

软件自由法律中心 GPL 软件许可证合规指导

作者：Eben Moglen 和 Mishi Choudhary

翻译：王东芳

2015 年 8 月 21 日

目录

GNU GPL 合规的 3 个 W	3
版权和 Copyleft	3
Copyleft 的概念和许可机制	4
许可条款分析	7
GPLv2	7
GPLv3	12
GPL 特例许可	19
AGPL	20
LGPL	21
了解合规责任	26
谁应履行合规义务？	26
如何履行合规义务	27
合规的关键在于管理	28
有准备的合规原则	29
处理合规投诉	30
附件 1：源码提供要约	32
参考文献	35

本文档呈现的是软件自由法律中心（以下简称“SFLC”）及本文作者的分析观点。本文不代表 SFLC 的任何客户或机构客户的观点、意图、策略或法律分析结果。本文不构成有关任何特定事件或任一方的法律建议或意见。特定法律意见应始终由合格的法律顾问在特定情况下运用相关法律知识提供。

©2014, Software Freedom Law Center. 本文档采用创造性共同署名 - 相同方式共享 4.0 国际许可协议 (CC-BY-SA 4.0) 进行许可。

SFLC 于 2008 年发布了第一版 GNU GPL 及相关许可证的合规指导。我们认为在 GPL 软件广泛应用 6 年（包括 Android 的巨大成功，以及其他依赖于 GPL 软件嵌入设备的系统）后，是时候对我们的这份指导进行重大修订了。过去十年间自由软件在产业应用中的变化也使我们强调与以往不同的重点。本版指导包括对 GNU GPL 家族许可证¹合规义务的逐条分析，旨在提供有关许可证要求的明确法律指导。不过，在分析之前，我们首先对 Copyleft 的基本概念进行描述，这对于初次接触者来说可能更有帮助。我们还会强调软件管理，即引入、使用、修改和分发第三方软件的一般业务流程，此流程在各类规模的企业中都与业务日益相关。因此，在本指导的最后一章，我们讨论软件管理和合规之间的关系，并提供一些直白、实用的建议来帮助企业了解如何回应版权所有者的质询和合规投诉。

本合规指导的英文原文发表在：

http://softwarefreedom.org/resources/2014/SFLC-Guide_to_GPL_Compliance_2d_ed.html。

¹本档有关 GNU 家族许可证的合规事宜，即 GNU 通用公共许可证 (GPL) 第 2 版和第 3 版、宽通用公共许可证 (LGPL) 第 2.1 版和第 3 版、Afero 通用公共许可证 (AGPL) 第 1 版和第 3 版。我们已经讨论了这些许可证，并将其统称为“许可证”或“GNU 许可证”。提到特定许可证或特定版本时，我们会以缩写形式和版本号表示。根据自由软件社区的长期惯例，我们用“GPL 软件”来表示“依据 GPL 许可的软件”

作者及译者简介

Eben Moglen

美国哥伦比亚大学法律及法律史教授、GPL 许可证起草人之一、自由软件基金会 (Free Software Foundation) 首席法律顾问、理事会成员、软件自由法律中心 (Software Freedom Law Center) 创办人及主席。

Mishi Choudhary

软件自由法律中心 (Software Freedom Law Center) 法务总监，印度 SFLC 主席、执行理事。

王东芳

美国 Ladas & Parry LLP 知识产权律所中国首席代表，曾在华为公司知识产权部任国外专利流程主管、商标总监，并参与大量商业软件和自由软件法务工作。

GNU GPL 合规的 3 个 W

自由或开源软件许可证大致可分为几类。很多通常称为“宽松”或“弱 Copyleft”的许可证旨在维护开发者的自由权利。本文所述的自由软件基金会“Copyleft”许可证同样如此，但更重要的是保障了所有用户的自由权。Copyleft 许可证的目的正是确保某个程序或基于该程序的任何作品的所有用户至少拥有以下四项基本自由：

0. 自由地出于任何目的运行该程序，无需获得任何附加许可；
1. 自由地阅读、学习、理解和使用该程序源码所传授或包含的任何知识或技巧；
2. 自由地修改、改编、改进或二次使用该程序中的任何或全部代码；和
3. 自由地将该程序分享给任何人或不予分享，无论是修改版还是原版。

要理解许可证条款，最简单的方法是从“why”开始，而不是“what”。GNU 许可证由开发者及其律师撰写，以便其他开发者无需律师的任何协助就能使用，旨在确保该程序、修改版程序或包含其程序部分代码的新程序的所有用户拥有上述自由权。这些自由权的精髓在于防止自由程序的私有化改进。如果 GPL 程序被私有代码改进，那么无论如何封装或编写，都无法实现开发者想要保护用户权利的本意。Copyleft 存在的原因就是通过阻止私有改进来保护所有下游用户的自由权利。正如 GPLv3 的前序所述：

为了保护您的权利，我们需要预防他人否定您的这些权利或要求您放弃此权利。因此，您在分发或修改该软件时须承担一定责任，即尊重他人自由的责任。

本文阐释了这些责任，以及如何最有效地将其传递给用户，用户至少包括直接用户、修改者、封装者和分发者。

版权和 Copyleft

适用于软件的主要法律为版权法。Copyleft 运用版权法的功能性部分实现特殊结果（法律保护自由分享），这构成了此类许可证的核心法律原则。Copyleft 修改或像黑客一样“黑掉”旨在增强作者或发表者权利的版权法，从而增强用户的权利。

任何基于 Copyleft 程序的作品必须以与原程序相同的 Copyleft 许可证授予许可，这有时称为 Copyleft 的“遗传效应”或“共享”原则。这一“互惠”或“共享”规则既能保护开发者，避免其面临项目的“私有”软件竞争对手，也能保护用户，确保其不仅拥有当前程序版本的所有四项基本自由权利，还能够同样自由地使用该程序的所有未来改进版本。

为使用户能够自由地学习、修改和共享,所有的 GPL 程序用户在获得二进制或可执行代码后,都有权要求分发人提供完整对应源码 (Complete and corresponding source code, 简称 C&CS)。如果完整对应源码未与二进制代码一起分发,则需要附随一份书面要约 (即表示愿意提供完整对应源码的书面声明), 无论使用何种介质分发软件均是如此。要求分发者提供或表示愿意提供 C&CS 是保护用户权利的基本要求。² 满足此要求也是许可证合规的最重要组成部分。许可证中有关提供完整对应源码的要求非常具体: 二进制代码的提供者必须提供或表示愿意提供所有源码、生成文件, 以及生成用户获取的可执行二进制代码所需的其他编程素材。

在 Copyleft 许可下, 每个用户都应有权自由地修复 bugs、提升性能和二次使用代码, 否则就是剥夺了软件作者原本要让用户享有的完整软件价值。如果该许可条款不能执行, 那么私有代码改进最终将成为行业发展趋势, 而 Copyleft 程序则淹没在私有代码层之下。因此不管用户拿到软件的途径是数字网络传播, 还是计算机可读介质, 或实际产品中的嵌入, 用户都有权要求源码。软件的形式对用户的权利范围或大小没有影响。

版权法授予作者排他性的专有权利。Copyleft 的“排外”应用虽然为用户提供权利保护, 但作为版权所有人的作者或其代理人仍然是能够维护和保护用户权利的唯一人选。版权侵权诉讼是维权的最终法律解决机制, 而且版权法本身也只允许版权所有人或其代理人针对侵权行为采取行动。社区也会在开源合规方面帮助版权所有人行使权利, 但只有版权所有人或其合法代理人才可在此法律体系中提起维权诉讼。

Copyleft 的概念和许可机制

根据我们与社区和企业的合作经验, 目前存在很多对 GNU 许可证条款和核心概念的误解。在本节我们会介绍许可证的各个功能部分如何实践 Copyleft 的基本概念, 之后我们对每一个许可证的具体条款以及这些条款如何赋予特定合规义务也作出解释。只要您以许可证的本意及其核心概念为基本参照点来看问题, 即使一开始看起来随意或不合理的条款也会变得容易理解了。

“作品”和 Copyleft

版权法保护的单元是“作品”。从这个意义上来说, 许可证所述的“作品”是任何可获得版权保护或受版权法条款约束的事物。GNU 许可证则将作品的范围运用到了极致。根据 GPL 协议, 任何“作品”或“基于作品的作品”都受许可证约束 (包括完整对应源码要求), 除非适用特定排除条款。律师之间经常会就这一原则产生理论上或推断性的争议, 因为“作品”并不是计算机编程的单元。在判断某一“程序”、“目标码”、“功能”、“库”或其他任何软件单元在与其他 GPL 代码组合后是否属于“作品”的一部分时, 我们必须解答的问题恰恰无法在版权法

²批评 Copyleft 或 GPL 制定此要求“有害”的各方应该知道程序员应有权对代码进行任何处理。因此, 他们可能会将无源码要求的许可证视为“更加自由”。但是, 如果许可的目的是为了保障用户的权利, 则不可避免地会要求相关方提供完整的源代码。没有源代码, 就无法有效保障用户的自由权利。

中以相同的技术术语直接获得答案。律师在此问题上的结论往往基于根据文学或艺术版权情况制定的规则在不同的计算机编程情况下的应用。关键在于 GNU GPL 许可证旨在“保护到当地版权法认可的“基于作品的作品”的每一个版本,但仅此而已”,适用于当地版权法认可的“基于作品的作品”的所有版本(但不包括未来版本)。

对于此范围, GNU GPL 许可证表明其条款适用于所许可的作品,并延伸至“基于该程序”的所有作品。GPLv2 是美国律师为美国读者编写的许可证,其中提到:

“基于程序的作品”指该程序本身或其任何版权法意义上的衍生作品。也就是说,“基于程序的作品”指包含该程序或该程序部分内容的作品,无论是完全一致还是经过修改和/或翻译为其他语言。

遗憾的是,这种形式的解释没有帮助,反而导致数年来大家一直在讨论“衍生作品”法律原则(美国的概念)对软件所起的作用,但毫无结果(美国法院也基本上没有提供任何指导)。所以在 GPLv3 协议中,我们及我们自由软件基金会的客户决定放弃所有对美国“衍生作品”的解释,转而回到基本概念:GPL 协议涉及获许可的作品以及所有基于该作品的所有作品,其中,“基于该作品”被定义为在版权允许的情况下任何对被许可作品的修改或引入。

但是 GNU GPL 类的许可证确认:许可证以外的范围对保护开发者权利来说非常重要,正如许可证本身界定的范围对保护用户权利至关重要。当一个开发者的作品“隔离并独立”于任何 GPL 类软件代码时,则 Copyleft 义务不适用于该单独分发的作品,尽管该作品本可以与 GPL 软件结合。GPL 类许可证并非要将 Copyleft 效力延伸至超出版权范围,正像 GPLv2 §2 以如下文字所确认的:

如果识别出(其他软件代码的)部分代码并非基于“软件”生成,且有合理理由认为这些代码是隔离并独立存在的,那么本许可证及其条款不适用于作为独立作品分发的这部分代码。但是,如果这部分代码是整个作品的一部分,而整个作品都是基于 Copyleft“程序”生成,那么这个作品的整体都应合规本许可证。

GPLv3 §5 在保护开发者对此类作品的权利方面更进一步,其中有关防止 Copyleft 过度应用的规定如下:

当本许可证所涉作品(covered work)与其他隔离且独立的作品存放于同一个存储或分发介质上,但该独立作品并非所涉作品的延伸,也不会与此所涉作品结合成为一个更大的程序,且这一组合及由此产生的版权不超出单独作品给用户的使用范围和法律权利时,这种组合叫做“集合”。在集合中的所涉作品不会导致本协议适用于集合中的其他部分。

GPL 类许可证清晰阐明 Copyleft 的范围限制为版权范围内,而不是像有时暗示的那样以在“早期绑定(early binding)”编程语言中将软件代码的“动态链接

