

8

The Common Public License (CPL)

CPL as a Template

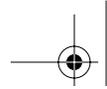
IBM has long participated in the open source community. Its involvement along with other major software companies in the Linux project, the Apache project, and many other open source activities is well known. IBM also has its own commercial open source license, the IBM Public License, one of many vendor-specific licenses approved by Open Source Initiative.

But IBM also wanted a license that was available for other companies to use, companies who were distributing open source software that might be useful for IBM and others to use or sell with no ambiguous license provisions hanging over them.

IBM's attorneys designed the Common Public License (CPL) to be a template license. Here's how the template works: The CPL applies to "the accompanying program." (CPL first paragraph.) This introduces an interesting problem: How does a license *accompany* a program?

One way is to include a license as a *shrink-wrap* or *click-wrap* license that must be acknowledged before installation or first use. As I've described, the physical process of accessing the software requires a *manifestation of assent* and is evidence that





the *accompanying* license was available to be read. But after that assent, don't the license and the software go their separate ways, one to be installed and the other to be thrown away? Is there any convenient way for someone who receives a copy of the software to remember what license applies to it?

The CPL has no specific answer, although it requires that a copy of the license *be included* with each copy of the Program. (CPL section 3.)

The technique described in the GPL, to include a licensing statement in the source code of the Program, is obviously the most convenient. Such licensing statements can be placed immediately following the copyright notice. This technique is consistent with the word *accompany* in the CPL.

Presumably that licensing notice will remain with the source code as long as the copyright notice does:

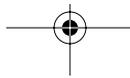
Contributors may not remove or alter any copyright notices contained within the Program. (CPL section 3.)

A Digression about Well-Written Licenses

The first open source licenses, the BSD and GPL, were written almost fifteen years ago. That was the time of UNIX. We used slow-speed modem data connections back then, before the high-speed Internet was available worldwide. Personal computers were much more primitive beasts.

Just as these fifteen years have witnessed improvements in software, so too have they produced improvements to software licenses. Attorneys are no longer struggling with unknown concepts when dealing with open source, and so, as the licenses described in this book demonstrate, competent attorneys are writing very good open source licenses.

The CPL is a very good one because it precisely describes a reasonable reciprocal bargain that promotes *free software*. It has





seven brief sections (totaling only nine pages in the Appendices to this book) in which it defines terms, grants the appropriate licenses, states the reciprocity obligations, and then deals with the commercial and legal realities of:

...The laws of the State of New York and the intellectual property laws of the United States of America. (CPL section 7.)

The CPL is fully compatible with the Free Software Guidelines, the Open Source Definition, and the Open Source Principles listed in Chapter 1.

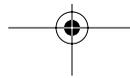
Although this license may not be appropriate for everyone (see Chapter 10) it exemplifies the important qualities of a well-written open source license. Notice that important words are not used in the CPL without a definition (with two interesting exceptions). Notice that the CPL can be used as a template between any *Contributor* and any *Recipient*. Notice that the words *shall* and *must* and *may not* always mean something mandatory, and the word *may* is always permissive.

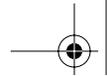
Some amateurs believe they can write open source licenses. They should first read a good license like the CPL and ask themselves if they can do as well.

Grant of Copyright and Patent Licenses

The CPL grants all the rights necessary for open source software:

...Each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free copyright license to reproduce, prepare derivative works of, publicly display, publicly perform, distribute and sublicense the Contribution of such Contributor, if any, and such derivative works, in source code and object code form. (CPL section 2[a].)





It also grants a patent license compatible with open source:

...Each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free patent license under Licensed Patents to make, use, sell, offer to sell, import and otherwise transfer the Contribution of such Contributor, if any, in source code and object code form. (CPL section 2[b].)

The patent license in the CPL is limited in much the same way as it is in the MPL. Both licenses *exclude combinations of the licensed software with other software or hardware*. The MPL's language is much more complicated, involving a positive statement and an exclusion. (See MPL sections 2.2[b] and 2.2[d].) The CPL states essentially the same limitation much more clearly in three sentences:

This patent license shall apply to the combination of the Contribution and the Program if, at the time the Contribution is added by the Contributor, such addition of the Contribution causes such combination to be covered by the Licensed Patents. The patent license shall not apply to any other combinations which include the Contribution. No hardware per se is licensed hereunder. (CPL section 2[b].)

