

번역본(2015.04)

번역 : 한국저작권위원회

제공 : SFLC

소프트웨어 자유 법률 센터
일반 공중 라이선스 준수 가이드
개정 2판

에벤 모글렌(Eben Moglen)

미시 차우드하리(Mishi Choudhary)

2014년 10월 31일

내용

GNU GPL 소개, 사용 이유, 사용 방법.....	3
카피라이트(저작권) 및 카피레프트	5
카피레프트의 개념 및 라이선스 메커니즘	7
라이선스 조항 분석.....	14
GPL 버전 2	14
GPL 버전 3	27
GPL 특별 예외 라이선스.....	45
AGPL.....	46
LGPL	50
컴플라이언스 책임에 대한 이해.....	62
컴플라이언스 의무 주체는 누구인가?.....	62
컴플라이언스 의무 충족 방법.....	65
컴플라이언스의 핵심, 거버넌스.....	67
준비된 컴플라이언스의 원칙.....	70
컴플라이언스 관련 문의 처리.....	72
첨부 1 소스 코드 제공의 서면 제안.....	77
참고 문헌	83

본 문서는 소프트웨어 자유 법률 센터(Software Freedom Law Center, SFLC) 및 저자의 법적 분석을 제시하고 있다. 본 문서는 SFLC의 고객 또는 고객사의 견해, 의도, 정책, 또는 법적 분석을 기술하고 있지 않으며 실제 상황이나 단체에 관한 법률 조언 또는 의견을 담고 있지 않다. 특정 상황에 관한 법 적용에 대해 법률 상담이 필요한 경우, 전문 법률 기관의 자문을 구해야 한다.

©2014, 소프트웨어 자유 법률 센터. 크리에이티브 커먼즈-저작자표시-동일조건변경허락 4.0(CC-BY-SA 4.0) 국제라이선스.

소프트웨어 자유 법률 센터(Software Freedom Law Center, SFLC)는 2008년 처음으로 GNU GPL 및 관련 라이선스의 컴플라이언스(compliance) 가이드를 발간했다. SFLC는, GPL이 적용된 소프트웨어가 임베드된 디바이스를 사용하는 안드로이드 및 여타 시스템이 큰 성공을 거두는 등, 지난 6년간 GPL 소프트웨어의 채택이 증가했다고 여기고, 해당 주제에 대한 가이드를 대대적으로 개정해야 할 시점이라 판단하였다. IT 관련 분야에서 지난 10년간 카피레프트 소프트웨어를 사용할 때 나타나는 여러 가지 변화들로 인하여, 라이선스와 관련하여 다른 부분을 강조해야 할 필요가 발생했다. 본 가이드는 라이선스의 의무 사항에 대한 법적 가이드를 명확하게 제시하고자 각 GNU 및 GNU 계열 라이선스의 컴플라이언스에 대한 조항별 분석을 담고 있다.¹ SFLC는 분석을 제시하면서 본 주제가 생소한 독자를 위해 카피레프트의 의미를 설명하고자 한다. SFLC는 또한 소프트웨어 거버넌스를 강조하고자 하는데, 이는 제삼자 소프트웨어를 수용, 사용, 수정, 배포하는 일반 비즈니스 프로세스로, 대기업 및 중소기업에 더 밀접한 관련이 있다. 본 가이드 마지막 챕터에서 SFLC는 거버넌스 및 컴플라이언스의 관계에 대해 언급하고 저작권자의 문의 또는 컴플라이언스 소송에 대처할 직접적이고 현실적인 조언을 제공하고자 한다.

GNU GPL 소개, 사용 이유, 사용 방법

¹ 본 문서는 GNU 계열 라이선스, 즉, GNU 일반 공중 라이선스(GNU General Public License, GPL) 버전 2 및 버전 3, 약소 일반 공중 라이선스(Lesser General Public License, LGPL) 버전 2.1 및 버전 3, 아페로 일반 공중 라이선스(Affero General Public License, AGPL) 버전 1, 버전 2, 버전 3의 컴플라이언스에 대해 다루고 있다. SFLC는 해당 라이선스 전반에 걸쳐 언급하고 있으며, 이를 '라이선스' 또는 'GNU 라이선스'라 칭한다. SFLC가 특정 라이선스 또는 특정 버전을 지칭할 경우, 버전 번호와 함께 약어로 언급한다. SFLC는 'GPL 하에 라이선스 되어 있는(licensed under the GPL)'을 자유 소프트웨어 커뮤니티에서 오래 사용된 'GPL이 적용된(GPL'd)'으로 사용한다.

