翻译来自<https://www-archive.mozilla.org/mpl/relicensing-faq>

**1. MPL-2.0是如何起草的?**

MPL-2.0经过了21个月的公开起草过程，包括来自MPL用户、律师和开源社区团体（如Eree软件基金会和开源倡议组织）的广泛反馈。

**2. 修订过程是否公开记录?**

有关修订过程的历史文件可从本网站获得（https://www-archive.mozilla.org/mpl/relicensing-faq）。

**3.MPL-1.1的第6节说Netscape可以更新许可证，但Mozilla不是Netscape。Mozilla还能更新许可证吗?**

作为Mozilla基金会创建的一部分，Netscape赋予了Mozilla分发新许可证的能力，使Mozilla基金会成为Netscape的接班人。

**4. 我的项目使用MPL-1.1，为什么要升级到MPL-2.0?**

如果您的项目是在MPL-1.1下许可的。有几个重要的原因，为什么你应该把你的项目移到MPL-2.0:

MPL-2.0使遵从更简单，无论是对您还是对从您那里接收代码的人；

MPL-2.0为您和您的贡献者提供了与其他开源许可证更一致的专利保护，并允许您的整个社区在贡献者被起诉的情况下保护其贡献者；

与Apache和GPL的兼容性使您和更广泛的开放源码社区更容易重用和重新分发代码；

关于更详细的原因列表，请参见关于Q8问题。

**5. 我是代码的作者，我已经放置在MPL-1.1，我如何在MPL-2.0下重新许可我的项目?**

如果您是作者，要将许可证从MPL-1.1更改为MPL-2.0，请使用MPL-2.0附录A中的新标题替换旧的许可证头。

**6. 我正在分发由其他人根据MPL-1.1条款编写的代码，我可以在MPL-2.0而不是MPL-1.1的条款下分发这些代码吗?如果是这样，如何做?**

是的，你可以，因为MPL-1.1允许任何人在MPL-1.1的新版本下分发其授权的代码，为此, 请使用MPL-2.0附录A中的新标题替换旧的许可证头。

如果你收到只有MPL-1.1授权的代码,而不是MPL和GNU的多重授权（如Mozilla Tri-License），则还必须添加附录B到许可证头。

**7. 在MPL-1.1和MPL-2.0之间有什么没有改变?**

许可证中最重要的部分——文件级copyleft——在MPL-2.0和MPL-1.1中本质上是相同的。

**8. 在MPL-1.1和MPL-2.0之间有什么变化?**

主要的变化是简化。例如，不是精确地指定源代码必须可用的时间，而是当可执行文件可用时源代码必须可用；许可证标题变得更短，通知要求也简化了。总的来说，这个许可证相当短，而且应该更容易理解。

许可证中还添加了一些新特性。例如，许可证现在与Apache许可证兼容——任何遵守MPL条款的人也应该遵守Apache许可证的条款。类似地，默认情况下，许可证允许代码与GPL或LGPL许可的代码一起分发。此外，专利保护与其他许可证(如Apache)的使用更加一致，同时也允许社区的任何成员为被起诉侵权的贡献者辩护。

对于那些想要更详细地分析这些变化的人，还可以逐字逐句地列出所有变化的红线（https://www.mozilla.org/en-US/MPL/2.0/differences/）。但是，请注意，这些更改是广泛的，您可能会发现，仅仅阅读新许可本身就可以更清楚地理解其内容。

**9. 为什么新的许可证删除了政府实体语言?**

根据MPL许可的软件是根据48 C.F.R. 2.101定义的商业项目，除非它最初是根据美国政府的指示编写的，并供美国政府使用。因此，在不改变美国政府用户对MPL许可代码的权利和责任的情况下，从许可证中删除该语言以简化许可证。

**10. 为什么MPL-2.0不要求分发者在源代码中包含许可证的完整副本?**

在编写MPL-1.1时，源代码分发主要通过压缩文件和zip文件进行，相比之下，现代的软件分发通常是在网络上file-by-file的基础上进行的(例如http://hg.mozilla.org)，因此，MPL-2.0要求分发者告诉接收者如何获得许可(例如，通过在源文件的头文件中的链接到它)，而不是要求他们分发许可证本身。在大多数情况下，一起分发许可证和代码——正如MPL-1.1所要求的那样——仍然是满足这一要求的最好和最简单的方法。然而，考虑到现代源代码和可执行形式的分发多样性，给分发者更多的灵活性并让其选择有效通知机制是明智的。

**11. 为什么新的许可证在源代码的定义中删除了构建脚本和文档的引用?**

如果您选择在MPL下授权软件，那么在MPL下分发所有软件(包括接口文件和构建脚本)被认为是最佳实践。然而，由于这些条款只适用于分发特定形式或特定类型的软件，现在MPL应用于各种软件的不同方面的分发，我们删除这些引用，并将更广泛的“首选形式”的定义视为更适用的情况。

**12. 第3.2节现在要求我“通知”接收者如何获得源代码形式，能举例说明“通知”的意思吗?**

在历史上，通知接收者源代码的可用性是通过软件的“about”框完成的。但是，对于所分发的软件类型来说，其他通知用户的机制也是可以接受的。例如，当在网站上使用MPL许可的JavaScript时，收件人可以通过在网站的“法律”或“通知”部分提供源的链接得到通知。更普遍地说，通知应该放在一个人们可能会寻找关于他们所收到的软件的法律信息的合理的地方。

**13.与MPL-1.1不同的是，MPL-2.0包含了关于与GPL、LGPL许可的代码分发的明确规定，为什么?**

提供一种显式的机制，声明MPL可以和GPL、LGPL代码一起分发，有几个显著的好处:

它允许消除常见的双许可证和三许可证，从而减少许可证扩散，因为(出于兼容性和扩散目的)每个双许可证和三许可证都是一个单独的许可证。

除了与Apache兼容，它还声明了一系列向上兼容的自由软件许可证，覆盖了世界上大部分的自由和开源软件。

通过要求修改的初始分发在两个许可证下授权，而不仅仅是在GPL下授权，有助于保护原始许可方重新集成下游所做修改的能力。