Its clarity of language is one of the main advantages of the CPL over the MPL. But this provision still isn't very clear, is it? Just what do such limited patent licenses really mean? For the CPL, I must first define three terms:

"Contribution" means ... changes to the Program, and ... additions to the Program. (CPL section 1.)

"Licensed Patents" mean patent claims licensable by a Contributor which are necessarily infringed by the use or sale of its Contribution alone or when combined with the Program. (CPL section 1.)





“Program” means the Contributions distributed in accordance with this Agreement. (CPL section 1.)

Consider a *Contributor* who wants to add or change something in the *Program*. Assume that this new feature or function, the *Contribution* by itself, *necessarily infringes* the claims of one or more of *Contributor’s* patents. One would expect *Contributor* to license those patent claims or his or her *Contribution* could not be used. Those patent claims are *Licensed Patents*.

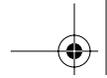
But *Contributor* intends something more. He or she wants to combine a *Contribution* with the *Program* as it was received. Assume that this *combination (Contribution-plus-Program)* *necessarily infringes* the claims of one or more of the *Contributor’s* patents. One would expect *Contributor* to license those patent claims also or the *Contributor’s Contribution* could not be used in combination with the *Program* as it was received. Those patent claims are also *Licensed Patents*.

The first *Licensed Patents*—those relating to the *Contribution* alone—are always licensed by the *Contributor* to make, use, and sell the *Contribution*.

The second *Licensed Patents*—those relating to the *Contribution-plus-Program*—are not licensed by the *Contributor* for use with *Contribution-plus-Program* unless, at the time the *Contribution* is added by the *Contributor*, the combination of the *Contribution* and the *Program* as it was received necessarily infringed.

This confusing provision has the interesting effect of excluding from the patent license, for example, a license to *Contributor’s* pending patent applications if they hadn’t been issued at the time the *Contribution* was added. Such an exclusion would not be allowed for a patent license under the open

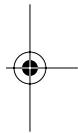




source-compatible W3C patent policy described in Chapter 13.

The *Licensed Patents* are not licensed for any *other combinations* which include the *Contribution*. This means the following are not licensed:

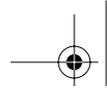
- Combinations of the *Contribution* with software other than the *Program*.
- Combinations of the *Contribution* with later versions of the *Program* unless the *Licensed Patents* were necessary for the current version of the *Program*.
- Entirely new software that embodies any of the *Licensed Patents*, even if those new programs perform the same functions as the original *Program* or *Contribution*.



Those seem to be fairly broad limitations. When described this way, are they consistent with the open source principles in Chapter 1? Why would such limitations be needed?

To understand that one must view patents from the perspective of an international company with the largest patent portfolio in the world. IBM is prepared to license some of its patents for use in an open source *Program*. Other companies and individuals will also be *Contributors*, and they too may have patents to license. The CPL guarantees that IBM and all others will (in effect) cross-license necessary patent rights to make, use, and sell the *Program including Contributions*. The entire community, including IBM and the other *Contributors*, will benefit from enhanced versions of this open source software.





But what might IBM's competitors do with IBM's patent licenses? Will they find new applications for those patents outside of the *Contribution*? Will those competitors combine *Contributions* with other software in new and different ways IBM never thought of before?

Is IBM prepared to license all those potential uses? No. IBM wants to limit its license to those specific uses and combinations that it contemplated at the time of its initial *Contribution*.

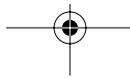
The MPL and the CPL, and most of the other corporate licenses listed at www.opensource.org, contain this kind of restrictive patent license. A licensee creating derivative works from such software—and remember, the Open Source Principles guarantee that freedom—may not exceed the scope of the initial patent license. The *freedom to create derivative works* is not absolute.

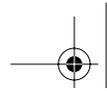
This is true under any of the open source licenses in this book—including those licenses with implied patent grants. In some situations, it may be necessary to return to the *Contributor* and request an additional patent license in order to make, use, sell or offer to sell, or import a derivative work. Any licensee of open source software who intends to create and distribute derivative works should ensure that he or she has the necessary patent licenses to do so.

Patent licenses are particularly important for companies that make, use, or sell industry standard software. The importance of broad patent rights for such software is discussed in Chapter 13.