(dynamic linking)”与“静态链接 (static linking)”区分开来这种方式进行解释。偶尔会有人提出，对于与 GPL 代码“动态”链接的子程序 (subroutine)，其链接本身不属于主作品的 Copyleft 范围。这其实是误解。两个软件组件结合到一起形成一个作品时（无论是主要程序与一些子程序库的结合、两个有各自方法 (method) 的对象 (object) 的结合，还是一个程序和一个“插件 (plugin)”的结合），如果这种结合需要获得组件版权所有人的许可，但没有获得，或者虽然获得了但未合规许可证条款，则此结合会对该组件版权所有人构成侵权。在与 GPL 或 AGPL 组件结合时，该结合作为整体必须且只能合规 Copyleft 的要求，否则不能使用 GPL 组件。而无论是在分发可执行文件前通过链接器连接，还是在运行时通过操作系统内核连接以便共享库来提高执行效率，或者以运行语言“后期绑定”引用（如 Java 程序），都没有影响。³

自动向下游授予许可

每次分发 GPL 程序时，接收者会自动依据相关 GPL 条款从每个上游许可人处获得一个许可证。只要分发的作品中包含其他版权所有人的素材，这些版权所有人就都是上游许可人。这相当于是一个三维（而非线性）许可权利。该作品的每个接收者都默认或直接从每位许可人处获得许可。

这一自动向下游授予许可的机制是 Copyleft 得以运行的核心。每个许可人独立授予许可，而且每个许可人都可在遭到侵权后独立终止该许可。在 GPLv2 协议下，受到侵犯的许可会自动终止，而在 GPLv3 协议下，违反许可证协议的一方可能可在许可终止前进行补救。侵权方的下游用户只要未侵权，其获得的许可就仍然有效，因为他们是从所有的上游权利人处直接获得许可，而不依赖于分发软件的侵权方所授予的许可。基于相同原因，任何侵权人即使再获得一份该程序副本，也无法获得任何新许可权利。也就是说，一旦该程序的任何上游许可人终止了授予侵权人的许可，该侵权人就无法再自动获得新许可，即使再获得一份该软件也不行。

预防附加限制

用户的自由如果受到 Copyleft 以外的额外条款限制就不会得到保护，因此要实现 GPL 许可证的目标，绝不能放弃“不得附加限制”这一原则。GPLv2 要求只有 GPLv2 协议才能适用于基于 GPLv2 的作品。GPLv3 在第 7 条列举了几条允许的附加条款，但也仅可在特定情形下对许可证进行非常有限的变更。除了这几项例外情况外，严格适用“不得附加限制”原则。正因如此，要求同意或格式协议（包括“点击同意”这类安装程序）均违反 GPL 协议。

施加矛盾条款

“不得附加限制”原则旨在避免 Copyleft 软件在分享过程中遭到故意修改。但如果某一方受到特定限制，而该限制又无法体现在许可证中，怎么办？例如，如果 A 无权将其从 C 处获得的所有程序权利传递给 B，则 B 得到的权利就比 GPL

³请注意，LGPL 对静态或动态链接模块和所涉作品的要求不尽相同。有关详情，请参见下文。

本应赋予其的少，因为法律结论永远是 A 授予的权利不可能超出其本身所获得的权利。为了用户利益，GNU GPL 许可证解决了此难题：如果施加的法定条件（因法院裁决或接受其他机构的法定限制而受约束）阻止您授予 GPL 许可证中的所有权利，那么您就不能分发该程序。Copyleft 许可证不允许以任何外部法定条件为由不合规 GPL 条款。

此原则与“施加矛盾条款”在专利许可等方面也很重要。如果您接受的专利许可包含比相关 GNU GPL 许可证更具限制性的条款，则该专利许可中的矛盾条款将阻止您分发整个程序。这并不是基于专利权的存在或主张，也不是因为提供与 GPL 不兼容的专利许可导致产生这一规定，而是您对这些条款的接受或者施加此类规定的法院裁决造成了这一情境。

许可条款分析

我们已经讨论了为什么 GNU 家族许可证要求用户及分发者采取措施来保护下游用户的自由权利，现在我们可以集中精力关注这些要求的具体内容以及如何满足这些要求。对于此许可证家族的所有成员（其中的规定赋予了合规义务），后面的章节在如何最高效地履行合规义务方面提供了一些具体建议。在下面这节中，我们会更广泛地讨论如何通过更完善的软件管理实现合规，以及如何回应合规方面的质询和投诉。⁴在实践中，我们发现按照许可条款的顺序呈现这些信息对律师有很好的参考作用。

GPLv2

GPLv2 发布于 1991 年，在 24 年后仍然是所有自由软件许可证类型中最为广泛使用的一种。⁵GPLv2 的前序部分主要重申了许可人的社会意图。在前序部分和许可证的第一节中都可以找到如下要求：源码或二进制代码作品在分发时必须附随完整的许可证文本，以确保用户实际注意到其权利。针对此要求引发的合规投诉在所有投诉中所占的比例非常大，这主要是因为用户在购买实体产品后未获得有关存在 GPL 软件和适用许可条款的信息。最有效的合规管理方式是在编译、打包和分发软件时将许可文本视为“生成目标”，以便许可文本能够与软件的其他附随文件一起，在一个产品包中经过同样的生成和认证阶段（与二进制代码本身的生成、测试和打包方式一样）。很多与 SFLC 合作的公司都已通过类似方式在这些节点或类似节点完美地解决了合规问题。这些公司中不乏业务形式多样的大型公司，其内部有半独立的业务部门来制造和分发包含 GPL 软件的产品。

GPLv2 许可证开头提出的另一项合规义务是不得修改许可证文本。偶尔会有分发者想要删除许可证中的文字，或者对条款增加救济或其他“辅助”条款，这都会带来合规问题。与 GNU 家族的其他许可证一样，GPLv2 是自由软件基金会的版权作品，只允许原文复制。

⁴ 此处所提供的大多数材料均可在自由软件基金会制定的权威且极其实用的“GNU 许可证常见问题解答”中找到。您可以访问 GNU 网址进行查看：<https://www.gnu.org/licenses/gplfaq.html>。

⁵ <http://www.blackducksoftware.com/resources/data/top20opensourcelicenses>

第 1 节：复制和分发未经修改的程序版本

本节讨论原样复制和分发所收到程序的权利，规定如下：

- 在显著位置显示正确的版权声明；
- 保留或增加软件维保免责声明，除非您愿意提供维保服务，这当然也是可以的；
- 保留所有关于 GPLv2 的声明；以及
- 提供许可证本身的文本。

与美国流行的观点正好相反（直到 1978 年，想要获得版权保护都必须有版权声明），伯尔尼公约规定，版权的保护不需要声明和登记。在美国和世界上的大多数国家/地区，代码一旦写成，每个开发者就拥有其所编写的代码的版权。不过版权声明和在美国登记仍然具有一定法律效果。几乎所有自由软件许可证都要求增加并保留版权声明。有关维护版权声明的最佳实践详细信息，请参见 SFLC 的“在自由软件项目中管理版权信息”一文。⁶

第 2 节：修改程序和分发修改版程序

第 2 节是 GPLv2 中 Copyleft 内容的核心部分。此节直接要求修改后的程序版本必须以同样的许可条件进行分发。修改权是 4 项基本权利之一，因此 GPLv2 保护而非限制这一权利。如第 2 节所述，GPL 程序的任何修改版本事实上都属于“基于程序的作品”，只能依据 GPLv2 所包含的条款分发，不得适用任何其他条款。

第 2(a) 条要求所有修改版本必须作出标记，指明修改内容、修改日期和修改人身份的基本信息。在源码中以合理形式标记这些信息即可实现合规。这并不是要求提供源码版控制系统中的所有可用信息，也不需要替换目标码层的更改记录 (ChangeLog) 或类似文件。正确的合规方式可帮助程序员在使用或审核某个特定源码文件时了解该文件来自项目或程序软件的哪个版本，从而跟踪近期实质修改记录。

另外，为了保护他人的法律声明，第 2(c) 条规定如果修改前的程序“在运行时通常会交互式读取命令”并显示或打印出法律信息，那么所有版权声明、维保免责声明、修改标记和许可文本链接都必须在交互式应用中显示或打印出来。例如，GNU Debugger、GDB 或 GNU Zile。

GPLv2 §2 不仅包含合规要求本身，也阐明了不适用 Copyleft 的情况。§2 的另外三段在意图及实践方面对 Copyleft 的范围进行了限制。第 2 节明确规定，如果非 GPL 许可人的作者编写了隔离且独立的作品，而且该作品可能以有效形式与 GPL 软件组合，则 GPLv2 无意将 Copyleft 范围延伸至完全由该作者编写的代码。如果此类隔离且独立的作品是单独分发的，或者仅与 GPL 软件集合于同一软件分发介质（例如 DVD 或 USB 盘）上，许可证明明确表示对此类代码放弃任何 Copyleft 义务。只有当这一先前隔离的作品与 GPL 代码组合为一个

⁶<http://softwarefreedom.org/resources/2012/ManagingCopyrightInformation.html>

新的“基于软件的作品”，或该作品并非独立而是必须与 GPL 代码组合才能执行时，Copyleft 义务才适用于该新作品。⁷在集合情境中，许可证明明确规定该集合不可基于保护伞许可证发布，保护伞许可证会禁止用户按照每个软件各自的许可证行使权利。

第 3 节：以目标码形式复制和分发经过修改或未修改的程序

GPLv2 §3 规定了分发可执行或二进制版本作品的条款。尽管 §2 是 Copyleft 的核心条款，但 §3 规定了最重要的合规义务。如果没有可构建的源码，用户无法学习、理解、修改和分享软件，因此 §3 要求分发者提供源码。§3 中描述了 3 种主要合规方法：

1. 如果 GPL 程序的可执行版本可在网上复制获取，其完整对应源码也应在“同一”位置找到，以供用户复制；
2. 如果 GPL 程序的可执行版本是通过物理介质分发的，其完整对应源码应通过同一介质或者 DVD 或 USB 存储卡等介质随附于该程序分发；
3. 以前述或其他方式分发的可执行软件也可附随书面要约，即表示愿意提供完整对应源码的声明，其有效期应至少为 3 年。如果您只是单纯转发他人 GPL 代码包的非营利分发者，而该代码包仅随附书面要约，并不包含源码，那么您也可分发您所收到的书面要约。有关符合规定的书面源码提供要约示例，请参见附件 1。

什么是“完整对应源码”？首先，源码指“可用于修改作品的首选代码形式”。对于解释型计算机语言（包括外壳脚本、Perl、JavaScript 和 APL 等），可用于修改作品的首选代码形式可能只有一种。（不过请注意，压缩、最小化、模糊化以及对代码的其他编译可能导致“非源码形式”的作品产生。）

如 §3 所述，完整“对应”源码包括源码、构建脚本、生成文件、配置文件和其他必要素材，这些必要素材可将程序构建为与可执行文件完全一致的版本，也可用于修改源码并构建修改版本。缺少构建准确交付版本的任何必要元素都意味着分发的源码不完整且不“对应”。无法用于修改和重建修改版本的“对应”源码（例如因为过于模糊化而无法实际修改）也不视为“完整”。

此完整要求包括系统库文件例外规定，§3 中的相关定义如下：

通常随运行可执行程序的操作系统主要组件（编译器、内核等）一起分发的代码（无论是源码还是二进制代码），除非组件本身与该可执行程序一起分发。

⁷其中涵盖针对违规的所有主要侵权责任。间接版权责任会以其他方式附上。假设我创建了某件 GPL 作品的专有增强版并单独交付、宣传版本有效性并鼓励下游用户制作和再分发侵权组合作品（包括专有增强版和原始 GPL 作品），那么一旦发生此类主要侵权行为，GPL 许可版权持有人能够以版权侵权为由向我提起索赔。有关详情，请参阅 *MGM Studios, Inc. v. Grokster, Ltd.*, 545 U.S. 913 (2005)。

