, whether or not the acquisition was an asset sale or stock sale, the licensor may be uncomfortable with a competitor or potential competitor owning SmallCo.

If the licensor wants to protect itself from the stock sale example where SmallCo remains the contracting party but is now owned by BigCo, the licensor should expressly state in its form software license agreement that changes of control are deemed assignments under the assignment clause, insert a "change of control" provision, or both. Change of control provisions, in essence, say that the licensor has a right to terminate the software license agreement if a certain percentage of ownership of the customer changes hands. Then, in our example, the licensor would have the option to terminate the software license agreement if SmallCo was acquired by BigCo via a stock sale.

Finally, in the merger example, SmallCo may or may not need to assign the software license agreement to BigCo or another entity. Mergers can take many different forms, and various different statutes set forth different rules on whether or not a merger involves an assignment of a contract "by operation of law." If drafted properly, the "change of control" provision set forth in the preceding sentence that provides the licensor a termination right would likely protect the licensor in the merger example. Also, the licensor should ensure that its assignment provision specifically says that SmallCo may not assign the agreement "whether voluntarily or involuntarily, directly or indirectly, or by operation of law, merger, consolidation or otherwise." This language would likely be interpreted by a court to prohibit the assignment or transfer of the software license agreement to BigCo (or any other entity) in connection with a merger of BigCo and SmallCo.

Part 3: Indemnification

KEY TAKEAWAYS

A form software license agreement should:

1. limit the licensor's indemnification obligations to third-party claims that allege the software infringes third-party intellectual property rights;
2. contain certain exceptions to the licensor's indemnification obligations;
3. set forth certain indemnification procedural requirements; and
4. contain language that states the customer's remedies set forth in the indemnification section are the customer's sole and exclusive remedies for third-party infringement claims.

Indemnification is one of the most challenging concepts to understand in software license agreements for business teams and young attorneys alike. An indemnification obligation can be a very expensive obligation as well, if triggered. As a result, the indemnification provision is probably the most negotiated provision in software license agreements.

"Indemnification" typically refers to a set of related obligations that are triggered when one party is sued by a third party that is not a party to the software license agreement. It is important to remember that indemnification, if properly worded, does not relate to claims or lawsuits between the licensor and customer, but rather the obligations that "kick in" when the licensor or customer is sued by another party.

"Market" terms dictate that licensors should include an indemnification obligation in their form software license agreement that says the licensor will defend (i.e. pay lawyers to defend) any lawsuit against the customer initiated by a third party to the extent the lawsuit alleges that the licensor's software infringes the third party's intellectual property rights. In addition, the indemnification obligation will say that the licensor is responsible for the financial fallout from such lawsuit. Although licensors are expected to provide such indemnification protection to the customer, there are numerous provisions that the licensor should include to narrow the scope of its indemnification obligations.

First, a form software license agreement should not include any licensor indemnities except for the third-party intellectual property infringement indemnity. Indemnification of claims arising from licensor's breach of the software license agreement and violation of law should not be included in a form software license agreement because many customers will not require such indemnification.

As alluded to above, the provision should clearly only apply to third-party lawsuits (not lawsuits filed by the customer) and to damages "finally awarded by a court." This language puts clear parameters around the costs, expenses, and damages the licensor will need to pay. The licensor could potentially be liable for a much broader set of costs, expenses, and damages if the indemnification is worded more broadly.

It is also critical to have a set of exceptions that says the licensor is not obligated to defend and indemnify the customer from a lawsuit if the lawsuit arises from certain occurrences that are not within the licensor's control. Typical exceptions are:

(1) customer's breach of the software license agreement or use of the software other than as intended under the agreement;

(2) modification of the software by any party other than the licensor;

(3) software that is developed in compliance with specifications provided by the customer; and

(4) combination of the software with other software and technology not provided by the licensor.

The software license agreement should also include provisions addressing the indemnification procedure. Common issues to address are:

(1) required notice by customer to licensor;

(2) licensor's sole control of the defense of the indemnified claim; and

(3) required assistance to be provided by customer to help licensor defend the indemnified claim.

A form software license agreement should expressly state that the licensor is excused from its indemnification obligations if the customer's failure to timely notify the licensor of the claim or the customer's failure to reasonably cooperate with the licensor in defense of the claim materially prejudices the licensor's ability to defend the indemnified claim.

The licensor should also address what happens to the software license agreement if the customer is sued by a third-party alleging that the software infringes such third party's intellectual property rights. The licensor will not want to continue to be obligated to provide software that is infringing (and possibly continue to incur more damages), so it should reserve the right to procure a license for the infringing software, modify the software so that it is no longer infringing, or terminate the software license agreement and refund only those pre-paid amounts for the remaining portion of the term.

The licensor should include "sole and exclusive remedy" language with these express remedies that says all of the remedies set forth in the indemnification section (defend, indemnification, and the repair, replace, or refund remedies) are the licensor's sole and exclusive liability, and the customer's sole remedy, for any third-party infringement claim. The rationale is that the licensor should not expect to get sued by the customer directly for additional damages if the licensor defended the third-party claim, paid the resulting damages, and undertook the additional repair, replace, or refund remedies.