Reciprocity under the CPL

The CPL contains a reciprocity obligation much like the one in the GPL. Software licensed under the CPL can be used





to create a derivative work (e.g., *Program*) which can then be distributed by a *Contributor* under its own license agreement. But that other license agreement is required to be very like the CPL:

A Contributor may choose to distribute the Program in object code form under its own license agreement, provided that:

a) it complies with the terms and conditions of this Agreement; and

b) its license agreement:

... iv) states that source code for the Program is available from such Contributor, and informs licensees how to obtain it in a reasonable manner on or through a medium customarily used for software exchange. (CPL section 3.)

Of course, this only pertains to derivative works distributed in *object code form*. For derivative works distributed in source code form, the CPL is more restrictive:

When the Program is made available in source code form:

a) it must be made available under this Agreement; and

b) a copy of this Agreement must be included with each copy of the Program. (CPL section 3.)

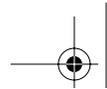
Most derivative works of CPL-licensed software are distributed under the CPL itself, not some other license made to comply with it.

Exception to Reciprocity

There is a very important explicit exception to the CPL's reciprocity obligation:

Contributions do not include additions to the Program which: (i) are separate modules of software distributed in





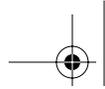
conjunction with the Program under their own license agreement, and (ii) are not derivative works of the Program.
(CPL section 1.)

Does this have the same effect as the GPL? Instead of the ambiguous language of the GPL and LGPL that causes so much uncertainty about linking, the CPL offers two simple tests for exclusion from reciprocity. Both must be true:

1. The *Contribution* must be a *separate module of software*. The term *separate module of software* is not defined in the CPL. (Neither, you will recall, was the word *file* defined in the MPL.) As with other important concepts in any technical field, *separate module of software* is a term of art in the field of computer engineering that will be defined by experts when a judge or jury needs to do so during litigation. I'm sure most readers of this book will find the concept of a *separate module of software* fairly self-evident and will know what steps to take to ensure that engineers avoid creating *Contributions* that are subject to reciprocity.
2. A *Contribution* must not be a *derivative work*. This explicit statement in the CPL, of course, is the same conclusion I drew when I discussed linking in the GPL and LGPL. Does avoiding reciprocity always boil down to avoiding the creation of a derivative work?

Anyone can get around the *reciprocity* obligation of the CPL by both (1) creating a *separate module of software* and (2) making sure that *separate module of software* isn't a *derivative work*.





As I will describe in Chapter 9, the OSL and AFL licenses do not include the first element of this exclusion from reciprocity. The MPL's concept of *files* and the CPL's concept of *separate module of software* are not included in the OSL and AFL. All one must do to avoid reciprocity is to avoid creating a derivative work.

Of course, that is not nearly so simple a change as I make it seem. I defer until Chapter 12 the technical discussion about how courts determine whether derivative works have been created.

Patent Defense

The CPL license terminates automatically under two situations as of the date that a *Recipient* initiates certain kinds of patent litigation.

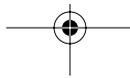
Many commercial open source licenses contain this kind of patent defense clause. A company such as IBM, with its vast portfolio of patents, wants to be able to terminate patent licenses when it is sued for patent infringement. That *defensive* use of patents is an important part of such companies' patent strategies.

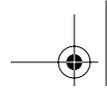
This is the first situation:

...If Recipient institutes patent litigation against a Contributor with respect to a patent applicable to software (including a cross-claim or counterclaim in a lawsuit), then any patent licenses granted by that Contributor to such Recipient under this Agreement shall terminate as of the date such litigation is filed. (CPL section 7.)

This termination provision applies to "litigation against a Contributor" and "a patent applicable to software," regardless of whether it is applicable to the software licensed under the CPL.

This is the second situation:





8 • The Common Public License (CPL)

171

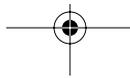
...If Recipient institutes patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Program itself (excluding combinations of the Program with other software or hardware) infringes such Recipient's patent(s), then such Recipient's rights granted under Section 2(b) shall terminate as of the date such litigation is filed. (CPL section 7.)

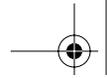
This termination provision applies to litigation against “any entity” and “a patent applicable to the Program” only.