여러 형태의 자유 또는 오픈 소스 소프트웨어 라이선스에는 대표적인 라이선스가 몇 가지 존재한다. ‘퍼미시브(permissive)’ 또는 ‘약한 카피레프트’라고 이미 널리 알려진 라이선스들은 대부분 프로그래머의 자유를 지키는 것과 관련이 있다. 본 가이드의 주제인 자유 소프트웨어 재단(Free Software Foundation, FSF)의 ‘카피레프트’ 라이선스는 프로그래머의 자유를 지키는 것과도 관련이 있지만, 모든 사용자의 자유 보호에 더욱 중점을 두고 있다. ‘카피레프트’ 라이선스는 프로그램 사용자 또는 해당 프로그램 기반 저작물 사용자 모두에게 아래와 같은 네 가지 자유를 보장하고자 하고 있다.

0. 추가적인 허가가 없어도 프로그램을 자신이 원하는 목적에 맞게 자유롭게 실행할 자유
1. 프로그램의 소스 코드에 포함되거나 해당 소스 코드를 통해 알 수 있는 노하우(know-how) 또는 기술을 읽고, 연구하고, 이해하고, 사용할 자유
2. 프로그램 코드의 일부 또는 전체를 개작, 수정, 개선, 또는 재사용할 자유
3. 해당 프로그램의 원본 및 개작 버전을 다른 이들과 공유하거나 공유하지 않을 자유

라이선스가 “무엇인가”를 소개하는 것보다는, “왜 라이선스를 사용하는가”를 먼저 살펴보면, 라이선스를 이해하는 것이 더 쉬워질 수 있다. GNU 라이선스는 프로그램 개발자들 혹은 그들의 변호사들에 의해 만들어졌는데, 그들이 이와 같은 라이선스를 만들 때는, 변호사의 도움이나 지원을 전혀 받지 못하는 개발자들도 이 라이선스를 자유롭게 사용할 수 있도록 만들었다. 이들은 자신들의 프로그램을 사용하는 모든 사용자들, 자신들의 프로그램의 개정 버전을 사용하는 모든 사용자들, 그리고 자신들이

만든 프로그램을 새로운 프로그램의 일부로 포함시켜서 사용하는 모든 사용자들의 자유를 보장하는 것을 목표로 하였다. 이와 같은 자유의 본질은, ‘카피레프트 프로그램의 독점화 강화 방지’이다. 만약 GPL 하에 놓인 프로그램이 사유코드와 결합하여 더 발전될 수 있는 경우가 있다면, 그와 같은 결합이 서로 패키지화되든지 새롭게 쓰이든지 여부와 상관없이, 사용자들의 권리를 보호하고자 하는 원 개발자의 의도는 방해받을 수밖에 없다. 따라서 “왜 카피레프트를 사용하는가”에 대한 해답은, 소프트웨어와 관련된 ‘독점적 성격 강화 금지’를 통하여 전체 다운스트림 사용자(downstream users)의 자유를 보호하는 것에서 찾을 수 있다. GPL 버전 3 전문(前文)에는 다음과 같이 명시되어 있다.

사용자 권리 보호를 위해, 우리는 다른 이가 사용자의 권리를 부인하거나 포기하도록 요구하는 행위를 방지해야 한다. 이에, 사용자는 소프트웨어의 복제물을 배포하거나 이를 개작할 경우, 특정 의무를 져야 하는 데, 이는 다른 이의 자유를 존중할 의무이다.

본 문서에서 해당 의무가 무엇인지 설명하고 있으며, 중간 사용자, 개작자(modifiers), 패키저(packager), 재판매자의 희생을 최소화하면서 어떻게 사용자를 위해 해당 의무 위반으로부터 가장 효율적으로 면책될 수 있는지 설명하고 있다.

카피라이트(저작권) 및 카피레프트

소프트웨어에 기본적으로 적용되는 법 제도는 저작권법이다. 카피레프트는 독특한

결과(자유 공유의 법적 보호)를 이끌어내기 위해 저작권법의 일부를 기능적으로 사용하는데, 이는 해당 라이선스의 중요한 법적 원리를 구성하고 있다. 저작자와 출판사의 권리 강화를 위해 일반적으로 사용되는 저작권법을 사용자 권리를 강화하기 위해 ‘해킹(hack)’하고 수정하였다.