Finally, the licensor may include a provision that says its indemnification obligations are subject to the limitation of liability provisions, which means that the licensor would only be "on the hook" up to a certain amount for all of its indemnification obligations described above. Although some licensors do include this position in their form software license agreement, most customers will expect the licensor's indemnification obligations for third-party intellectual property infringement claims to be uncapped (subject to unlimited liability). Most licensors accept this position because, in practice, licensors want to be in control of defending a lawsuit in which a third-party claims the licensor's software infringes intellectual property rights of a third party.

PART 4: REPRESENTATIONS AND WARRANTIES

KEY TAKEAWAYS

As a starting point, a licensor should include the following in its form software license agreement:

1. a narrow warranty of function (further described below);
2. a specific warranty period during which the warranty of function applies;
3. warranty disclaimer and no-reliance language; and
4. sole and exclusive remedies for the customer if the licensor breaches the warranty of function.

For purposes of this article, we will treat representations and warranties synonymously and refer to them collectively as "warranties." In essence, a warranty is a statement of a past, present, or future fact or condition with a guarantee that such past, present, or future fact or condition remains true for a certain period of time. Warranties are distinct from contractual obligations or covenants that are promises to do, or refrain from doing, certain acts in the future. A party to a contract may be liable to the other party for various damages and remedies if a warranty it provides is or becomes untrue.

Licensors with extreme leverage may be able to provide their software "as-is" with no warranties of any kind, but most licensors will be expected to provide a basic warranty of function in their form software license agreement. Some licensors may include, and some customers almost certainly will request, additional warranties addressing viruses, intellectual property rights, quality of services, data security, open source software, and other various topics. Whether or not a licensor should proactively include, or provide in response to a customer's request, such additional warranties is a fact-specific determination that depends on a variety of factors, including the license fees being paid for the software, size and sophistication of the licensor and customer, mission criticality of the software, and historical relationship of the licensor and customer. For most licensors who license software on a form software license agreement, inclusion of a warranty of function and no other warranty is a strong starting point for the licensor.

A warranty of function is a guarantee that the software will operate as promised for a certain period of time. The warranty should be limited to a guarantee that the software will perform as set forth in the licensor's technical software documentation. A licensor should avoid warranting that the software will operate as promised by its sales team, on its website, or in marketing materials.

The warranty of function should also be limited to apply during a relatively short period of time, typically two to three months to no more than one year from the date the software is made available to the customer. Software, by its nature, will eventually require updates, upgrades, bug fixes, etc. In practice, the warranty of function provides free maintenance and support services to the customer. Accordingly, if the warranty of function is not limited to a certain warranty period,

a licensor may be required to provide free maintenance and support services for the duration of the software license agreement via the warranty of function. After the specified warranty period, the licensor should require the customer to purchase maintenance and support services to obtain updates, upgrades, bug fixes, etc.

The licensor should further protect itself by including warranty disclaimer and no-reliance language. This language says that certain warranties that default law implies into every software license agreement are not applicable and that the only warranties enforceable against the licensor are the warranties expressly set forth in the software license agreement itself. If drafted properly, this language also protects the licensor from other warranties and liabilities that may be implied by default law or asserted by a customer.

Further, a form software license agreement should set forth specific remedies for the customer if the warranty of function fails during the warranty period. Typically, a licensor agrees in the software license agreement to repair the software so that it complies with the warranty of function or replace the software with new software that provides substantially similar functionality in accordance with the warranty of function. If the licensor is not able to successfully repair or replace the software, the software license agreement may permit the licensor to terminate the agreement and refund the customer all, or some portion of, amounts paid under the agreement. Importantly, if the licensor agrees to a warranty of function for a relatively long warranty period, it should avoid providing a full refund of all amounts paid as a remedy because the customer may have used and benefited from the software for months or years without issue.

Finally, a form software license agreement should contain proper “sole and exclusive remedy” language. This language says that the remedies described in the preceding paragraph are the only remedies available to the customer if the software does not operate in accordance with the licensor’s technical software documentation. So, the customer would not have the right to sue the licensor for additional damages if the software “didn’t work” during the warranty period, so long as the licensor repaired or replaced the software or provided the customer a refund as set forth above. This protection can be critical for the licensor because, without this language, the licensor may be “on the hook” for the express remedies set forth in the software license agreement in addition to money damages claimed by the customer in a potential lawsuit.

PART 5: MAINTENANCE AND SUPPORT

KEY TAKEAWAYS

A form software license agreement should clearly set forth:

1. the maintenance and support services that will be provided under the form software license agreement, if any;

2. whether maintenance and support services have their own separate term and fees or whether maintenance and support services are provided for the entire duration of the software license agreement and included in the license fees; and

3. the distinction between maintenance releases of the software that are included with maintenance fees and new software that may require new license fees under a new agreement.

In the software licensing context, "maintenance" typically refers to the licensor's obligations to provide software updates, upgrades, releases, bug fixes and patches; and "support" typically refers to the licensor's obligations to provide technical support (ex. telephone help desk). If the licensor intends to offer maintenance and support services under a separate agreement than the form software license agreement, the form software license agreement should expressly say so. If maintenance and support is provided under the form software license agreement, the following key concepts should be addressed:

(1) scope of maintenance and support services;

(2) term (duration) of maintenance and support services; and

(3) fees for maintenance and support services.