The first provision terminates “any patent licenses granted by that Contributor to such Recipient under this Agreement.” The second provision terminates “rights granted under Section 2(b).” Curiously, there are no patent licenses granted by the CPL other than those in its section 2(b). I don't understand why the two termination provisions are worded differently in this way.

Notice also that the termination provisions apply to the patent license only; the copyright license remains. So if there are no patents that the Contributor actually licensed (i.e., the intellectual property in the software is merely copyrightable, not patentable), the termination provision doesn't apply. The CPL license provides no patent defense benefits to a licensor without patents.

Some companies do not want to in-license software under this kind of patent termination provision. Their concern is with the first half of section 7, which applies to infringement litigation “with respect to a patent applicable to software.” This is the scenario they don't like: Suppose Company A licenses its software under the CPL to Company B. Company B then accuses Company A of infringing an entirely different software patent unrelated to the licensed software. Company B's license to the software terminates.





Should Company B have accepted Company A's software in the first place? Should it ever accept the risk of relying on open source software under the CPL if by doing so it may make the rest of its software patents unenforceable against Company A?

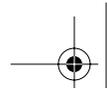
Some companies refuse to accept such license conditions. Open source projects need to decide whether such license conditions will frighten away too many prospective licensees. This may also present an opportunity for open source projects to use dual licensing, where they can offer a lower risk license alternative to such risk-averse companies—at a price. (See the discussion of dual licensing in Chapter 11.)

By the way, this situation can occur under the MPL as well. Under the defensive termination provisions in both the MPL and CPL, the licensor's patent licenses terminate if the licensee sues the licensor for patent infringement. Under both MPL and CPL, a licensee may eventually have to choose between continuing the license and suing for patent infringement.

Is that really such an unreasonable bargain? In return for accepting valuable free software from Company A, Company B must accept that its software patents are effectively unenforceable against Company A. But the software is free! Why should it not come at a price? Why isn't *reciprocity of patent licenses* a reasonable bargain?

One further comment: The OSF and AFL licenses described in the next chapter take license termination for patent infringement one step further than the MPL and CPL. In those licenses, both the *copyright and patent* licenses terminate, not just the patent licenses. Some believe that such enhanced reciprocity is justified, specifically for open source projects that don't own patents.





Defend and Indemnify

The CPL is the first major open source license to announce certain special responsibilities of licensees who are commercial distributors. It is the only place where the CPL uses the word *should*, implying that it has a philosophical or practical business objective in mind:

Commercial distributors of software may accept certain responsibilities with respect to end users, business partners and the like. While this license is intended to facilitate the commercial use of the Program, the Contributor who includes the Program in a commercial product offering should do so in a manner which does not create potential liability for other Contributors. (CPL section 4.)

What, by the way, is a “commercial product offering”? Almost certainly it is a product that one can obtain at a store or online. Does the term apply to software distributed alone, or to software that is a part of some physical commercial product? Does it require that an offer for the product be made to the public as a whole, or merely to other *Contributors* in the context of an open source development project? Does the term apply where a distributor offers software to the public at *zero price*? Does it apply when the price merely covers the costs of distribution? The CPL is silent on those questions. This is an important undefined term in the CPL. I assume this ambiguity was intentional.

The CPL seeks to protect other Contributors from the acts of a Commercial Contributor. It does this through an agreement to defend and indemnify:



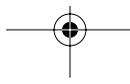


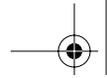
If a Contributor includes the Program in a commercial product offering, such Contributor (“Commercial Contributor”) hereby agrees to defend and indemnify every other Contributor (“Indemnified Contributor”) against any losses, damages and costs (collectively “Losses”) arising from claims, lawsuits and other legal actions brought by a third party against the Indemnified Contributor to the extent caused by the acts or omissions of such Commercial Contributor in connection with its distribution of the Program in a commercial product offering. (CPL section 4.)

This provision is important in the context of consumer protection, which is mandated in various ways by all civilized countries. The laws acknowledge that products introduced into the stream of commerce sometimes harm people, their property or their businesses. In many jurisdictions, any company responsible for introducing a product into the stream of commerce is potentially liable to pay for *Losses* caused to consumers by that product.

Under the laws of some countries, this potential liability often cannot be disclaimed regardless of what a license says. Disclaimers of liability such as the one in the CPL and other licenses simply don't apply in a commercial–consumer situation in many countries. (See CPL section 5; MPL section 9; GPL section 12; OSL/AFL section 8.) Liability disclaimers are contrary to law and voidable in some situations by injured consumers.