此例外规定旨在让分发者可不向下游用户交付行使权利时不需要的代码，因为可推定用户已获得作为操作环境一部分的此类代码。

未提供或未表示愿意提供完整对应源码是最常发生的违规问题。针对商业分发者的所有社区诉讼案件都与此相关。核实分发者为应对合规要求而提供的源码是否完整且对应也是合规维权中最繁杂困难的一部分。所有这些问题都源自管理不到位。分发者不保留嵌入其产品的二进制程序的源码，而且缺乏知晓如何构建软件组件的人力资源。在这之后，对于遭到修改并分发的作品，如果原版权所有人想要获得源码，各组织也不知道如何提供，或者只能提供不完整或不全面的代码，这体现了各方的无助和无知，而这两项都不是合法理由。

其实这类问题本不应或不必发生。如果完整对应源码的原始码同时也是软件生成过程中的“构建目标”，那么只要生成二进制代码就同时会产生完整对应源码，这源码就不应该丢失。只要组件能够被生成，其编程知识就不应该离开其编程组织。在网络上提供面向公众的源码也可作为产品发货或交付生产的一部分自动完成。在与全球各类公司的合作中，我们经常发现在软件开发过程中进行一些相对简单且成本较低的改进就可完全消除违规问题，而这些违规问题可能会躲过“合规”工具的审查，导致后续花费更多来进行补救。

使用书面要约虽然可推迟提供源码的时间，但从长远来看会增加合规成本。即使已停止分发产品，该要约也必须持续至少三年。这也可能使“任意方”有权要求完整对应源码，既使其并未获取二进制代码。

对于 GPL 程序的商业分发者来说，既使只是再分发他人所编的程序，也受到 GPL 条款的约束。在这种情况下，仅转发从上游许可人或供应商处收到的相同源码书面要约并不足以满足合规要求，而且也别期望用户可从您的供应商处获得源码。此时，建议您行使用户权利，向供应商索要为您提供的所有 GPL 程序的完整对应源码，这一过程最好可自动完成。收到这些完整对应源码后，将其与您的产品一起转发便可确保符合许可证规定。

GPLv3 协议中的第 6 节条款为等效条款，用于处理以非源码形式传递源码的问题。

第 4 节

第 4 节直接规定了“不得附加限制”概念，同时禁止依据非 GPLv2 协议进行分许可。此处所述禁止值得注意。尽管通常认为只有在分发 GPL 代码时才会产生 Copyleft 义务，但这种观点是不正确的。分许可是一种单独行为，也会产生 Copyleft 义务。

在本世纪二十年代前，从理论上来说几乎不可能在分许可过程中不遵守 GPLv2 协议。但现在则并非如此。“软件即服务”的“云”产品（其中虚拟服务器组按效用计费出售）已经带来了严重的违规问题。提供此类虚拟化服务器产品的公司与用户之间的协议往往声称“许可”用户使用虚拟化服务器组中的软件，而这些软件几乎总包含 GPL 操作系统组件，这就导致这些组件适用与 GPL 不一致的分许可条款。要避免这一问题，其实无需利用技术或编程。

GPLv3 协议完全禁止分许可，详情请参见下文。GPLv3 软件在“云”或“软件即服务”上的部署可在技术上完全达到合规要求，但也可能因为错误分包而导

致侵权。

GPLv2 协议的第 4 节规定，如果违背条款，则自动终止许可。此规定在 GPLv3 §8 中有所变动，详情请参见下文。因为一旦侵权，许可便会自动终止，所以要想在这之后获得重新分发权利，必须获得版权所有人的重新授权（根据 GPLv3 协议，可以自动进行此过程）。尽管社区版权持有人对于依据侵权人的请求恢复其分发权一直持一致的合作态度，但由于从事版权运营的盈利机构更多使用的是 GPLv2 协议，因此未来不可避免地会出现一些在分发权自动终止后需要支付巨额费用才能恢复该权利的案例。因为自动终止许可后会立即产生禁令，因此依赖于 GPLv2 版权营利的非社区版权所有人可能会通过自动终止许可获得不公平优势。正因如此，商业分发者现在应倾向于从上游技术供应商处获得 GPLv3 许可，而不是 GPLv2 许可。

第 5 节：无需接受许可

英国的普通法传统在 500 多年前就已将“许可”定义为产权持有人授权非拥有者使用其产权的单方面许可，这一定义一直沿用至今。在传统法学院课程中，教师通常会以宴会邀请为例来解释许可定义。例如，如果我邀请你来我家共进晚餐，但当你跨进门内时我起诉你非法入侵，那么你的抗辩理由就是“许可”，即我允许你进门的单方许可。在全球法律分析中，将许可证混淆为契约的观点普遍存在，因为在许多其他法律体系（包括那些声称源自罗马法律的法律体系）中，还未认可非契约性的单方面义务概念。（不过也不能控告受邀前往某地的人非法入侵。）GNU 许可是普通法意义上的许可：具有限制的单方面版权许可。GPLv2 §5 陈述了这一经常产生争议但又不容置疑的法律概念区别。一方无需接受许可，另一方也不寻求其接受。不仅用户无需通过点击生效或其他程序接受许可，而且许可证中授权的许可条款还禁止施加此类附加条件。根据 §4 和 §5 的规定，需要明确避免通过合同形式、接受程序、认同监测或任何其他活动将单方面许可转变成双边义务。

第 6 节：自动向下游授予许可

此节便是 GPLv2 协议中的“自动向下游授予许可”条款。每次再分发 GPL 程序时，接收者都会自动从各原许可人处获得许可，从而在合规许可证条件的前提下，对软件进行复制、分发或修改。如上所述，无需采取任何措施来确保下游接收者接受许可证条款。这样做可使每位版权持有人与每位下游再分发者在代码分发链条上都具有对应的法律关系或直接关系，从而产生两种法律效果。首先，如 §6 所述，即使直接上游供应商因违反许可证而导致许可终止，只要用户本人合规，就仍然有权对软件执行包括修改和再分发在内的所有操作。用户所拥有的许可权并不取决于上游提供者是否合规，因为用户的许可证是由版权持有人直接下发的。其次，一旦许可自动终止，即使从其他供应商处再获取软件也无法恢复权利：许可权限只能由原许可人授予，如果原许可人自动终止了许可，则任何中间许可证持有人都无法恢复这些已终止的权利。

如 §6 明确规定，许可人不可强制第三方代码接收者或分发者履行合规义务。被许可人从原许可人处获得或失去许可都取决于其自身行为。

第 7 节

此节条款用于处理以下问题：许可人因为义务冲突而不能传递许可证所保证的自由权利。合规第 7 节条款意味着，如果适用于您的条件与许可证要求相冲突，则您无法且不得使用本许可证再分发他人的代码。因此，如果您具有其他法律义务，则不得再分发该程序或基于该程序的作品，除非版权持有人授予您附加许可。

术语“适用于您的条件”对解读 GPLv2 协议第 7 节条款至关重要。如果法院裁决或其他和解协议要求您接受一些条件，则这些条件便属于“适用于您的条件”。所以，如果某强制许可或法院调解协议包含按销售单元进行许可计费或禁止向下游转售等条款，则产生第 7 节中所述的义务。接受专利许可或签署限制权利的任何其他合同也会产生相同效果。

因此，合规 GPLv2 §7 条款涉及的是法律审查，而不是技术或编程实践。据我们所知，社区组织以前并未因用户违反第 7 节条款而提起过诉讼或辩论程序，但在实践中我们经常与各公司讨论是否应取消或修订其之前已签署的某些协议条款，以免出现 §7 所述问题。

GPLv2 协议第 7 节的另一关键作用是避免在专利纠纷中达成不同的和解协议。如果某和解协议只允许和解方在无专利风险的情况下分发 GPL 软件，则不属于有效限制，只有禁止下游方再分发才属于真正的限制。毫无疑问，这样做限制了 GPL 协议所赋予的权利，导致与第 7 节条款冲突。

GPLv3

经过一年半的公开讨论，GPLv3 协议最终于 2007 年发布，其中增加了一些适用于特定情况的附加合规义务，但也使主要合规义务更易于履行，从而大大降低了因非故意侵权而造成许可自动终止的风险。

GPLv3 协议的撰写原则是避免使用专用于美国或某系统的词汇，以免引入仅适用于某系统的法律假设。例如，GPLv2 协议中的“分发”和“衍生作品”术语就曾造成分析困惑，对于理解整个许可证几乎有害无益。自 1991 年以来，随着 GPL 软件在全球广泛使用，FSF 及其律师们决定避免使用专用于某系统的表述，以免产生许可证理解方面的困惑。

GPLv3 协议中慎重使用了两个非国际版权法律语言的术语。“传播”许可证所涉作品指需要根据本地法律体系从版权持有人处获得许可才可进行的任一活动。出于个人原因使用或修改作品被明确排除在“传播”范围之外（无论各国版权法如何规定），以防止各国版权法影响如前所述的第 0 至 2 项自由权利。另一项好处是，由于“传播”包含任何特定版权制度下所授予的全部专有权利，因此自然而然地合规要求有效许可才能获得专有权的所有制度。

任何使他人能够接收或复制作品的传播活动都称为“发布”。在一般情况下，发布会产生 Copyleft 义务。“通过计算机网络与用户交流，而不涉及副本传送”被明确排除在“发布”范围之外，此规定区分了 GPLv3 协议和 AGPLv3 协议，详情请参见下文。

使用这些术语使人们只根据本地法律知识或技术事实便可对 GPLv3 协议进行权威解读。对在本国国内版权法框架下的 GPLv3 合规性存有疑问的各方只需了解以下两个问题，一个法律问题，一个事实问题：

1. 根据本地版权法，我是否需要获得许可证才可开展此活动？
2. 我开展的活动是否让任何其他方能够制作或接收此程序副本？

如果本地版权律师对第一个问题的回答是否定的，则您所开展的活动不属于传播活动，因此受到 GPLv3 协议许可，无需获得许可证。如果本地版权律师对这两个问题的回答都是肯定的，则您所开展的活动属于“发布”活动，需要履行下文 §§4-6 所述的 Copyleft 义务。

GPLv3 协议要求作品发布者提供构建程序所需的所有源码，包括支持库和编译脚本等。GPLv3 §1 中的“对应源码”定义如下：

生成、安装、(对可执行作品而言)运行目标码以及修改作品所需的所有源码，包括用于控制这些活动的脚本。但并不包括作品的系统程序库，或者在未经修改的情况下用于开展这些活动、但并非该作品一部分的常用免费程序或通用工具。例如，对应源码包括与作品的源码文件相关联的接口定义文件，以及作品通过这些子程序和其他部分之间的密切数据通讯或控制流等方式明确要求的共享库和动态链接子程序源码。

GPL 的两个许可版本都免除提供完整对应“系统库”源码的要求，通常称为“系统库例外规定”。如上文引语所述，“系统库例外规定”使分发者免于交付不必要的源码，这些源码可能随操作系统一起提供，也可能属于公开代码，或是用于生成该程序的通用编程工具。

第 2 节：基本许可

此许可证明明确表明您可以不受限制地运行该程序的未修改版本。在合规此许可证条款的情况下，即使不发布作品也可进行修改或传播。例如，如果您的许可遭到终止，则将无法制作新的修改版本来提供他人。

第 2 节明确规定此许可证禁止分许可。“软件即服务”中的轻率合同条款或其他模糊不清的内容可能并已经与本条款冲突。

第 4 节：发布完整副本

根据 §4 的条款，任何人都可无限制地发布程序源码的完整副本。发布者的合规义务仅限于附加版权声明以及随源码提供许可证文本。如果缺少版权声明，则必须添加。按照下面 §7 的规定，所有权利归属声明和附加许可（或者新添加的权利归属声明和许可）以及所有上游保修免责声明都必须保留。自动化软件管理程序能够通过简单且低成本的方式履行 §4 中规定的所有义务。