The scope of maintenance and support obligations should be clear, and the agreement should address the distinction between maintenance releases of the software that are included with maintenance fees and new software that may require new license fees under a new agreement. To help define the scope of maintenance and support services, licensors may include a service level agreement ("SLA") in their form software license agreement. The SLA is often an exhibit that sets forth the specific standards that the licensor promises it will achieve while providing maintenance and support services. A licensor-friendly SLA contains "easy to achieve" obligations, such as setting forth:

(1) *estimated response* times (as opposed to *guaranteed fix* times) after the customer contacts the licensor with an issue with the software;

(2) specific, limited windows of time during which support is available (ex. Monday – Friday 9 a.m. ET to 4 p.m. ET);

(3) the licensor's obligation to use "commercially reasonable efforts" to correct software errors (rather than guarantying successful outcomes); and

(4) "sole and exclusive remedy" language.

The sole and exclusive remedy language, in essence, says that the customer is only entitled to a small fee credit on future amounts owed under the software license agreement if the software or licensor fails to satisfy the requirements of the SLA. The customer is not entitled to sue the licensor, claim any additional damages associated with a breach of the agreement, or otherwise terminate the agreement. A licensor-friendly SLA will not contain promises that software errors will be corrected within a certain period of time or promises that updates or new versions of the software will be released on a certain schedule.

A software license agreement should also address the term of the maintenance and support services. It is common for maintenance and support services to have a separate term from the term of the software license agreement itself. For example, the software license agreement may be a 3-year agreement while the maintenance and support services under such software license agreement are year-to-year with options for the customer to not renew maintenance and support services.

The software license agreement should also be clear with respect to maintenance and support fees. A common issue in software license agreements is determining whether or not certain maintenance and support activities are included within a warranty (and related remedial provisions) at no additional charge to the customer or part of maintenance and support services that may be subject to additional fees.

In addition, a licensor should consider the maintenance and support fees when a customer "signs up again" for maintenance and support services after it previously terminated maintenance and support services. A licensor wants to avoid a situation where a customer turns maintenance and support "on and off" at the customer's discretion so that it only pays for maintenance and support when the customer needs such services. To address this situation, a licensor may require such customer to pay the licensor all maintenance and support fees as if such customer was enrolled in maintenance and support services for the entire term.

Moreover, if a customer elects not to subscribe to maintenance and support, certain representations, warranties and performance guarantees under the software license agreement arguably should not apply to that customer and its use of the licensed software. Maintenance is often used to release patches and other updates that correct errors or address other defects in software. A software license agreement should make it clear that promises made as to performance, non-infringement, errors, etc., may be dependent on the customer accepting and implementing certain maintenance and support services.

PART 6: BOILERPLATE

Every word in your form software license agreement matters.

While many businesses and their attorneys spend the majority of their time and effort on the most material provisions of a software license agreement (license grant, payment terms, limitation of liability, indemnity, etc.), often times disputes (and your likelihood of success therein) will be significantly impacted by the less negotiated terms.

A prime example of this is what attorneys call the "boilerplate" provisions of a contract. These are typically found at the end of the contract, often times labeled as "Miscellaneous", and rarely do businesses and attorneys give these provisions the attention they deserve. Below is a brief discussion of three common "boilerplate" provisions in a contract, along with some quick and easy practical tips for addressing these critical but often overlooked provisions in your form software license agreement.

Dispute Resolution (choice of law and venue)
All software license agreements should contain choice of law and choice of venue provisions. The choice of law provision tells the court the parties' intent regarding the *substantive law* that should govern any disputes arising out of the contract. Choice of forum (also known as submission to jurisdiction) provisions specify which court shall adjudicate any disputes arising out of the parties' contract and, indirectly, the *procedural law* that governs these disputes.

Choice of law and forum provisions that are not given proper attention could result in unintended issues if disputes arise out of the contract. For example, if a business agrees to a choice of forum where the entity is not qualified to do business, it may not be able to bring claims in that forum. The forum court could apply its choice of law rules, leaving the choice of governing law outside the parties' control. Additionally, a choice of forum that has no relationship to the parties or the agreement could be unenforceable in some jurisdictions. With respect to choice of law, certain states may have unfavorable substantive law. Finally, using an improper nexus phrase ("arising out of" or "relating to") when defining what claims are governed by the choice of law/forum provisions may result in too many or too few claims being governed by the choice of law/forum provisions.

When drafting your choice of law and forum provisions, remember the following:

- Pick a governing law/forum that is enforceable;
- Check the relevant substantive law applicable to the transaction;
- Consider policy preferences of the selected state and the depth of their substantive law on relevant issues that would likely govern the transaction at hand;

- Add language as needed to ensure all desired claims related to the agreement are brought in the chosen forum, not just the claims directly arising from a breach of the agreement; and
- Consider carving out claims for equitable relief; availability of equitable relief may be limited if choosing another state as the forum for all claims.