So who then potentially pays when consumers sue? First, an individual plaintiff may sue a company for actual *Losses* incurred. More seriously perhaps, class action lawsuits may also be filed for individually small *Losses* to large numbers of similarly placed consumers; a defendant may pay the combined *Losses* of all members of the class. Defendant companies





with deep pockets are particularly vulnerable to consumer lawsuits and to large jury verdicts for injured consumers.

Second, in the United States and in some other countries, each party in a lawsuit is generally responsible to pay its own attorneys' fees and litigation costs. This is not cheap. Merely defending such lawsuits occasionally bankrupts defendants, leaving little or no money to pay for *Losses*. Consider, for example, the cost of litigation relating to asbestos and silicone breast implants.

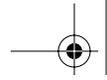
Also consider this example outside of the software field. When Firestone tires began to fail on Ford automobiles, injured plaintiffs sued both companies. It became the court's problem to determine degree of liability, if any, of each of the defendants, and then perhaps to allocate the damages accordingly. Legal procedures for analyzing degree of liability and for allocating damages vary widely around the world. Potentially, both Ford and Firestone would pay the judgments assessed against them individually, and each would pay its own attorneys' fees and costs.

But if Firestone had an obligation to *defend and indemnify* Ford, then it would be entirely Firestone's money on the line. Firestone would pay all judgments, and Firestone would pay all attorneys' fees and costs.

Broadly speaking, under the CPL, a *Commercial Contributor* must *defend and indemnify* every other *Contributor*. To the extent that IBM (and any other *initial Contributor*) allows others to be its *Commercial Contributors*, it is those other companies that will bear the burden to *defend and indemnify*.

This, of course, states the CPL's rule incompletely. The obligation to defend and indemnify applies only to the extent the *Losses* were caused by "the acts or omissions of such Commercial Contributor." This means that a Commercial Contributor may still prove it is not directly at fault. But because of its





acceptance of an obligation to defend and indemnify, it cannot rely on the other companies to step in to protect it.

The CPL obligation to *defend and indemnify* does not apply to “Losses relating to any actual or alleged intellectual property infringement.” (CPL section 4.) This is consistent with the CPL’s warranty disclaimer, which disclaims the warranty of noninfringement. (CPL section 5.)

The obligation to *defend* can be very costly for a company that is a Commercial Contributor to open source software. Paying damages for an injured consumer can require a deep reach into the bank account. An obligation to defend and indemnify every other Contributor can be particularly painful where a Commercial Contributor must pay for particularly complicated or expensive consumer injuries. Commercial Contributors need to assess their exposure carefully under the CPL before distributing software under that license.

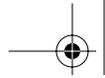
As for individual Contributors who are not directly distributing a “commercial product offering” (whatever that term really means), the *defend and indemnify* provision doesn’t apply to them.

Ownership of the CPL License

I previously wrote about ownership of *software being licensed*, but I should also comment on ownership of the *license* itself. We must also distinguish between ownership of copyrightable intellectual property that is the license and ownership of a copy of that license.

When a *Contributor* licenses software to a *Recipient* under the CPL, a new copy of the license is created binding the parties to the terms of their agreement. IBM, the author of the CPL and the owner of the copyright in that work, expressly





authorizes everyone to make such copies of the license but reserves the right to create derivative works of the license:

Everyone is permitted to copy and distribute copies of this Agreement, but in order to avoid inconsistency the Agreement is copyrighted and may only be modified in the following manner. (CPL section 7.)

The right to create derivative works of the CPL is retained by an Agreement Steward, initially IBM. Because copyright law protects the CPL license itself, you can be confident that the version of the CPL you are offered by a prospective licensor is one that the Agreement Steward has blessed.

The CPL describes what happens if the Agreement Steward publishes a new version of the CPL:

The Program (including Contributions) may always be distributed subject to the version of the Agreement under which it was received. In addition, after a new version of the Agreement is published, Contributor may elect to distribute the Program (including its Contributions) under the new version. (CPL section 7.)

Notice that a “Contributor may elect” to use the new CPL but is not required to do so.

For these reasons it is important to keep track of software not just in terms of which license you used, but which versions of the license. Proper record keeping is essential to managing open source licensing so you can know your rights and obligations.