카피레프트 프로그램 기반 저작물은, 동일한 카피레프트 라이선스 하에서 허가를 받아야 한다. 이는 카피레프트의 ‘세습적 효과’ 또는 ‘균등한 공유’ 원칙으로 언급되곤 한다. 이러한 ‘호혜’ 및 ‘균등한 공유’ 규정은 ‘특허를 받은’ 자신의 프로젝트 경쟁자와의 마주하고 싶지 않은 개발자와 자신이 사용하는 프로그램의 현 버전뿐 아니라 향후의 개작 버전에서도 GPL 라이선스가 보장하는 네 가지 기본 자유를 누리고자 하는 사용자를 보호한다.

사용자의 연구, 개작, 공유의 자유를 이행하기 위해, 바이너리 또는 실행 복제물을 받는 모든 GPL 프로그램 사용자는 완전하고 부합하는 소스 코드(C&CS)를 받을 권리를 가진다. 만약 C&CS가 바이너리와 함께 제공되지 않는다면, 소프트웨어 배포 매체에 상관없이 바이너리 복제물과 함께 C&CS를 제공한다는 서면 제안을 제공해야 한다. C&CS 제공 또는 서면을 통한 제안 의무는 사용자의 권리를 보호하기 위한 가장 기본적인 의무이다.² 해당 의무 충족은 라이선스 컴플라이언스의 중요한 요소이다. C&CS 제공에 대한 라이선스 의무 사항은 매우 엄격하다. 바이너리의 공급자는 반드시 모든 소스 코드, 생성 파일(makefiles), 사용자가 받은 바이너리 실행물 생성에 필요한 기타

² 카피레프트 또는 GPL이 이러한 의무 사항을 ‘전염성이 강하게(viral)’ 하게 만든다고 비판하는 당사자는 프로그래머가 코드를 가지고 원하는 대로 무엇이든 할 수 있어야 한다고 믿고 있으며, 소스 코드 의무 사항이 없는 라이선스가 ‘더 많은 자유’가 있다 보고 있다. 하지만 라이선스의 목적이 사용자의 권리를 보호하는 것이라면, 완전한 소스 코드 제공 의무는 피할 수 없는 것이며 소스 코드 없이 사용자의 자유가 효과적으로 보호될 수 없다.

구축 자료를 제공하거나 자료 제공을 서면으로 제안해야 한다.

모든 개별 사용자의 버그 수정, 기능 향상, 코드 재사용이 자유롭지 않다면, 사용자는 개발자가 사용자가 보유하길 바라는 소프트웨어의 온전한 가치를 박탈당하게 된다. 만약 라이선스의 조건을 집행하지 않는다면, 궁극적으로 첨단 기술의 독점적 성격이 강화되고, 카피레프트 프로그램이 지속적인 독점적 성격 강화로 인해 사라지게 된다. 따라서 소스 코드 의무 사항은 사용자가 네트워크를 통해 디지털 전송 형태, 컴퓨터 판독 가능 매체에 고정된 비트, 또는 실질적인 디바이스에 내장된 소프트웨어 형태로 소프트웨어를 받는지에 따라 적용된다. 소프트웨어의 형태는 사용자 권리의 범위 또는 질(質)에 영향을 미치지 않는다.

저작권법은 원저작자에게 독점적 권리를 부여한다. 사용자 권리 보호를 위한 카피레프트의 '예방적' 사용과 관련하여, 저작권자로서 원저작자나 그 대리인이 사용자의 권리를 집행하거나 보호하는 의무를 진다. 저작권 위반에 대한 소송은 저작권 시행을 위한 궁극적인 법적 메커니즘이고, 저작권법은 저작권자 또는 그 대리인에게 위반에 대해 조치를 취할 수 있도록 하고 있다. 커뮤니티에서 저작권자의 권리 집행을 지원함으로써 컴플라이언스에 노력을 기울이고 있으나 오직 저작권자 또는 그 법정 대리인만이 실제로 법 집행 절차를 개시할 수 있다.

카피레프트의 개념 및 라이선스 메커니즘

SFLC는 커뮤니티 및 기업과 함께 일하면서 GNU 라이선스의 조건 및 핵심 개념에 대해 많은 오해가 있음을 확인했다. SFLC는 본 섹션에서 라이선스 운영 방식이 카피레프트 기본 개념을 어떻게 이행하고 있는지 제시하고 각 라이선스의 조항과 해당 조항이 어떻게 컴플라이언스 의무를 부여하는지 설명한다. 판단 기준을 라이선스의 의도 및 그 근거의 핵심 개념에 두게 되면, 처음에 임의적이거나 직관에 반대되는 것처럼 보이는 것도 더욱 쉽게 이해하게 된다.