Entire Agreement Clauses

Entire agreement provisions (also known as "Merger" or "Integration" provisions) serve as strong evidence that your license agreement expresses the parties' complete agreement regarding the agreement's subject matter. Any prior or contemporaneous understandings, agreements, representations and warranties cannot serve to contradict the terms of the license agreement if it is fully integrated. While courts differ on the amount of deference given to an entire agreement provision, most find it to be sufficient evidence that there is a fully integrated agreement between the parties.

It is important to draft an entire agreement clause and with two questions in mind:

(1) is this truly the entire agreement between the parties with respect to the subject matter at hand; and

(2) is the provision appropriate in scope in order to capture the parties' intent and to be enforceable?

With respect to the first question, often times parties give little to no consideration as to whether an agreement is actually the entire agreement of the parties. Is there an NDA in place that should remain in place? If you are a large company, are you sure that there is not another agreement in place that is governing part of the same relationship or purpose? Do your diligence to make sure that the entire agreement provision is actually capturing everything that should be considered the "entire agreement" of the parties on the subject matter.

As for drafting this provision, draft entire agreement provisions with as much specificity as possible to increase their evidentiary value on the question of integration. Define the subject matter appropriately. Include all the documents (Exhibits, Statements of Work, etc.) that should be included.

Notice

Notice provisions create the rules by which notice must be given under your software license agreement. While this may seem simple and immaterial, think about all the ways you might be required to give notice under your contract, and how important notice is to such provisions. Do you want to terminate the contract? To do so will likely require notice within a specific time period. Looking to renew or not renew the contract? You better do it within the defined notice period. Need to make an indemnity claim or make the other party aware of a suspected breach? You guessed it – you will need to provide notice in accordance with the notice provision requirements.

Given how important notice is under a contract, you would think parties would spend more time reviewing the notice provision. Here are a few items to consider when drafting the notice provision in your software license agreement:

- What is the proper method for notice? Is email sufficient? Does it need to be certified mail, return receipt requested? Many old contracts still specify facsimile as an approved method of notice, but is that still appropriate?
- Who is notice supposed to go to? Officers are the most appropriate parties, but consider having a copy of the notice delivered to your attorneys.
- When is notice effective? Is it effective on the date the notice is sent or received? This will likely matter when calculating issues like termination, renewal, fee calculations, expiration, etc.

PART 7: LICENSE GRANT

KEY TAKEAWAYS

Software license grants should:

1. define the recipient(s) of the license grant (usually limiting the license grant to the customer legal entity);
2. address the non-exclusive, non-transferrable, non-sublicensable and term-limited nature of the license grant;
3. define rights included in the license grant that are not included in The Copyright Act of 1976; and
4. contain express restrictions addressing reverse engineering and creation of derivative works.

The license grant is the core provision of a software license agreement. This article focuses on end user license grants that permit the customer to use the software for its own internal business operations, as opposed to reseller or distributor license grants or license grants that permit the customer to use the software to provide services to its customers. An end user software license grant may read something like this:

> *"Subject to the terms and conditions of this Agreement, Licensor hereby grants to Customer a non-exclusive, non-transferable, non-sublicensable license during the term of this Agreement to reproduce and internally use the Software in object code form solely for Customer's internal business purposes."*

We can break the license grant above into its component parts to analyze critical aspects of a license grant. For purposes of this article, the first important part of the license grant is *"to Customer"*. This license grant is clear that "Customer," and only "Customer," is a recipient of the license grant. Importantly, "Customer" should be defined to include only one legal entity. Software licensors should avoid license grants that include numerous different parties as recipients of the grant (ex. *"to Customer and its affiliates and authorized users"*). This approach creates numerous ambiguities and unintended consequences. For example, is an affiliate of customer at the time of the license grant that later becomes unaffiliated with customer still permitted to use the software as a third-party beneficiary under the software license agreement? This is just one of many examples.

The second important part of the license grant is *"non-exclusive"*. It is very rare for a software licensor to grant an exclusive license to a customer. Accordingly, the non-exclusive nature of the license grant is rarely negotiated. Nevertheless, every license grant of any kind should address exclusivity, and it is best to be clear that the licensor is permitted to grant the same license to other customers.

Next, we move to *"non-transferable, non-sublicensable"*. Some software licenses permit the customer to distribute the software and grant other parties licenses to use the software. While such rights to distribute and sublicense are appropriate in certain agreements (ex. software distribution agreements, value added reseller agreements, etc.), this article focuses on software license agreements addressing customer as the ultimate end user. For such licenses, the licensor should ensure that the customer is prohibited from transferring, assigning or sublicensing the software and any rights to use the software. For a more detailed discussion regarding the potential issues created by a customer assigning or transferring a software license, see Part 2 of this article series.

Every license grant should address the duration of the license, which brings us to *"during the term of this Agreement"*. This provision, along with other provisions in the software license agreement, states that the customer is only permitted to use the software during the term of the software license agreement. When the software license agreement terminates, so does the customer's license to use the software. This distinction is critical because perpetual software licenses are relatively common in the software licensing industry, although not as common as they once were. So, best practice is to ensure clarity that the customer is not the recipient of a perpetual license (unless that is the intent of the licensor).