第 5 节：发布经过修改的源码版本

§5 规定的声明义务与 §4 的规定相同，不过添加了一些附加要求：提供修改声明、修改日期和修改人身份信息（如上述 GPLv2 §2(a) 要求）。

§5 重申了 GPLv2 §2 中的集合规定，如下所示：

如果该许可证所涉作品与其他隔离且独立的作品编译于同一存储或分发介质上，但该独立作品并非此所涉作品的扩展，也不会与此所涉作品组合为一个更大的程序，且这一组合及由此产生的版权不用于限制用户的使用范围和法律权利不得超出单独作品的相关权限，则这种组合称为“集合”。在集合中引入所涉作品不会导致此协议适用于该集合中的其他部分。

第 6 节：发布非源码形式的副本

第 6 节规定了分发“非源码形式”程序时的合规义务。任何不利于开发者修改的代码形式都称为“非源码形式”。因此，除了二进制代码或可执行代码，非源码形式还包括经过混淆处理、最小化、压缩或其他不利于修改的呈现形式。目前在网络上分发 JavaScript 时不合规 GPLv3 协议的情况越来越多，虽然还未导致任何纠纷，但如果一直这样发展下去，可以预见未来这一领域必定会产生纠纷。

第 6 节要求提供完整对应源码的条款与上述 GPLv2 §3 中的条款非常相似，但进行了一些更改以便更易于在现代技术条件下合规。在 GPLv2 协议下，源码可通过网络分发，不过与 GPLv2 §3 相比，“相同地点”要求在 GPLv3 中略微宽松：

如果要在网络服务器上复制目标代码，那么对应源码可位于另一台支持等效复制设备的服务器上（由您或第三方操作），但您需要在目标码旁清楚标明对应源码所在的位置。无论对应源码托管在哪个服务器上，您始终有义务确保用户在需要时可找到该源码。

此条款首次允许第三方在商业分发情况下提供完整对应源码。分发非源码形式副本的一方仍然有义务明确指明（在非源码下载内容“旁边”）提供源码所在的第三方服务器位置，并确保该第三方服务器在规定期间正常运作。

第 6 节还允许在二进制代码或其他非源码形式代码依据端到端网络协议（如 BitTorrent）分发的情况下，使用此类服务器提供源码。唯一相关要求与上文所述相同，即必须让每一端都知晓源码在服务器上的位置。

第 6 节对书面要约的相关规则进行了少许修改。根据上述 GPLv2 §3(b) 条款，书面要约必须在三年内持续有效。而除了书面要约的有效期限至少为三年外，GPLv3 §6 还要求只要该软件所嵌入的实体产品仍在提供其他备件和/或客户服务，则该书面要约就应持续有效。

第 1 节中有关对应源码“系统库”例外情况的定义与已讨论过的 GPLv2 协议中的相应定义有较大差异。新的定义如下：

可执行作品的“系统库”除了作为整体的作品外，还包括以下两种代码：(a) 打包主要部件时包含在代码包中，但并非主要部件的部分代码；(b) 仅用于让作品可与主要部件一起运行的代码，或者用于实施标准接口的代码，此类标准接口的实施源码本身已公开。在此规定中，“主要部件”指运行可执行程序的特定制操作系统（如果有）中的某个主要且不可或缺的部件（例如内核、窗口系统等），或者用于生成本作品的编译器，或用于运行本作品的目标码解析器。

此定义阐明了该程序目标操作系统的分发过程中所包含的语言处理器（编译器和解析器）不属于对应源码。如 GPLv2 协议中所述，随附于操作系统主要部件的库或者其他启动或辅助代码也不属于对应源码。同样，以公共文档形式发布的“标准接口”（以源码形式向公众公开）也不属于对应源码。

GPLv3 协议中讨论最为广泛的新要求也包含在 §6 中，即要求在“用户产品”中提供 GPLv3 代码的“安装信息”。§6 的提供“安装信息”要求旨在保护用户权利，让用户可以修改其所购买和使用的产品中嵌入的 GPLv3 代码，并运行修改后的版本。如果没有此条款，则“锁定”设备以阻止法定持有人修改软件便会损害 GPL 许可证作者希望用户拥有的自由权利。如果所有嵌入式设备都执行此类锁定，则如美国最高法院曾经说过的一样，GPL 所保证的自由权就会变成“空头支票”(*Edwards v. California*, 314 U.S. 160, 186 (1941) (Jackson, J. concurring))。

“用户产品”指：

- (1)“消费品”，即通常用于个人、家庭或日常使用的有形个人财产；
- 或 (2) 任何为了安装到住所而设计或销售的产品。

“用户产品”在通过永久转让所有权或控制权，或者基于固定期限出借或出租时，产品中 GPL 程序的非源码形式副本便已向用户发布。向用户提供对应源码时应随附足够的技术信息：

方法、过程、授权密钥，或安装和执行用户产品中的修改版 GPL 软件所需的其他信息。这些信息必须足以保证修改后的目标码不会仅因遭到修改而无法继续执行。

除非设备中采用了技术措施来阻止修改版本的安装和执行，否则此要求无关紧要。因此从 2007 年以来的经验可以看出，如果生产厂商想要锁定其产品，则应避免使用 GPLv3 软件。据我们所知，自 GPLv3 发布之日起，尚未出现任何针对违反此安装信息要求的维权行动。

第 7 节：附加条款

GPLv2 协议不允许添加任何附加限制条款，但 GPLv3 协议允许进行一些特定的有限变化，与严格的 GPLv2 协议下 Copyleft 相比，GPLv3 更好地兼容了其他许可证。第 7 节包含可用于补充此许可证条款的附加许可和非许可条款。任

一版权持有人均可在其对许可作品的贡献版本中附加 §7 中条款外的额外许可条款。GPLv3 协议声明，所有附加许可条款都必须以书面形式提供，这一点在 GPLv2 中并未规定。条款起草者认为，GPLv2 协议允许附加条款以非正式或非书面形式提供的规定会造成不必要的疑惑。如果软件发布者希望在发布软件副本前删除之前由上游作者授权的书面附加许可条款，则可将其删除。

第 7(a) 节 — 保修免责声明。版权持有人有权免除保修责任，或以与 §§15 和 16 条款规定不同的方式限制其应承担的责任。GPL 的全球适用要求起草者允许对这些条款进行必要更改，因为目前国际上尚未就保修和免责法律达成共识。

第 7(b) 节 — 保留适当的法律声明。第 7 节允许要求保留适当法律声明的附加条款，包括有关所涉作品执行结果的声明。

第 7(c) 节 — 此条款禁止歪曲原始材料，使 GPLv3 与要求以“合理”方式标记修改版本（与 GPL 的精确标记要求不同）的宽松许可证相符。

第 7(d) 节 — 限制以宣传为目的使用许可方的姓名。添加此条款是为了与禁止在未修改版本上使用许可方姓名以及其他具有广告宣传禁令的许可证相符。GPL 与“BSD 广告条款”的冲突长期以来不时为用户带来困扰，此条款的制定解决了此问题。

第 7(e) 节 — 不授予商标法权利。此条款在“不许可商标权”这一点上与宽松的许可证条款具有类似效用。

第 7(f) 节 — 允许要求作者和许可方赔偿，从而解决了与 Apache 软件许可证 2.0 冲突的两个主要部分之一。

除了上述附加条款外的任何其他非许可条款均视为 §10 条款禁止附加的“进一步”限制。如果您添加了第 7 节中所述的附加条款，则必须确保添加的条款位于相关源文件中，或在显著位置标明附加条款的位置。第 7 节中的合规义务完全属于法律审查问题，不存在技术或编程方面的后果。

示例：GNU 编译器集 (GCC) 运行时间库特例 GCC 运行时间库特例（“特例”）属于 GPLv3 协议第 7 节所述的附加许可。此特例条款的制定目的在于允许编译非 GPL（包括私有）程序，此类程序可利用特例所涉的标头文件和运行时间库，并包含编译器在编译过程中内嵌在程序目标代码中的 Copyleft 工具链中的代码。GCC 运行时间库特例中的所涉文件指任何在许可证标头处包含例外情况声明的文件。其中包括 libgcc、libstdc++、libfortran、libgomp、libdecnumber、libgcov 以及其他使用 GCC 分发的库。

第 8 节：终止许可

GPLv3 将 GPLv2 中的“如果违反条款将自动终止许可”规定改为“如果是首次违反条款，则可在一定时期内自我修正，并自动恢复许可证所授予的权利”，这一变动是新版本中对基本合规规则最重要的修改之一。

但是，如果您在发现违反许可证条款后停止了所有侵权行为，则您从某个特定版权持有人处获得的许可证能够通过以下方式恢复 (a) 暂时恢复许可证，直到版权持有人最终明确终止您的许可证；(b) 如果

在您停止侵权行为后的 60 天内，版权持有人没有以某种合理方式向您提供侵权通知，则您可永久恢复许可证。

进一步来说，如果某个版权持有人以某种合理方式向您提供侵权通知，而这是您首次收到来自该版权所有人的此类通知（针对任何作品），并在收到该通知后的 30 天内修正了侵权行为，那么您从该版权持有人处获得的许可证将永久恢复。

与 GPLv2 相比，GPLv3 规定了对发现并修正问题的激励措施，这正是新版本的一大主要优势。自动终止的许可只有在获得版权持有人同意后才可恢复，这虽然能够避免失去分发权，但也在鼓励隐藏问题，而不是解决问题。“整体分发侵权”案例指未能在分发设备内嵌的整个操作系统时提供所有 Copyleft 程序的对应源码，这会导致一次性侵犯成百上千个版权作品的版权。这种情况对侵权者恢复许可造成了巨大障碍，而如果没有恢复许可证，他们就不能分发产品。在最近的一个案例中，我们受聘对一家在软件编程合规方面相对缺乏经验的中型企业进行审计。在审计过程中，我们发现该企业已发生“整体分发侵权”情况。该企业早期产品中的软件在分发时不合规并且已导致许可自动终止，只是该企业尚未发现。依据 GPLv2 协议，即使新产品完全合规，也需要联系 100 多个软件包的所有权利人，才能确保其拥有继续分发新产品的权利。

第 10 节：下游接收者的自动许可

本节列出了本文档其他部分所讨论的无附加限制和向下游接收者自动授予许可原则。向分发程序的接收对象主张专利权属于附加限制，是第 10 节中所明确禁止的。

兼并和并购 第 10 节还阐明，无论是通过收购资产还是转让控制权来收购企业，收购方都是被收购方的下游用户。这就产生上游版权持有人自动向下游授予的新许可、被收购企业所做所有修改的许可，以及向新的下游收购方提供源码的权利。根据我们的经验，在大多数并购情况下，处理这些许可和权利的成本都极高，而且效率低下。其实，只要收购方在收购项目完成前声明豁免和免除被收购方的 GPL 合规责任，就可解决这个问题。如果收购方已有足够完善的软件管理系统，则在整个交易过程中获得的所有软件都可纳入已购买软件的标准管理流程，这样并购后的新实体就可以自动完成合规。

第 11 节：专利

GPLv3 提供以下两项专利承诺：

1. 禁止向下游分发对象主张专利权：GPLv3 §10 明确规定不可施加附加条件来要求被许可方的直接分发对象接受专利许可或支付专利许可费。此款制定了有关 GPL 软件专利权用尽的统一规则，不考虑任何特定法律体系或区域法律下的国内专利法。

2. 贡献者版本中的专利许可：第 11 节指出，任何向 GPL 软件贡献代码的人都需要将其中涉及的专利许可授予用户。此规定旨在防止社区内的成员以激进方式向用户主张自己所修改的代码部分的专利权，即防止社区“内部人员叛变”。如果引入修改代码可导致修改后的软件构成侵犯贡献者专利权，贡献者会将原软件中的专利权许可授予所有后续用户、软件修改人或软件衍生作品的修改人，但不会授予代码修改部分中属于他人的专利权许可。此条款还规定，“贡献者版本”完成后获得的专利权也会在版本获得或完成时授予用户。如果某个拥有众多此类专利权的公司收购或聘用了程序修改者，则根据本条款，收购者已获得和后续获得的专利权也会自动传递。例如，微软公司收购诺基亚后，微软基于诺基亚曾修改的任何 GPLv3 程序的任何贡献者版本当前或以后获得的此类专利权都会自动向下游授予许可。微软收购诺基亚导致 GPLv3 程序的微软专利诉讼量整体下降这一现象至今未在行业内得到充分关注。