‘저작물’ 및 카피레프트

저작권법의 단위는 ‘저작물’이므로, 라이선스에서 언급되는 ‘저작물’은 저작권이 있을 수 있거나 저작권법 조항의 적용 대상이기도 하다. GNU 라이선스는 GNU 라이선스와 관련된 모든 저작물에 적용 된다. ‘저작물’ 또는 GPL에 따라 허가받은 ‘저작물을 기반으로 한 저작물’은 특별한 조건이 없을 경우, 완전하고 부합하는 소스 코드 관련 의무가 포함된 GPL 의무의 대상이 된다. 이 원칙은 종종 변호사 사이에서 이론 또는 추론에 근거한 논쟁을 일으키는 데, 이는 ‘저작물’이 컴퓨터 프로그래밍의 단위가 아니기 때문이다. 소프트웨어의 ‘루틴(routine)’, ‘오브젝트(object)’, ‘기능(function)’, ‘라이브러리(library)’, 또는 기타 소프트웨어가 GPL 코드와 결합한 경우, 하나의 ‘저작물’의 일부분인지를 결정하기 위해, 우리는 저작권법 상의 전문 용어로 바로 답변할 수 없는 질문을 해봐야만 한다. 해당 주제에 대해 변호사가 내린 결론은 문학 또는 예술의 저작권 맥락에서 만들어진 규정을 컴퓨터 프로그래밍이라는 또 다른 맥락에 적용하는 기준이 된다. 여기서 요점은 GNU GPL 라이선스가 국가별 저작권법이 승인한 ‘저작물을 기반으로 한 저작물’의 각 버전을 다루는 ‘올타리 한쪽 끝에서 다른 쪽 끝까지 발을 경작’하는 작업이며, 그 이상은 아니라는 점이다.

GNU GPL 라이선스에서 언급된 라이선스 적용 범위는 라이선스 저작물과 'GPL 프로그램 기반'의 모든 저작물이다. 미국의 변호사들이 자국 내 독자를 위해 만든 라이선스인 GPL 버전 2에서 다음 명시하고 있다.

'GPL 프로그램 기반 저작물'은 GPL 프로그램 또는 '원본 또는 개작 그리고/또는 다른 언어로 번역된 GPL 프로그램 또는 그 일부를 포함한 저작물'인 2차적저작물을 의미한다.

안타깝지만 이런 설명 방식은 그리 도움이 되지 않는다. 이로 인해 (미 법원이 가이드 제시에 실패한) 소프트웨어 내 '2차적저작물(파생저작물)' 원칙(미국의 개념)의 역할에 대해 성과 없는 논의가 수년간 진행되었다. 그래서 GPL 버전 3에서 SFLC와 FSF의 고객은 미국의 '2차적저작물'에 대한 설명을 빼고 기본 개념으로 돌아가기로 했다. GPL은 저작물과 저작물을 기반으로 한 저작물 일체를 다룬다. 여기서 '저작물을 기반으로 한'이란 저작권 허가가 필요한 저작물의 개작 또는 결합이라 정의할 수 있다.

그러나 GNU GPL 라이선스는 사용자의 권리 보호가 중요하듯 GPL 라이선스가 다루지 않는 프로그래머의 권리 보호도 중요하다 인정하고 있다. 프로그래머의 저작물이 이와 결합할 수 있는 GPL 라이선스 프로그램 코드와 '분리되고 독립적'이라면, 카피레프트의 의무가 따로 배포된 저작물에 적용되지 않는다. GPL 버전 2 제2조는 저작권 범위까지 카피레프트를 확장하려고 하지 않음을 명백히 밝히고 있다.

만약 [다른 프로그램 코드]의 확인 가능한 부분이 GPL 프로그램에서 파생되지

않았다면, 합리적으로 고려했을 때, 그 자체로 독립적이고 분리된 저작물이라면, 해당 부분을 분리된 저작물로서 배포한 경우, 본 GPL 라이선스 및 그 조항은 해당 부분에 적용되지 않는다. 그러나 GPL 프로그램 기반 저작물의 일부로 동일 부분을 배포한 경우, 전체 배포에 대해 본 GPL 라이선스가 반드시 적용되어야 한다.