The next, and perhaps most important, part of the license grant is *"to reproduce and internally use"*. These are the verbs of the grant that state what the customer is permitted to do with the software. The most relevant intellectual property right that covers software is copyright, so license grants tend to, and should, reflect the exclusive rights granted by The Copyright Act of 1976: reproduce, distribute, prepare derivative works of, publicly perform and publicly display. The "distribute, prepare derivative works of, publicly perform and publicly display" rights are problematic for a licensor in an end user software license agreement and should not be included in the license grant. The license grant should include the right to reproduce (right to make copies) because every customer will make a copy of the software by installing the software on its computers or other hardware. Consequently, the license grant section should include language that restricts the number of copies the customer is permitted to make to protect against the customer making an unreasonable number of copies of the software.

Many license grants, like the example above, include additional verbs that do not align with the statutory copyright rights. "Use" is the most prominent example. "Use" has become so ubiquitous in the software licensing industry that many customers will expect to see it in license grants. However, courts have interpreted "use" to convey numerous different rights to customers when it is included in a license grant and not expressly defined in the software license agreement. Accordingly, licensors should take care when including "use" and other verbs that don't align with the copyright statutory rights. Licensors should consider defining "use" and adding modifiers like "internally" in the example above to ensure "use" is interpreted to mean the right to operate the software internally.

We move on to *"the Software in object code form"*. Perhaps surprisingly, the definition of "Software" is an often overlooked component of the license grant. As always, the goal of the license grant should be precision. The "Software" definition should leave no room for interpretation as to what software the customer is permitted to use. See Part 5 of this article series for a discussion on whether or not upgrades and new versions should be included within the "Software" definition. In addition, the license grant should state that the customer is permitted to receive and use the Software in its machine readable object code form and not in its human readable source code form. A discussion of the distinctions of object and source code and the considerations a licensor must weigh when granting a source code license are beyond the scope of this article. Suffice it to say, licensors should take care before granting any party a license to access and use its source code and should ensure certain restrictions are included in any such license.

"Solely for Customer's internal business purposes" is the last component of the example license grant. Many license grants use this language or similar language. Like "use" above, it is best for the licensor to be specific regarding the meaning of this language. Typically, the purpose of this language is to limit customer's use of the software for customer's own internal business operations and to prohibit the customer from

(i) using the software to provide services to its customers,
(ii) permitting its customers to access the functionality of the software, and
(iii) processing third-party data using the software.

Finally, every license grant should include a set of restrictions. This language works hand-in-glove with the license grant to provide clarity on the permitted and prohibited uses of the software. An example of standard restriction language is below.

> *"Customer shall not (i) reverse engineer, decompile, or disassemble the Software by any means whatsoever, (ii) alter, modify, enhance, or create a derivative work of the Software, or (iii) remove, alter, or obscure any product identification, copyright, or other intellectual property notices in the Software."*

PART 8: PRICE AND PAYMENT TERMS

KEY TAKEAWAYS

Form software license agreements should:

1. address both license fees and professional services fees (if applicable) with precision,
2. ensure clarity with respect to fee structure and payment terms, and
3. address taxes.

When licensing software, customers typically pay for:

(1) the license to use the software, and

(2) various professional services associated with the license to use the software.

In most cases, payment of fees by the customer is the central benefit of the bargain for the licensor. Accordingly, the price and payment terms of the agreement are critical for the licensor. Nevertheless, these sections don't always receive the attention that they deserve.

There is a great variety of pricing structures for the license to use software. Some licensors license software under annual flat fee arrangements. Others license software under variable fee arrangements based on the number of transactions, users, seats, etc. As always, the goal of pricing language should be precision. We can use a very simple example to illustrate the nuances of pricing language that sometimes go unnoticed by business teams and attorneys alike. In this example, we will assume the software license agreement provides for a usage fee based on the number of transactions. Often, one will see such license fee pricing terms in a table like the one below, with little or no additional detail.

Transactions	Price Per Transaction
0 to 500,000	**$3.00**
500,001 to 1,000,000	**$2.00**
1,000,001 to 5,000,000	**$1.00**
5,000,0001 +	**$0.50**

Various questions may come to mind when pricing terms like those set forth above are included in a software license agreement, only some of which are set forth below:

- Are there any minimum license fee payment obligations?

- Can customer simply not use the software to execute any transactions and owe $0.00?
- Does the "transaction meter" ever go back down to 0 at any point after contract signature?
- What is the definition of a "transaction"?
- If the customer uses the software to execute 3,000,000 transactions, are all 3,000,000 transactions at the $1.00 rate?
- How often are payments due?
- How soon after the date (or receipt) of the applicable invoice must customer pay the invoiced amounts?
- How are the number of transactions reported and confirmed?

Your form software license agreement should ensure that there is no ambiguity regarding any of these issues.

Payment structures for professional services (implementation services, maintenance and support services, training services, etc.) similarly have a great deal of variety. However, they tend to fall into two general categories (or a combination thereof):

(1) time and materials, and

(2) flat fee.