第 11 节还努力减少相关方共谋实施歧视性专利许可而破坏 GPL 代码共有权这一现象。如果 A 从 B 处获得了专利许可证，而该许可证只对 A 有利，却对 A 的客户或分发对象不利，则 A 就将来源于 B 的专利风险转移给了其他人。在很多情况下，这种结果是可接受的。但如果只有 A 可分发该程序、获得该程序的许可证并基于此许可证分发该程序，则 A 实际上是在帮助 B“将该程序私有化”。第 11 节提出了一些 A 可执行的方案，包括放弃该许可证或将该许可证适用对象扩展至相关各方。但最合规且成本最低的方案是：只需确保该程序的源码可永久通过第三方获得，以便希望自担风险开发、商业部署或再分发代码的人可以获得源码。

除了此“下游保护”规定，条款 §11 还采取另外两项措施，来防止滥用歧视性专利许可策略：

如果在某项单独交易或协议下，您获得一份所涉作品的发布权来发布或传播该作品，并向收到该作品的某些组织授予了专利许可，以授权他们使用、传播、修改或发布该作品的特定副本，则您所授予的专利许可会自动延伸给接收该作品以及该作品的所有衍生作品的所有接收者。

例如，如果 B 公司提供来自 A 公司的大量折扣券，这些折扣券可使第三方拥有 A 分发的 GPL 程序，并且只在第三方通过 B 购买 A 公司产品或使用 B 发放的折扣券购买 A 公司产品时，才会将该程序的专利权授予第三方。第 11 节事实上剥夺了 B 在交易中的价值，因为专利许可是自动扩展至所有其他用户的。经验表明，B 将反过来声称折扣券不适用于所涉软件包中的任何 GPLv3 程序。但这样做行不通。

如果某一专利许可的范围不包括或禁止实施此许可证明明确授予的一项或多项权利、或以不实施一项或多项此类权利为前提条件，那么该专利许可具有“歧视性”。如果您与从事软件销售业务的第三方通过协议约定您支付给第三方的费用基于您发布作品的范围，第三方向从您接收作品的任何用户提供歧视性专利许可，该许可 (a) 与您

发布的所涉作品副本（或基于这些副本复制的副本）相关，或者 (b) 主要用于涵盖所涉作品的特定产品或软件集合，或者与该产品或软件集合相关，则您不得发布该所涉作品。如果您与第三方签署的这一协议或获得前述专利许可的日期早于 2007 年 3 月 28 日，则不受此条款的约束。

此条款的引入旨在防止在自由软件基金会组织公开讨论实施 GPLv3 期间出现更多的公司间特定协议。据我们所知，目前尚未出现有关此条款的合规问题。

专利主张和报复：如果分发者或贡献者针对自己分发或贡献的作品主张专利权，GPLv3 规定了具有限制的专利报复方案。§§8、10 和 11 节共同描述了防御性的专利许可暂停和终止条件。

第 8 节的终止条款明确表明，如果被许可方违反 GPL 条款规定，则除了版权许可外，贡献者还可终止其根据 §11 条款授予该许可方的任何专利许可。因此，如果下游被许可方提起违反 §10 的专利侵权诉讼，则贡献者可以终止其授予该下游被许可方的专利许可。

第 12 节：不得放弃他人的自由权

GPLv3 §12 与 GPLv2 §7 的功能相同。GPLv2 条款中所举的示例不仅没有将问题解释清楚，而且让人更加困惑，所以被删除了。从所有其他方面（包括合规结果）来看，两个版本中的这两个条款具有相同效果。有关详情，请参见上文。

第 13 节：与 GNU Affero 通用公共许可证一起使用

此节专门许可将 GPLv3 作品与 AGPLv3 作品相链接或相组合。下文将就 AGPLv3 讨论这一许可的合规结果。

组合后的作品整体适用 GPLv3 协议，但如果该作品是通过网络使用的，则应适用 AGPLv3 协议。

无论您多么喜欢 GPLv3，都不能依据 GPLv3 发布或修改 GNU AGPL 代码，反之亦然。但是，您可以将这两种许可证下的单独模块或源文件组合入同一个项目。

当 GPLv3 代码与 AGPL 代码通过链接形成组合作品后，每个许可证下的 Copyleft 义务不可扩展至另一方。也就是说，进行此类组合后，Affero 的许可义务只适用于引入组合作品的 Affero 代码部分。如果在收到此类组合作品后，不希望合规 Affero 的许可义务，则需要删除其中的 Affero 代码。

GPL 特例许可

GNU 类路径和 GPL 链接例外情况

GNU 类路径是依据 GPL 许可的一组基本 Java 编程语言库。此类路径依据 GPL 进行许可，但有一个特例：

将此基本库动态或静态链接到其他模块即基于此库创建一个组合作品。因此，GNU 通用公共许可证的条款与条件适用于整个组合作品。

作为特例，此基本库的版权持有人允许您将此库与适用任何许可证的独立模块相链接以生成可执行程序，您可以随意选择该可执行程序适用的许可证，只要同时满足该独立模块的许可证条款和条件即可。独立模块指不以此库为基础的模块或衍生模块。如果您对此库进行了修改，此特例将扩展至您修改后的库版本，但这不是必须做法。如果您不愿意扩展此特例，可以将此例外声明从您的版本中删除。

AGPL

是否将 Copyleft 概念延伸至网络上自由软件所交付的服务是个复杂问题，自由软件社区已经就此讨论了很久。用户基本自由权中的第 0 条要求允许用户自由地出于任何目的运行程序，这当然也包括向他人提供计算服务。基本自由权中的第 2 条要求尊重私有修改权。这两项要求结合起来就是任何人都能够以提供计算服务为目的，运行私有修改的 GPL 程序。

但当服务提供方利用修改后的自由软件提供服务时，开发用于提供网络服务（尤其是提供网络应用基础架构服务）的软件的程序员希望让用户也享有与修改者同样的权利，而该权利是最初开发者就授予修改者的。联合或“自由”服务平台均受益于此许可模式。

自由软件基金会并未努力解决单个许可证的内部冲突，而是在较长时间里尝试不同的许可证来使此类“Copyleft 服务”软件可与 GNU GPL 达到部分兼容。目前已发布并使用的两个此类许可证分别是 AGPLv1 和 AGPLv3，这两者的体系结构不同，但意图相同。⁸AGPL 中的“A”代表“Affero”，这是 FSF 董事 Henry Poole 开创的某个软件项目的名称。AGPLv1 由 FSF 编写，发布于 2003 年，专门用于 Affero 项目。AGPLv3 则与 GPLv3 同时设想和起草，旨在进一步统一这些许可证之间的关系，并允许用户在 AGPLv3 项目中使用 GPLv3 代码。

在 GPLv2 许可证中，“分发”和分许可是产生 Copyleft 义务的条件，GPLv3 中又增加了“发布”条件，而 AGPL 许可证的共同特点是除了这些条件外，还另外规定了一个单独的产生条件。AGPL 许可证的 Copyleft 范围与各版本 GPL 的 Copyleft 范围相同。只有导致提供对应源码义务的条件和许可证文本有所变化。

AGPLv1

AGPLv1 在 GPLv2 文本上增加了一项附加要求，参见 §2(d)：

如果您获得的软件用于通过计算机网络与用户交互，且该版本可让用户请求获得该软件的完整源码，那么您不得从您修改的软件版本或基于该软件的作品中删除此附加服务，并且必须向通过计算机

⁸<http://www.blackducksoftware.com/resources/data/top20opensourcelicenses>

网络与该软件交互的所有用户提供平等机会，让他们能够请求通过 HTTP 立即传输修改版本或其他衍生作品的完整源码。

此要求与 GPLv2 §2(c) 中的“在交互使用中保留声明”要求类似，是 Bradley Kuhn 在担任自由软件基金会执行董事期间首创的。此要求的合规方法很简单：如果“用于通过计算机网络与用户交互”的软件中存在请求服务器端源码功能，不将其删除即视为合规。

AGPLv3

AGPLv3 在 GPLv3 的文本上增加了一项附加要求，此要求包含在 AGPLv3 §13 中。与 AGPLv1 类似，AGPLv3 也没有对 Copyleft 的范围进行任何修改。在决定按照 AGPLv3 §6 提供或表示愿意提供何种“对应源码”时，只需了解同一程序以非源码形式通过物理介质分发时需提供何种源码即可。

AGPLv3 的附加激发条件如下：

如果您修改了该程序，修改后的程序必须明确地向所有交互用户提供机会，来使其能够通过计算机远程网络（如果您的版本支持此类交互）接收该版本的对应源码，即通过一些方便用户复制软件的标准或常用方法在网络服务器上免费提供对应源码。此处所述的对应源码包括按照以下规定纳入 GPLv3 所涉作品的对应源码。

合规要求：如果您修改的软件版本中包含通过远程网络与用户交互这一体系结构，就必须为用户提供机会，以便其能够通过网络接收服务器端的对应源码。如果对应源码档案本身是编写软件过程中的“生成目标”（对于所有 Copyleft 作品，均建议如此操作），那么有关早期绑定语言的合规十分简单，运行中的软件在收到相关请求时将此源码档案（也可能包括同一系统中其他运行程序的源码文件档案）作为固定数据输出到通信流中即可。这种技术性的合规方法也解决了以早期绑定编译语言编写的软件的“Copyleft 范围”问题：如果运行时程序的组件是在生成时就已决定的，那么用于生成该程序的代码在去掉系统库和其他任何特例代码后即对应源码。

总之，GPLv3 §13 和 AGPLv3 §13 ¶2 规定了组合 GPLv3 许可代码和 AGPLv3 许可代码的兼容结构。GPLv3 允许其许可证下的代码与 AGPLv3 代码相组合，而且每个部分适用各自的许可证。如果没有各自版权持有人的允许，AGPLv3 代码不能在 GPLv3 下重新许可，反之亦然。在这两种许可证代码组合成的任何作品中，GPLv3 源码都属于对应源码的一部分。

LGPL

LGPL 条款适用于 GNU C 库等其他作品，该条款有时候称为“弱 Copyleft”许可证，因为根据 LGPL 条款许可的代码可与非自由软件代码组合，这一用法并不少见。虽然当前使用的两种 LGPL 版本在内容和结构上都具有实质不同，但

都属于强 Copyleft 许可证，只是在涉及链接时允许较多特例。这一区别经常遭到误解，而理解这一点就可确保恰当合规许可证条款了。当 LGPL 代码与非 Copyleft 许可证下的代码组合时，从不会将其置于新条款或替代条款下。虽然允许组合，但该程序的 LGPL 组件仍需合规版权持有人为其选择的条款。在文件级别上，这与 Eclipse 类许可证的性质并不相似。Eclipse 许可证只能保证按照 Copyleft 条款分发源码，而二进制代码则由再分发者自行决定适用何种许可条款。各版本 LGPL 均旨在确保下游用户享有组合作品中 LGPL 部分代码的相关权利，包括源码形式和非源码形式。

LGPLv2.1

LGPLv2.1 本身属于 Copyleft 许可证，§6 是 LGPLv2.1 允许非 Copyleft 组合的一项特例规定。从分析角度看，独特之处在于 LGPLv2.1 §6 这一特例导致此 Copyleft 许可不属于 GPL。LGPLv2.1 §§1-5 无论在哪方面都与 GPL 的任何版本不符。

第 0 节：定义和许可范围 LGPLv2 协议的第 0 节明确规定，将程序“直接”翻译为另一种 LGPL 程序或例程库的编程语言是对该许可程序进行修改，属于创建新的修改版本。GNU GPLv2 和 GNU GPLv3 中不包含该定义。