GPL 버전 3 제5조에서 카피레프트의 지나친 확장으로부터 프로그래머의 권리를 보호하고자 다음을 명시하고 있다.

저장 또는 배포 매체 내에, GPL 저작물의 확장 형태도 아니고 더 큰 프로그램을 구성하기 위해 GPL 저작물과 결합한 것이 아닌, 분리되고 독립적인 저작물과 GPL 저작물의 편집저작물(compilation)'을 '집합저작물(aggregate)'이라 하며, 이러한 경우 해당 편집저작물과 그로 인한 저작권이 개별 저작물이 허용하는 범위를 넘어 해당 편집저작물을 사용하는 사람의 접근권 및 법적 권한을 제약하지 말아야 한다. 집합저작물에 저작물이 포함되어 있다고 해서 본 GPL 라이선스가 집합저작물의 다른 부분에도 적용되는 것은 아니다.

GPL 라이선스는 카피레프트의 범위를 저작권 수준으로 제한하고자 하고 있다. 그러나 GPL 라이선스는 '초기 바인딩(early-binding)' 프로그램 언어 내에 프로그램 코드의 '정적(static)' 연결(linking)과 '동적(dynamic)' 연결을 구분할 경우, 라이선스에서 제시한 것처럼 제한을 두지 않는다. GPL 코드와 '동적으로(dynamically)' 연결된 서브루틴(subroutine)은 단독으로 연결되므로 태생적으로 주요 저작물에 대한 카피레프트에 포함되지 않는다고 보는데, 이는 잘못된 인식이다. 하나의 저작물을

만들기 위해 두 개의 소프트웨어 컴포넌트를 합칠 경우(주요 라이브러리 서브루틴과 일부 라이브러리 서브루틴, 각각의 방법(methods)이 있는 두 개의 오브젝트, 또는 프로그램과 ‘플러그인(plugin)’의 결합), 만약 해당 결합저작물이 컴포넌트 저작권자의 허가가 필요한 데, 그 허가를 받을 수 없거나 이제까지 준수하지 않았던 조건하에서만 허가가 가능한 경우, 그 결합저작물은 컴포넌트의 저작권을 위배한 것이 된다. 결합저작물이 GPL 컴포넌트 또는 AGPL 컴포넌트와 결합한 경우, 카피레프트만이 유일하게 사용 가능하다. GPL 컴포넌트를 사용해야만 한다면, 해당 조항은 반드시 그 결합저작물 전체에 적용되어야 한다. 이는 실행물 배포 전에 해당 결합저작물이 링커(linker)로 만들어졌는지, 런타임(runtime) 시 효율적인 실행을 위해 라이브러리를 공유하고자 OS 커널을 통해 만들어졌는지, 런타임 시 언어 내 여러 레퍼런스(reference)의 ‘후기 바인딩(late-binding)’의 결과로 만들어졌는지 여부와는 무관하다.³

다운스트림 수신인 자동 라이선스

GPL 적용 프로그램 재배포 시, 수신인은 관련 GPL 조항에 따라, 재배포하는 저작물에 존재하는 저작물의 라이선스를 모든 업스트림 라이선서(licensor)로부터 자동으로 양도받는다. 이를 라이선스 권리가 수직적 흐름을 생성한다기보다 삼 차원적 흐름을 생성한다고 생각하면 된다. GPL 저작물의 수신인 전체는 ‘당사자 관계(in privity)’에 있거나, 모든 라이선서로부터 직접 라이선스를 받고 있다.

³ LGPL의 경우, 저작물과 정적(statically) 또는 동적(dynamically)으로 연결된 모듈에 대한 의무 사항이 다를 수 있다는 점을 주지해야 한다. 이는 아래 37쪽에 자세히 설명되어 있다.

이러한 자동 다운스트림 라이선싱 메커니즘은 카피레프트 작용의 중심에 있다. 모든 라이선서(라이선스 보유자)는 독립적으로 라이선스를 부여할 수 있고 라이선스 위반 시 독립적으로 라이선스를 종료할 수 있다. GPL 버전 2의 경우, 위반 시 자동으로 라이선스가 종료되지만, GPL 버전 3에서는 라이선스 조항을 위반한 당사자는 라이선스 종료 전에 조치를 취할 수 있다. 라이선스를 위반한 당사자의 다운스트림에 있는 당사자는 자신이 직접 라이선스를 위반하지 않은 경우 해당 라이선스를 유지할 수 있다. 다운스트림 사용자의 라이선스는 업스트림 라이선서로부터 직접 부여되며, 다운스트림 사용자에게 프로그램을 배포한 위반자의 라이선스에 의해 좌우되지 않는다. 같은 이유로, 프로그램의 다른 복제본을 획득한 위반자는 이를 통해 새로운 라이선스 권리를 획득하지 못한다. 프로그램의 업스트림 라이선서의 라이선스가, 라이선스 조항 위반을 이유로 종료될 경우, 다른 복제본을 획득한다 해서 해당 수신인에게 새로운 라이선스가 자동으로 부여되지 않는다.