Both fee arrangements have advantages and disadvantages. A time and materials price structure seems fair and straightforward (customer pays for work performed), but customers assume the risk that a certain task takes significantly more time than the customer anticipated (whether fair or not). Flat fee arrangements provide certainty to both licensor and customer, but disputes over "scope creep" (i.e. what services are covered by the flat fee and what services are subject to additional fees) often arise. Again, the key is precision. Often, one will see professional services fee pricing terms in a table like the one below, with little or no more detail. Are these flat fee arrangements or time and materials fee arrangements? What happens if licensor has spent 200 hours performing implementation services and implementation is not complete? All of these issues should be addressed in detail.

Services	Fees
Implementation	**$50,000** *200 hours at $250/hr*
Training	**$5,000** *20 hours at $250/hr*

Another matter that licensors may consider addressing in their form software license agreement is payment disputes. Many form software license agreements do not address payment dispute matters because licensors don't want to proactively grant the customer the right to dispute invoiced amounts. However, proactively addressing such matters in a form agreement does have some advantages. First, the form agreement can provide a short period of time (ex. 5 days after the date of the applicable invoice) during which customer must provide formal written notice of a fee dispute. Also, the form agreement can provide that the customer forever waives its right to dispute an invoice if it fails to provide timely written notice of dispute in accordance with the agreement. The form agreement may also give licensor the right to suspend performance under the agreement, or "shut off" the software if technically able to do so, without liability if the customer disputes an invoice and such dispute is not resolved within a certain period of time. The licensor can be even more aggressive by requiring customer pay all amounts invoiced (whether disputed or not) and seek refunds of amounts successfully disputed.

Finally, a form software license agreement should always address taxes. Tax obligations, especially sales taxes and value added taxes, applicable to a software license arrangement that also contains the provision of professional services can be very complicated to determine. Software licensors would be well served to ensure that they are in compliance with all of their tax obligations. A discussion of such obligations is beyond the scope of this article, but the licensor's form software license agreement should ensure that all fees under the agreement do not include such taxes and that customer will pay all such taxes, except to the extent forbidden under applicable law.

PART 9: INTELLECTUAL PROPERTY

KEY TAKEAWAYS

Form software license agreements should:

1. accurately reflect the intent of the parties with respect to ownership of intellectual property rights (“IP”) licensed and created as part of the relationship;
2. limit the scope of promises the licensor makes with respect to the licensed IP; and
3. include appropriate restrictions on the customer’s use of the licensed software and underlying IP.

Software license agreements often contain default provisions that set forth basic rules for ownership of IP. These provisions typically provide that

(i) each party will retain ownership of their own pre-existing IP;
(ii) each party will own the IP in what they develop; and
(iii) licensor will retain ownership of all IP in the licensed software.

As is often the case, default provisions that are not carefully drafted often create unintended issues.

Default Ownership

Software license agreements should carefully establish default ownership rules that not only provide for the licensor to retain ownership of all underlying IP in its software, but also anything that it (or its customers) might develop as part of the relationship.

A software licensor often develops modifications, updates, upgrades and enhancements (“Updates”) to its licensed software that it pushes out to all of its customers. While licensors are typically fine with licensing these Updates as part of the overall license grant, rarely do licensors intend to give up ownership of such Updates to its customers. Failure to carefully draft your license agreement to retain ownership of the IP in these Updates could result in inadvertently assigning over ownership therein to a customer. In the event a customer is permitted to create modifications or additions to licensed software that should be owned by the licensor, the software license agreement should specifically address ownership of such modifications/ additions and include a present assignment (“Customer *hereby* assigns…”) in such modifications/ additions back to the licensor.

Customer Feedback

A related concept is "feedback". It is common for customers to provide licensors with ideas on how to improve the licensor's software. Licensors are often worried that they may infringe their customers' IP if they use the ideas provided by customers to improve their software.

Although that worry is largely unfounded (a fundamental tenant of IP law is that mere ideas are, generally speaking, not protectable under IP law), a licensor-friendly form should contain a provision that says the licensor has the right and license to use such feedback and ideas to improve its software and that such feedback and ideas are not its customers' confidential information.

Customized Development

There may be certain relationships where a software licensor may include customized development as part of its relationship with its customer.

For example, a customer might need certain functionality specific to its business, or may want to pay licensor to develop customized functionality specific to its needs or business. When custom development is included as part of the arrangement AND the customer is paying specifically for such development, then a customer will often expect to own the underlying IP in such custom developments.

In these circumstances, your software license agreement should include a carefully and narrowly drafted provision that

(i) identifies with specificity the customized development;

(ii) grants ownership to the customer in such customized development (but nothing else); and

(iii) limits the grant of ownership to the copyright interest in the development (vs. including patent rights, which may expand the rights you are granting to more than what is desired).

Representations and Warranties

Negotiated software license agreements almost always include representations and warranties with respect to the software's underlying IP. Typically these representations and warranties address non-infringement – that the software licensed under the agreement (and the customer's use thereof) will not infringe the IP of a third party.

As previously discussed in this Series' article on representations and warranties, it is important to limit any IP representations and warranties by

(i) making them applicable only at the time the parties entered into the agreement;
(ii) requiring the customer to use the software as required by the licensor and the license agreement; and
(iii) limiting to IP of the respective jurisdiction in effect at the time the parties entered into the license agreement.