第 2 节 第 2(a) 节明确规定，如果被许可的作品是软件库（在 §0 中定义为“软件功能集和/或数据集，用于与使用这些功能和数据的应用程序相链接以形成可执行文件”），则仅在修改后的版本也为库的情况下，才可分发该修改版本。因此，LGPLv2.1 不允许将代码从库中改为放到非库的修改版本或基于该库的作品中。第 6 节未提供如下规则的特例规定：作品可由 LGPL 库的修改版本与其他代码组合而成，但 LGPL 代码必须继续构建为库，并且许可证条款继续适用于该库。

第 2(d) 节规定修改后的例程库版本无需与库函数无关的非参数全局数据便可实现库功能。这表示在修改库时，不得将密钥、令牌、表格或其他非函数性限定数据作为调用和接收库中程序服务的必要条件。第 6 节未将这些要求排除在外。实现合规要求避免此类伎俩，这一点自 1999 年发布 LGPLv2.1 以来已成为共识。

第 2 节的要求非常严格，单独来看会侵犯用户的自由权。但 LGPLv2.1 §3 允许用户将此许可证下的作品改为适用 GPLv2 及其后续版本的许可证。通过此方法，不想按照 LGPLv2.1 要求使用代码的用户有权自行决定适用 GPLv2 还是 GPLv3。

许可证第 6 节包含一项功能性的反锁定条款，要求任何作为私有或非 Copyleft 软件一部分的 LGPL 代码都必须能够与修改后的库再次链接，或允许重新安装修改后的库。

如 GPLv2 所述，第 2 节重申：

这些要求适用于修改后的整个作品。如果其中部分作品并非基于库衍生所得，且有合理理由认为该部分作品属于独立且分离的作品，则在您单独分发该部分作品时，本许可证不适用。但如果整个作品都是基于该库的衍生作品，且您将该部分纳入整个作品中进行分发，则

整个作品的分发都必须适用本许可证，而且对其他被许可人的许可延伸至整个作品，无论每部分的代码是谁编写的。

鉴于 §2 的上下文，显然不能将此条款解读成：在任何情况下都限制 §6 下的代码组合特例，但 §2 中的此规定重申了 §6 不适用于修改后的 LGPL 作品本身。

第 4 节 第 4 节涉及分发非源码形式的未修改作品或 §2 允许的修改版作品。合规此节的要求与合规上述 GPLv2 的相应条款要求一致。

第 5 节 第 5 节讨论版权法应用于软件作品时所引起的模糊性问题：

如果某程序未包含库中任何部分的衍生作品，但该程序用于通过编译或与库链接而与库一起运行，则此类程序称为“使用库的作品”。单独来看，此类作品不属于库的衍生作品，因此不包含在此许可证范围内。

不过，“使用库的作品”与库链接后会创建一个可执行文件，由于此文件中包含库的部分内容，因此属于库的衍生作品，而不是“使用库的作品”。也就是说，该可执行文件受此许可证的约束。第 6 节正是适用于分发此类可执行文件的条款。

当“使用库的作品”使用库中标头文件的素材时，虽然源码不是库的衍生作品，但作品的目标代码可能是库的衍生作品。事实是否真时如此？在作品无需与库链接或作品本身就是库的情况下，理清此问题尤为重要。相关法律并未明确定义这一点。

相应地，为避免产生此类模糊性问题，§5 继续指明：

如果此目标文件只使用数字参数、数据结构布局和存取元、小型宏命令和小型内联函数（长度不超过十行），则使用此目标文件不受限制，无论在法律意义上此目标文件是否为衍生作品。（包含此目标代码和库内容的可执行文件仍受第 6 节条款的约束。）

如果作品为库的衍生作品，则可按照第 6 节的条款分发作品的目标代码。无论包含此作品的任何可执行文件是否与库本身直接链接，均受第 6 节条款的约束。

此条款的目的显然是为了在遇到模糊情况时放弃 Copyleft，同时允许 §6 下的组合作品特例。此规定允许用户按照 §6 中规定的例外情况组合作品，对 Copyleft 的界定十分模糊。据我们所知，目前还没有因第 5 节条款引发的公开 LGPLv2.1 维权行动。

第 6 节 第 6 节规定了组合作品的例外情况，此规定经常被误解为一项“弱 Copyleft”许可，但实际上并非如此。此条款允许用户组合许可作品或基于该作品的作品：

以制作包含库部分的作品，并在分发该作品时自行选择适用的许可证条款，只要该许可证条款允许用户修改作品供自己使用，且允许通过反向工程定位修改内容。

这些针对组合作品的规定要求分发者所选择的许可证不能禁止用户修改整个作品，也不能禁止客户实施反向工程来定位修改内容。从过去维权的实际情况来看，这意味着分发者所选择的许可证不能禁止用户对组合作品中的 LGPL 代码进行修改或反向工程来定位修改内容。

第 6 节还规定：

分发的作品必须随附显著声明，以告知用户作品中包含 LGPL 库，且该库及其使用受此许可证约束，同时还必须提供此许可证文本。如果作品会在执行过程中显示版权声明，则必须在其中增加该库的版权声明，并提供链接，指引用户找到许可证文本。

此规定与 GPLv2 和 GPLv3 中的对应条款并不完全相同。合规此规定要求分发者在涉及“版权所有人”、“许可证”或“关于”的交互程序界面上采取略为不同的声明方式。

此外，§6 规定用户应提供或出示组合作品中 LGPL 部分的源码，或者利用“共享库”机制允许使用修改版本动态替换许可作品。

对于提供源码的情况，§6(a) 的规定与 GPLv2 和 GPLv3 不同：

(您必须) 随作品分发机器可读的完整对应库源码，包括作品中使用的任何代码修改(按上面第 1 节和第 2 节的要求必须分发)。如果作品是与库链接的可执行文件，则必须以目标码和/或源码形式分发机器可读的整个“使用库的作品”，以便用户可以对库进行修改，然后再次链接来生成包含修改库的修改版可执行文件。(可以理解为，如果用户修改的是库中的定义文件内容，就不需要通过再次编译应用程序的方式使用修改后的定义。)

源码不仅应完整、对应，而且提供的方式必须实现以下结果：用修改后的 LGPL 作品替代之前提供给用户的版本，仍然可重新生成整个组合作品的修改版本。例如，当 LGPL 代码与非 Copyleft 可执行代码静态链接时，分发的源码必须也包括足够的信息，以便拆分可执行文件，并重新链接到库的修改版本。

§6(c) 规定源码可按书面要约的形式提供，而不一定要直接提供；也可按照 §6(d) 条款通过网络分发。此外，§6(e) 规定分发者可“核实”用户是否确实收到相关源码，或者至少可核实分发者是否已向特定用户发送相关源码。这样做显然是为了避免在“整体分发”情况下用户要求重复分发。

如果组合作品的分发者不打算分发或提供 LGPL 组件的源码，则 LGPL 作品必须以“共享库”机制单独打包分发(遵循单独分发条款中有关源码分发的要求)，即：

- (1) 软件在运行时使用已存在于用户计算机上的库的副本，而不是将库的功能复制到可执行文件中；(2) 在用户安装的库经过修改之后，软件仍然运行正常，只要修改后的库版本与原库版本接口兼容即可。

总而言之，这些条款指：

1. 如果您创建的程序通过共享库机制链接到某个根据 LGPLv2.1 单独分发的作品，则可根据您选择的许可证条款分发该作品，而且无需发布 LGPL 作品的源码。如果您将该库与您的程序一起分发，或者在另一情况下将该作品单独分发或作为另一产品分发，则必须按照 LGPLv2.1 或 GPLv2+ 条款（可自行选择）分发对应源码。
2. 如果您选择将自己的程序与一个 LGPL 作品静态链接或组合，仍可自行选择许可证，但必须满足 LGPL 许可证中有关用户修改、反向工程和调试的限制条款，并且 LGPL 组件本身仍受 LGPL 条款约束。您必须表示愿意提供或直接提供 LGPL 组件的完整对应源码。您所提供的源码必须足以让用户利用修改版 LGPL 组件重新生成组合作品。

LGPL 合规问题较为普遍，可能是因为 LGPL 条款易于误读的原因。在实际工作中，我们代表一些许可方以用户名义维权。公开的 LGPL 纠纷并不多见，因为礼貌但坚决要求各方合规许可证条款时，各方都会选择合规。

LGPLv3

LGPLv3 Copyleft 许可证提供 GPLv3 组合作品特例，旨在修正 GNU 许可证家族的结构计划。因此，LGPLv3 是针对上述 GPLv3 §7 的附加许可。

第 0 节：其他定义 第 0 节将适用的“库”定义为包含一个或多个接口的作品，对该作品的“使用”是通过“应用程序”实现的。子级别定义“视为”对接口的使用。“应用程序”是使用一个或多个“库”接口的其他程序代码，可与接口另一端的代码组合以生成“组合作品”。

第 1 节：GNU GPL 第 3 节的例外情况 第 1 节从限制访问其他 GPLv3 §3 版权作品的“有效技术措施”中排除了干扰使用 LGPLv3 代码的做法。

第 2 节：发布修改版本 第 2 节进一步规定不得将密钥、令牌、表格或其他与功能无关的非函数性限定数据作为修改库的必要条件。这被称为“善意努力”要求，但未按通知予以纠正会视为缺乏善意的有力证据。如 GPLv3 §7 所规定，通过删除附加许可来使用 GPLv3 条款是达到合规的另一方法。

第 3 节：目标码引入库头文件材料 第 3 节声明了适用于版权范围内所有此类使用的规则（用户拥有酌情决定权），完全摒弃有关使用 LGPLv2.1 §5 所涉头文件和其他此类库资料的含糊不清部分：明确声明程序中使用库，并随作品一起提供 GPLv3 和 LGPLv3 许可证文本。

第 4 节：组合作品 第 4 节是 LGPLv3 的核心条款，针对组合作品许可。本节重申了 LGPLv2.1 §2 的许可限制规定，旨在阐明有关组合作品的许可条款不得禁止用户更改库代码或借助任何必要反向工程手段来定位此类库代码更改。

第 4(d)(0) 节包含提供最少对应源码要求，最少对应源码指“排除组合作品源码中此类代码后的那部分代码：单独来看是基于应用程序而不是 [库的] 链接版本。”综合以上 LGPLv2.1 §6 所述，用户也可按照 §4(d)(1) 下的“共享库”机制来分发源码。

此外，§4(e) 要求向用户交付“安装信息”以便按照 GPLv3 §6 将库的修改版本安装在“用户产品”中。鉴于 §4(d)(1) 下未界定库的最少对应源码，§4(e) 重申仍需按照 GPLv3 §6 的条款交付“安装信息”。

如前所述，GPLv3 的所有其他条款均有效，且不因 LGPLv3 中的例外许可而排除在外。

了解合规责任

在前面的章节中，我们解释了自由软件开发人员为何使用 GNU 家族的 Copyleft 许可证，并从许可人意图和目标方面详细解释了这些许可证的条款。接下来，我们要审视 Copyleft 项目中“下游”用户各方的合规责任，并提出一些有关解决此类问题的通用建议：用户要求提供源码时，应如何回应？版权所有人投诉存在合规问题时，应如何处理？

谁应履行合规义务？

如我们之前所解释的，许可证主要旨在保障用户有权出于任何目的运行程序，不受限制地复制、修改和共享。合规义务由许可证规定，许可证的起草者和许可人应考虑在其中设定尽可能少的合规义务，只要能保证下游用户的权利即可。Copyleft 是此系统的中心支柱：即要求基于上游免费代码的作品可获得“共享且以相同方式共享”。许可证辅助条款旨在防止削弱或违反“以相同方式共享”原则的行为。GNU 许可证创建了传统经济词汇中的“公共资源”概念；许可证施加的限制适用于公共资源的“治理”或“管理”规范，如捕鱼权限额或土地使用方面的地下水保护条例。各方对公共资源的义务取决于其在资源利用中所扮演的角色。我们可将各种角色及各方责任定义如下：