The licensor may also consider limiting such representation and warranty to non-infringement of third-party copyright and trade secret rights. Furthermore, your software license agreement should provide the customer with specific, exclusive and limited remedies in the event of a breach of a non-infringement representation/warranty. Typically these remedies include modification of the software to make it non-infringing, procurement of substitute software that is non-infringing, or a refund of customer fees paid by the customer under the license agreement.

Lastly, the license agreement should specifically exclude the application of the non-infringement representation/warranty in the event that the customer uses the software in breach of the agreement, fails to use it in accordance with the documentation, modifies or makes any changes to the software or uses it in connection with third party technology that it is not intended to be used with or is not otherwise approved by the licensor.

Restricted Use

Finally, a form software license agreement must include restrictions on the customer's use of the software and underlying IP. Software code can be protected under trade secret, copyright and patent law, but often times parties rely on copyright law as the primary form of IP protection.

As a copyright owner, you have many rights granted to you under copyright law, including the important right of making derivative works. Arguably the law would prevent a customer from making derivative works (even absent language in the agreement), as that right is specifically granted to the copyright holder. The law does not necessarily address a party's right to reverse engineer, disassemble and reassemble, decompile or otherwise attempt to discover the source code of the software.

Therefore, it is critical that any software license agreement include certain restrictions on the customer's use of the software and its underlying IP, and such provisions should specifically restrict a customer's ability to create derivative works in the software (even though you arguably have protections under applicable law) as well as reverse engineer, disassemble and reassemble, decompile or otherwise attempt to discover the source code of the software.

Moreover, relying on copyright law protection without robust contractual use right restrictions is even more risky after the US Supreme Court's 2021 decision in *Google LLC v. Oracle America, Inc.* Many legal experts and commentators have discussed the possible impacts on copyright law's applicability to software after *Google*.

PART 10: REMEDIES

KEY TAKEAWAYS

Form software license agreements should:

1. contain express remedies with an eye towards maximizing the licensor's remedies in the event of the customer's breach and minimizing the customer's remedies in the event of the licensor's breach;

2. state that all express remedies available to the licensor are "cumulative" and not in lieu of any other remedy available to the licensor under the agreement, law, or equity; and

3. state that all express remedies available to the customer are the customer's "sole and exclusive" remedies for the applicable breach.

Put simply, remedies are the rights that are available to a contracting party as a result of the other party's breach of contract, breach of warranty, or misrepresentation. Remedies are critical for all contracting parties because they answer the following question: *"What does my client or company actually get as a result of the other party's breach?"*

Form software license agreements do not always adequately address remedies. There are many reasons for this, but one main reason certainly is that applicable law (contract common law and statutory law) provides certain "default" remedies in every breach scenario, whether or not remedies are addressed in the contract itself. So, it seems, contract drafters assume their client or company will be able to obtain their preferred remedy if the other party breaches the contract. However, that is not always the case. Accordingly, from a licensor's perspective, form software license agreements should address remedies with an eye towards maximizing the licensor's preferred remedies in the event of the customer's breach, and minimizing the customer's remedies in the event of the licensor's breach.

The three main sources of remedies in the context of a software license agreement breach are the agreement itself, contract common law, and the applicable Uniform Commercial Code. A full discussion of all types of remedies and all sources of remedies is beyond the scope of this article. So, we will focus on the most prevalent remedy available under contract common law and statutory law (damages) and certain express remedies that can be set forth in a software license agreement itself, including the remedies set forth below.

Common Licensor Remedies in Software License Agreements:

- **Termination;**
- **Injunctive relief;**
- **Price acceleration;**
- **Interest on late fees;**
- **Self-help; and**
- **"Loser pays" attorney fees.**

Most people think about damages when discussing remedies for breach of contract. This makes sense because, at the end of the day, business deals are about money, and the non-breaching party wants to be put in the same financial position it would have been in if the other party did not breach the agreement. Further, damages are almost always available "by default" under applicable contract common law (assuming the non-breaching party did, in fact, suffer economic harm), so contract drafters need not set forth a damages remedy in the agreement itself. A prudent licensor, however, will realize that it may want other rights to exercise at its option, or obligations to impose upon the customer, if the customer breaches the agreement.

Termination

To start, the licensor may want to terminate the agreement. Although most form software license agreements address termination in one way or another, a licensor may want to pay particular attention to a few termination issues. First, the licensor may want very short cure periods, or no cure periods, for certain customer breaches, namely failure to pay on time and unauthorized use of the software. A common termination provision permits either party to terminate the agreement if the other party breaches the agreement and fails to cure such breach within 30 days of being notified of the breach.

But should a licensor be required to wait 30 days if the customer is using its software in an unauthorized manner (likely infringing intellectual property rights)? Further, if the licensor granted Net 30 payment terms, a termination provision like the one mentioned above could, in essence, change the payment terms to Net 60. Second, the licensor should set forth in the agreement certain obligations that the customer must perform upon termination of the agreement. Typically, the licensor will want the customer to cease all use of the software and return or destroy, all copies of the software on the customer's systems.