1. 代码使用者只有权利，没有义务。此类用户有权阅读、研究、理解、运行、修改、重用和转换所有形式的许可代码，只要许可证或任何限制例外条款具有相应规定。在权利行使方面，用户有权获取并利用作品的完整对应源码，此类源码包括最便于修改的程序源码、构建软件修改版本所需的辅助脚本、生成文件等，以及任何适用的安装信息。因为这些权利仅在用户知情的情况下才有效，所以用户必须获得相应通知。
2. 网络服务提供商 —— 当网络服务提供商在提供服务时使用的软件是专门用于通过网络提供服务的软件且为 GNU AGPL 软件时，此类提供商也对用户具有上述合规义务。

3. 许可作品分发者——无论是分发修改或未修改的作品版本、在设备中嵌入可执行的许可作品副本，还是仅出售或以其他方式传输数字副本，许可作品分发者至少应对（本人或中间方已向其分发副本的）用户履行义务。这些义务是否延伸至未直接收到作品的第三方取决于具体的许可证文本，以及分发者是直接分发源码还是提供要约表示愿意分发源码。此外，许可作品分发者有义务服务于上游用户、维护嵌入代码的合理法律通知，并恰当标明修订版本。
4. 为保护用户权利，服务提供商和分发者均应避免对下游用户施加任何额外限制。他们应避免使用“保护伞许可证”条款、最终用户许可协议（EULA）或分许可协议来制约下游用户的权利。按照 LGPL 条款，他们也不得对基于许可作品的软件适用此类条款：禁止替换大型非 LGPL 软件中的 LGPL 组件，或禁止反编译或反向工程来改进 LGPL 组件或修复 LGPL 组件中的缺陷。
5. 如果分发作品中包含专利内容，专利权人有义务不向其分发对象主张专利权。第三版 GNU 家族许可证要求修改和分发软件的专利权人不得向其版本分发对象主张任何权利，这不仅适用于其直接分发的作品，还包括后续任何包含专利权的版本和衍生作品。
6. 任何人如果基于约束性法律条件而不能按许可证履行义务，则不得成为前述的服务提供商或许可作品分发者。约束性法律条件指主动接受或迫于司法决定接受、限制义务履行方式的条件。如果遇到此类冲突义务情况，就意味着不能再自由地履行许可义务，因此不能再扮演服务提供商或分发者的角色。

如何履行合规义务

根据我们与构建 GPL 合规计划的商业伙伴的合作经验，以及作为 GPL 许可人的代理律师处理违规后果的经验，我们注意到企业所认为的合规与出现的问题、引发纠纷的原因和化解这些纠纷的方式等事实之间严重不符。我们经常发现公司准备斥巨资来规避不太可能出现的风险（在过去出现频率很低，且补救成本也不高），却疏于管理过去已引发诉讼和其他不良后果的风险。在此节中，我们会从广义上讲解如何使企业花最少的精力、以最低的成本履行合规义务，以防患于未然的精神讨论企业真正面临的合规风险。

根据我们的经验，曲解许可人的意图是导致实际合规风险与合规风险管理之间存在不符的原因所在。商业实体通常以为 Copyleft 项目社区将合规视作版权货币化方法，或意识形态上的努力来使私有软件按 Copyleft 条款再许可。假设许可人希望利用 Copyleft 违约谋取版税或使企业的私有产品以 Copyleft 协议分发代码，则企业需要管理此类风险：Copyleft 代码“片段”因“意外”或个别程序员疏于监督被复制到私有软件代码中。因此，在进行风险管理时，需要购买昂贵的“代码扫描”服务以检测此类意外的代码引入，而且应将精力集中在如何防止私有软件在编写过程中被自由软件“传染”。

但事实上，使用 Copyleft 的开发社区将违规视为改善合规性的契机。每次下游违规都代表用户权利的损失。项目开发者正如版权持有人一样，都需保障用户

的权利。他们的行为目的正是恢复用户的权利，保障项目贡献者编写软件的意图实现。项目开发者寻求合规时更多是因为软件交付方式受挫，而不是私有软件和自由软件的组合方式。尤其是：

- 未向用户提供购买产品中存在 Copyleft 软件及适用许可条款等必要信息；或
- 针对分发者已知正在使用且打算按照许可条款使用的 Copyleft 软件，用户无法通过可靠方式获取完整对应源码；或
- 用户按企业公布的地址联系企业请求其履行义务时，未得到任何回应。

在上述及类似场景中，项目目标在于确保严格履行对下游用户的义务，从而确保主动使用 Copyleft 软件所引发的合规义务得以履行。世界各地 Copyleft 社区曾提起的所有诉讼均可归因于此类违规，或是因为未实现合规，或者扫描程序或其他类似服务未能捕捉到此类违规行为。

将自由软件引入商业私有产品的做法时有发生。作为 Copyleft 开发社区的代理律师，我们确实隔段时间就会遇到此类问题，但并不是很频繁。据我们所知，社区方面未曾就任何此类问题提起过合规性诉讼。这些问题通常以友好合作的方式得到解决。

合规的关键在于管理

“软件管理”一词用于描述企业将软件存档、控制产品中引入何种软件、分发哪些软件、以及在买卖交易中适用何种许可证等流程。无论企业当前销售的是嵌入软件的实体产品还是软件产品和服务，完善的软件管理都是以最低成本履行合规义务的关键。

以假想的消费类电子设备制造商 Acme 公司为例。Acme 在某些设备中嵌入了 Android 系统，该操作系统包含 Copyleft 操作系统 (OS) 内核、Linux 操作系统、一些含 LGPL 库的定制应用程序和一些 GPL 实用工具。公司还在其他设备中嵌入自定义软件堆栈，此堆栈也包含 Copyleft 内核、经修改的 Linux 操作系统、适用 GPLv2 的 Busybox 和适用 GPLv3 的一套 GNU 实用工具。Acme 以数字化文件形式向转售其中某些设备的中间商提供固件升级服务。Acme 选择不随产品提供包含 GPL 源码的 CD 或 USB 记忆棒，而是在产品包装或文件中增加书面要约。因此，Acme 有义务向任何请求方提供产品中所有 Copyleft 程序的完整对应源码。

为履行该义务，Acme 需先准备一份“材料清单”，列出公司制造每个产品所用到的软件堆栈构件。“许可证分析”的目的是为了在着手开发前以及开发继续、修改材料清单的重复步骤中确认材料清单上的各个项目分别适用哪些许可证。出色的 FOSS 工具可用于许可证分析工作。产品文档所需的许可证文本在许可证分析环节中输出，并加入产品包装和在线文档。Acme 对其引入的任何 Copyleft 软件（无论是嵌入在硬件组件中，还是由第三方软件供应商提供）都会自动要求完整对应源码。通过以用户身份行使自己的权利，Acme 已准备好履行义务，并改善供应链中各实体的合规响应能力。

Acme 按 Copyleft 许可引入的所有源码（无论修改或未经修改）都在生成过程中进行传递。从初始测试版本到最终发布制造的所有阶段，Copyleft 二进制完整对应源码本身便是生成目标的一部分。发布包含此软件堆栈的产品时，针对所有 Copyleft 组件的完整对应源码会放在网站上供公众查看，以产品型号、修改代码或特定于设备和软件版本的任意散列码作为索引。此“堆栈码”也会印在产品文件中，并在产品外包装上以二维码形式显示。公司以数字化文件形式向中间商提供固件升级时，也会随附类似的堆栈码，以供其放在自己的网站上。

产品发布前的法律审查环节会确认 Acme 是否履行了其自制义务。任何 EULA（或适用于 Acme 产品中所有软件的保护伞许可证）和公司针对与产品相关的“软件即服务”部署制定的任何服务协议均需经过审核，以确定其是否明确合规产品软件堆栈中的任何 FOSS 许可证条款，进而确保未施加其他违反 Copyleft 的限制。

在软件管理过程中执行这些步骤后，Acme 坚信如果其产品的任何用户希望行使 GNU 家族许可证所赋予的权利，不用开箱便可请求并即时获得源码。Acme 也确信已以最小冲突和最低费用使每位购买其产品的用户都可获得相关的 Copyleft 许可证，而且公司自有产品的私有许可与产品中 Copyleft 许可不冲突，或不损害其有效性。

但这些步骤仍然无法保证 Acme 以后不会无意间将 Copyleft 代码引入公司的私有程序中，进而作为产品堆栈的一部分进行分发。但可以确信的是，Acme 已经避免了所有可能引起社区诉讼的违规情况。

有准备的合规原则

当然，真实世界不会像我们对 Acme 公司的简要描述这般这么简单。从汽车、家用电器再到联网或娱乐设备，每件现代产品都可能包含几十台电脑，而每台电脑又可能包含完整的嵌入式软件堆栈、混合 Copyleft FOSS、其他 FOSS 和私有程序。但以下几项基本原则在整个复杂领域中仍然适用：

1. 从下游用户行使许可证所授予的权利角度衡量您的合规情况。如果您在产品中引入了 Copyleft 软件，用户是否收到通知？用户是否可以轻松获取完整对应源码和适用的安装信息，而且最好是可自动获取？检查您交付内容的工具比仅可检查生成内容的工具更有价值。
2. 始终向软件和嵌入软件组件的供应商行使自己的权利，要求他们提供全部 Copyleft 作品的完整对应源码，最好通过自动化流程直接引入整个软件管理系统。在可能情况下，拒绝接收包含 Copyleft 软件的不合规组件：未按要求提供完整对应源码或者未在自动提供源码流程中在产品中随附“堆栈码”的组件。如果您的软件依赖于上游提供商，则不能仅因您所分发的软件由他人打包就忽视 GPL 合规要求。
3. 重点关注您已知正在使用的 Copyleft 软件。以往，因某些程序员不小心将一小段 Copyleft 代码引入私有软件产品带来的风险较为少见，且较容易化解。对相对较高风险的有效管理意味着您应确保可提供两年前发布产品中的 Coreboot Bootloader、Linux 内核、Busybox 或 GNU tar 的完整对应源码及生成文件等。

4. 不要盲目依赖代码扫描软件。用此类软件改进软件管理已为时过晚，而且无法捕捉到产品交付或售后提供源码问题。费用高昂的扫描软件只能处理不太重要的工作，根本无法处理更为重要的部分。性价比高的扫描软件倒是可用作自我管理和验证流程的补充，但不能作为主要的风险管理工具。

处理合规投诉

本指导的作者已经花了几乎三十年时间来维护 GPL，以个人名义或通过 SFLC 以集体名义参与了所有社区向美国法院起诉的 GNU Copyleft 维权案件。所有诉至法院的纠纷都在我们的帮助下达成和解，此类案例已有数十起。

在此背景下，我们也看到了互相误解所造成的后果。社区方面就违规行为提起投诉，以使违规方合规许可证要求。商业机构通常认为合规纠纷会导致索赔，或者此类纠纷是为了干扰其私有软件中的商业秘密部分，因此会以防御态度予以回应。社区方面已对 FOSS 领域的软件编程实践司空见惯，有时会误认为商业机构之所以不迅速提供其产品中主动引入的 Copyleft 程序的完整对应源码是在故意装糊涂。

根据我们的经验，如果能够在早期阶段以技巧性方式促进双方沟通，便可防止此类误解升级。以下指导以及一些过往案例可能会有所帮助：

1. 安排合适的人员回应来电。成功处理合规投诉的一个最重要原则就是保持沟通。将 FOSS 合规问题交给了解内部软件管理机制的特定个人处理可能是解决合规问题的最有效方法，此人可在合规问题出现时作为公关联系人与社区沟通。社区从未对配合早期沟通的一方提起过合规维权诉讼。

在一项涉及多名被告的合规维权案例中，一家大型跨国消费类电子设备制造商数月来一直未对履行源码义务的请求作出回应，仅因为公司法律总顾问在与本文作者之一（代表申诉的版权所有者的电话交流中个人口头保证会迅速合作，控诉方在提交诉讼前几个小时便将该制造商从被告名单中删除。

2. 假设投诉方已做好准备。传统上提出 Copyleft 违规投诉的组织（按历史顺序，依次是自由软件基金会 (FSF)、GPL-violations.org、软件自由法律中心 (SFLC) 和软件自由保护组织 (SFC)）都会在与明显违规的当事人联系之前，先对投诉情况进行全面调查并核实。投诉方也会准备好投诉所依据的证据事实。