Injunctive Relief

Injunctive relief is another critical remedy for a licensor, especially if the customer breaches the license agreement by using the software outside the scope of the license grant. While a licensor may also sue for damages in this context, the most immediate need for the licensor may be for a court order requiring the customer to stop the unauthorized use of the software. Although this remedy could be available to the licensor under applicable law, a licensor can benefit from a contract provision that expressly sets forth the customer's agreement that breach of certain contract provisions (ex. license grant and confidentiality) would cause irreparable harm to the licensor, that damages alone are not an adequate remedy, and that the licensor need not post bond to obtain such injunctive relief.

Price Acceleration

A licensor may also consider setting forth various remedies that focus on the customer fees or the customer's failure to pay the fees in accordance with the agreement. A licensor may require that all payments are due and payable immediately if the customer breaches the software license agreement.

Similarly, a licensor may want to reserve the ability to impose greater audit rights and otherwise alter the payment terms to provide more security in the event a licensee fails to make payments in accordance with the agreement or otherwise demonstrates signs of financial instability.

Interest on Late Fees

The licensor may also include a provision that late payments accrue interest. The interest remedy can be especially helpful to the licensor in connection with disputes over large lump sum payments. A customer that has failed to make such a payment may be incentivized to settle quickly and fairly if a significant interest charge is accruing over the course of a dispute (which, if drawn out to litigation, could last several years).

Self-Help

A controversial and risky remedy is a set of remedies referred to as "self-help." In the software licensing context, licensor self-help refers to electronic self-help where the software contains a "key" or other mechanism that (a) automatically disables the software after the agreed-upon term of the software license agreement or (b) the licensor can use to disable customer's ability to use the software if, in licensor's opinion, the customer has breached the software license agreement. If a licensor includes such disabling devices in its software, it should reserve the right to do so in its form software license agreement.

As a practical matter, these self-help remedies are very powerful for the licensor because they can help ensure that unauthorized use of the software will cease. Still, they are also very risky because exercising such measures could invite claims or counterclaims from the customer.

“Loser Pays” Attorney Fees

A less controversial and risky remedy to include in your form software license agreement is a provision that requires the “losing” party to pay the attorney fees incurred by the prevailing party in connection with any claim arising from the software license agreement. This provision reverses the so-called American Rule that requires two opposing sides in a legal matter to pay their own attorney fees, regardless of which party prevails.

The reversal of the American Rule can be of particular importance to “smaller” licensors that enter into agreements with “bigger” customers, because it may be cost-prohibitive for a licensor with few resources to pursue a claim against a customer with deep pockets. The cost of litigation is very high, so pursuing a claim against a customer with deep pockets may be cost-prohibitive for the “little guy” licensor. A “loser pays” attorney fees provision evens the playing field and incentivizes fair settlement.

If a licensor includes any of these specific remedies in its form software license agreement, the licensor should ensure that all express remedies in the agreement are “cumulative” and shall not be deemed to limit the licensor’s remedies to only those set forth in the agreement. In other words, the licensor should reserve the right to avail itself to all of the remedies that applicable law provides, in addition to the remedies set forth in the agreement.

Finally, the customer may be entitled to certain specific remedies set forth in the agreement. The licensor may set forth remedies that are available to the customer if the licensor breaches warranties. See Part 4 in this series on representations and warranties for a discussion of certain customer remedies that the licensor may provide to the customer.

Also, the customer may negotiate for additional express remedies, including refunds, price reductions, specific performance, set-off, payment withholding, cover, and source code escrow. Each of these specific remedies should be reviewed and considered in detail by the licensor. Importantly, if the licensor sets forth these customer remedies, it should set forth that such remedies are the customer’s “sole and exclusive” remedies. This ensures that the customer is not entitled to seek other remedies available under applicable law.

KEY CONTACTS

David A. Goeschel

Shareholder

david.goeschel@koleyjessen.com

402.343.3791

David Goeschel leads Koley Jessen's Commercial and Technology practice and has spent his entire career helping businesses increase their value through commercializing their intellectual property. David has extensive experience structuring and negotiating a wide range of commercial contracts and his background in intellectual property allows him to be a strategic advisor to clients ranging from start-up technology companies to Fortune 500 businesses across all industries regarding technology transactions, including software licensing, software development, and SaaS agreements.

Jack D. Horgan

Shareholder

jack.horgan@koleyjessen.com

402.343.3848

Jack Horgan counsels clients in the technology industry and works with them on their most critical day-to-day functions and agreements. Evaluating and mitigating risks in complex technology transactions is what he does best. Advising clients on both the provider-side and procurement-side of technology transactions, Jack frequently manages transactions that involve: software licensing, software distribution, software development, SaaS, PaaS, IaaS, cloud computing, outsourcing, ERP solutions, IoT, co-location arrangements, database licensing, artificial intelligence, NFTs, blockchain, mobile applications, patent and technology licensing, joint development arrangements, and IT professional services.

Want the latest technology contracts publications delivered straight to your inbox?
Sign up for our Commercial and Technology mailing list to keep up to date!