在多年前的一起非蓄意在产品中加入自由软件的案件中，某全球化制造商在其旗舰私有软件产品中使用了一个完整的 Copyleft 库，用于编写该软件产品的基本特征。当我们代表版权持有人与该公司联系时，该公司负责 FOSS 事宜的法律顾问要么再三强调不可能发生此类事件，要么否认产品中包含 Copyleft 代码，但我们的软件工程师可以清楚地在该公司的产品中看到相关代码。我们不得不坚持三次让他们与自己的工程师重复核对，最终他们承认确实犯了这一错误。事实一旦查清，就能很快得到解决，后来此案没有进入诉讼程序，该公司也未蒙受经济损失。

3. 让软件工程师参与到流程中。根据我们的经验，大多数合规问题的解决过程中最耗时且最难的部分是核查相关方是否确实提供了完整对应源码。如果没有双方软件工程师的直接沟通，要解决分发的二进制代码是否与已提供的源码对应这样的技术问题很可能会变得非常复杂、昂贵，而且可能会造成局面紧张。法律顾问通常不愿意让自己客户的雇员直接回应潜在敌对方的调查，这也是人之常情。但想办法促使软件工程师就技术问题进行信息交流不仅可缩短解决问题的时间、显著降低问题的解决成本，而且能够防止因彼此误解导致事件升级。

4. 承诺合规。按照 GPLv3 家族许可证的规定，如果违规方迅速采取行动来纠正违规行为，则许可终止会自动撤销（反复侵权的情况除外）。按照 GPLv2 条款，如有违规，许可会自动终止。但是，如果违规方在接到通知后及时纠正错误，则多数 GPLv2 许可人（其程序广泛应用于商业）会立即恢复违规方的分发权。在经过细节确认并达成共识的前提下，承诺合规大部分时候都可促成双方和解，只有在极特殊情况下才会出现例外。

对于误将 Copyleft 代码引入私有程序作为软件产品分发的情况，过去的经验表明，立即主动承诺移除 Copyleft 代码，并向客户和其他可联系到的接收方再分发合规版本是避免诉讼、达成和解的关键措施。

5. 开展合规讨论以改善关系。开发社区编写软件的目的是为了服务于包括您在内的广大用户。在开发者看来，在产品中使用 Copyleft 社区软件意味着您成为社区成员中重要且有价值的一员。解决合规问题是加强您与其他用户之间关系的契机，因为您的工程师与项目成员之间的沟通会在此过程中加强，而这些项目成员输出的软件正有利于您的商业利益。

根据我们的经验，正是因为曾在合规设诉过程中与项目开发打过交道，各公司后来才能够经常与项目开发这达成互惠的咨询或雇佣关系。在某些案件中，纠纷解决需要双方合作来改变公司产品中使用项目代码的模式。更为普遍的是在此过程中建立的沟通渠道后来用于了其他更富有成效的目的。

6. 当一切尘埃落定时，主动提出承担费用。大多数对合规投诉进行调查并核实改进情况的工作是由非营利性组织代表社区完成的，没有任何赢利。社区行为准则是反对利用他人的错误寻求租金和盈利，这意味着社区程序员在准备提起合规投诉、解决纠纷的过程中投入的实质性工作需要用社区的微薄收入支付。⁹在合规事件成功解决后，如果违规方主动提出承担社区方面的成本，将有助于保证未来双方继续保持良好关系。如果社区要求违规方承担这些费用也是完全合理且可预见的，不过主动承担费用可以避免事件升级，这是我们在执业过程中多次发现的。

⁹SFLC 向制作并分发 FOSS 的非盈利方提供免费法律援助。我们的运作（包括处理合规事宜的工作）资金并非来自和解收益，而是来自认同服务价值的公司和个人的捐款。SFLC 偶尔会作为我们的项目代表参与和违规方的经济和解。此类和解收益在 SFLC 历史收益中的比例不到 2%，且未达到任一财年年度收入的 10%。我们没有在任何和解费用中扣除特定案件费用，更不用说全权代表客户的费用。

附件 1：源码提供要约

从实践指南到 GPL 合规指导¹⁰

许多分发者在分发二进制代码时只会附随源码提供要约，而不是直接提供完整的源码包。如果分发源码的成本是按件计费，这种做法可带来价值。例如，对于永久性存储器因过小而无法容纳源码的嵌入式产品，或者如果产品发货时没有随附 CD，而只有手册或其他印刷材料，这不失为明智之举。

但这么做实际上明显延长了您履行义务的持续时间。要约的有效期必须为完整的三年，自最后一次分发二进制代码（按照 GPLv2 的规定）或者二进制代码或备件（按照 GPLv3 的规定）计起。源码请求和提供系统的有效期必须比产品生命周期持续更久。

此外，如需合规 GPLv2 的条款，那么您无法通过网络来提供源码。GPLv2 规定只能通过物理介质提供源码。这通常意味着您必须在产品生命周期结束后的很多年继续制作最新版本的“源码 CD”。

除了利用物理介质外，提供网络链接以供下载源码的方式也可接受。此方法能让能够快速上网的人更快地获取源码，而且通常能够减少物理介质分发请求数。（GPLv3 §6(b) 允许仅通过公共网络分发源码，完全不需要物理介质分发。我们会在本节的最后详细讨论这一点。）

以下是一份符合 GPLv2 和 GPLv3 的合规要约建议模板，此要约可随附在印刷材料中随二进制代码分发：

The software included in this product contains copyrighted software that is licensed under the GPL. A copy of that license is included in this document on page X . You may obtain the complete Corresponding Source code from us for a period of three years after our last shipment of this product, which will be no earlier than 2011-08-01, by sending a money order or check for \$5 to:

GPL Compliance Division
Our Company
Any Town, US 99999

Please write “source for product Y” in the memo line of your payment.

You may also find a copy of the source at
<http://www.example.com/sources/Y/>.

This offer is valid to anyone in receipt of this information.

有一些关于此要约的重要细节：首先，分发者要求收取复印费。GPLv2 允许“收取费用，但不得超过通过物理媒介分配原始码所产生的费用”，此费用必须合理。如果复印费和 CD 邮费超过 10 美元，则应寻找较便宜的 CD 储存和邮寄

¹⁰http://softwarefreedom.org/resources/2014/SFLC-Guide_to_GPL_Compliance_2d_ed.html

方法，没有必要向社区收取过多费用。滥用此条款在源码分发上获取利益很可能产生维权行动。

第二，最后一行明确此要约对请求源码的所有人均有效。这是因为 v2 §3(b) 规定要约应“向任何第三方”提供对应源码的副本。GPLv3 也包含类似规定，指明此要约必须对“拥有目标代码的所有人”有效。v2 §3(c) 和 v3 §6(c) 中指出了这些要求，以便非盈利性分发者可将这些要约随附于分发包一起传递。因此，这些要约不仅对您的客户有效，而且适用于收到客户二进制代码分发包副本的所有人。许多分发者会忽视此规定，以为自己只需理会直接客户的请求。

选择随附源码提供要约而不是直接提供源码的做法对于能够处理合规履行流程的公司而言特别有用。为了避免非盈利性分发者承担履行合规请求的义务，GPLv2 §3(c) 和 GPLv3 §6(c) 允许其仅传递自己收到的要约。

请注意商业分发者无法利用选项 (c) 的例外情况；因此，尽管源码提供要约必须对收到此要约（按照 v2 的规定）或目标代码（按照 v3 的规定）的所有人有效，但不会免除任何商业再分发产品者的合规义务。许可条款适用于分发 GPL 软件的所有人，不管其是否是原始分发者。以供应商 V 为例，其使用 GPL 源码开发了一个适用于嵌入式设备的软件平台。制造商 M 与 V 签署合同，将此软件作为固件安装在自己的设备中。V 向 M 提供软件，并在其中随附合规源码提供要约。在此情况下，M 不能仅直接将 V 的源码提供要约交付给自己的客户。由于 M 也是以盈利为目的 GPL 软件，因此也须合规 GPL 并向自己的客户提供源码（或 M 自己的源码提供要约）。

这一情况说明对于客户可能会再分发的产品，源码提供要约的做法并不可取。如果产品中已随附源码，则您向自己客户的分发符合规定，而且您的客户再向其客户分发（未修改的）产品也是合规的，因为两者的产品中都已包含源码。而如果您选择随附源码提供要约，则您的分发符合规定，但您的客户再分发时若没有附随自己的源码提供要约，则不会“继承”您的合规性。

按照 GPLv3 的规定分发产品时适用的条款具有较大差异。按照 v3，您只需通过网络服务器提供源码即可，前提是公众可获得源码且源码自产品或相关备件最后分发日起三年有效。所以，仅通过网络分发便可履行义务。这简化了仅按 v3 分发产品时的“源码提供要约”流程，减轻了商业分发者的合规负担。但是，在改为仅以网络为基础的履行流程前，必须先确认您是否确实可依据 GPLv3 分发所有软件。某些程序确实适用“GPLv2 或任何后续版本”（英文通常简称为“GPLv2-or-later”）许可。此类许可让您可按照 GPLv3 再分发产品。但是，一些热门程序只适用 GPLv2 许可，无法进行“任何后续版本”许可（英文通常简称为“GPLv2-only”）。对于后者，您不能仅通过网络履行义务。

如果确定整个产品中的所有 GPL 作品都支持升级到 GPLv3（或者一开始时便已采用 GPLv3），则只需随附如下简单源码提供要约：

The software included in this product contains copyrighted software that is licensed under the GPLv3. A copy of that license is included in this document on page X . You may obtain the complete Corresponding Source code from us for a period of three years after our last shipment of this product and/or spare parts therefor, which will be no earlier than 2011-08-01, on our website at

<http://www.example.com/sources/productnum/>.

按照 GPLv2 和 GPLv3 规定，每次分发产品时，源码提供要约必须附随于许可证文本一起提供，电子版或打印版形式均可。

参考文献

1. Free Software Foundation, FSF FAQs about the GNU Licenses. <http://www.gnu.org/licenses/gpl-faq.html>, October 2014. (Accessed Oct 23, 2014).
2. Free Software Foundation, How to use GNU licenses for your own software. <http://www.gnu.org/licenses/gpl-howto.html>, April 2014. (Accessed Oct 23, 2014).
3. Free Software Foundation, Various Licenses and Comments about Them. <http://www.gnu.org/licenses/license-list.html>, October 2014. (Accessed Oct 23, 2014).
4. Bradley M. Kuhn, Aaron Williamson, Karen M. Sandler, A Practical Guide to GPL Compliance. <http://softwarefreedom.org/resources/2008/compliance-guide.html>, August 2008. (Accessed Oct 23, 2014).
5. Software Freedom Law Center, Managing Copyright Information Within a Free Software Project. <http://softwarefreedom.org/resources/2012/ManagingCopyrightInformation.html>, September 2012. (Accessed Oct 23, 2014).
6. A History of the GPLv3 Revision Process, Eben Moglen. https://www.softwarefreedom.org/resources/2013/A_History_of_the_GPLv3_Revision_Process.pdf, July 2013. (Accessed Oct 23, 2014).
7. Free Software Foundation, GPLv3 Discussion Draft 1. <http://gplv3.fsf.org/comments/gplv3-draft-1>, January 2006.(Accessed Oct 23, 2014).
8. Free Software Foundation, GPLv3 Discussion Draft 2. <http://gplv3.fsf.org/comments/gplv3-draft-2>, July 2006.(Accessed Oct 23, 2014).
9. Free Software Foundation, GPLv3 Discussion Draft 3. <http://gplv3.fsf.org/comments/gplv3-draft-3>, March 2007. (Accessed Oct 23, 2014).
10. Free Software Foundation, GPLv3 Discussion Draft 4. <http://gplv3.fsf.org/comments/gplv3-draft-4>, May 2007. (Accessed Oct 23, 2014).