추가 제한 금지

만약 라이선스 당사자가 '제한적인 조항들'을 카피레프트에 추가할 수 있으면, 사용자의 자유는 보호될 수 없다. 그러므로 GPL 라이선스의 주(主)목적은 달성하기 위해선 '추가 제한 금지' 원칙은 포기할 수 없다. 따라서 GPL 버전 2의 저작물을 기준으로 한 저작물에는 GPL 버전 2의 조항만이 적용되어야 한다. GPL 버전 3 제7조에서 특정 상황에서 라이선스 변형을 매우 제한적으로 허용하기 위해 허용 가능한 추가 조항을 나열하고 있다. 그러나 이러한 예외 조항과 함께 '추가 제한 금지' 원칙은 엄격하게

적용된다. 이와 같은 이유로, ‘수락을 위한 클릭(click to accept)’ 설치 루틴을 포함한 수락 요청이나 조치는 GPL 조항 위반이다.

대립 조항 시행

‘추가 제한 금지’ 원칙은 공유 과정에서 당사자 어느 일방에 의한 의도적 수정으로부터 카피레프트를 지키는 역할을 한다. 하지만 어느 일방이 라이선스에 없는 제한하에 있다면 어떠한가? 만약 A가 C로부터 받은 프로그램의 권리를 B에게 모두 양도할 수 있는 법적 권리가 없다면, B는 GPL이 B에게 제공할 수 있는 권리보다 적은 권리를 가지게 되는데, 이는 A가 자신이 소유한 권리 그 이상을 양도하지 못하도록 법이 정하기 때문이다. GNU GPL 라이선스는 사용자 권리를 위해 이러한 문제를 해결한다. 법 조항에 따라(법원이 명령한 경우나 여타 법률 문서에 명시된 제한 조건을 수락한 경우) GPL에 포함된 모든 권리를 양도하지 못할 경우, 배포 자체가 허용되지 않는다. 카피레프트 라이선스는 외부 법 조항으로 인한 카피레프트 컴플라이언스 불이행을 허용하지 않고 있다.

특허 라이선싱과 관련해 ‘대립 조항 시행’을 금하는 것은 중요하다. 관련 GNU GPL 라이선스 보다 제한적인 조항을 포함하고 있는 특허 라이선스를 수락한 경우, 특허 라이선스의 대립 조항으로 인해 프로그램 배포가 전적으로 금지된다. 해당 조항은 특허권 그 자체나, 특허권의 행사, 또는 GPL과 호환하지 않는 특허 라이선스 제공으로 인해 발효하지 않으며, 해당 조항을 수락한 경우나 이를 부과하는 법원 명령이 내려졌을 경우에만 발효된다.

라이선스 조항 분석

이제 우리는 GNU 계열 라이선스에서 프로그램의 일부 사용자와 배포자에게 사용자 자유의 다운스트림을 보호하기 위한 조치를 취하도록 명시한 이유에 대한 논의로부터 구체적으로 어떠한 의무 사항이 존재하며 이러한 의무 사항을 어떻게 충족시킬지에 대한 논의로 넘어가기로 한다. 다음 섹션은 각 계열의 라이선스별 본문에 명시된 조항에 따른 컴플라이언스와 이를 이행할 가장 효과적인 방법을 구체적으로 기술하고 있다. 또한, 더 나은 소프트웨어 거버넌스를 통해 컴플라이언스를 달성하는 방법을 보다 광범위하게 논의하고 컴플라이언스에 대한 문의와 소송에 대응하는 방법을 알아보도록 할 것이다.⁴ 실무에서, 우리는 이러한 정보를 라이선스 조항 순으로 제공할 경우 변호사들에게 유용한 참고 자료가 된다는 것을 발견하였다.

GPL 버전 2

1991년 발표된 GPL 버전 2은 23년이 지난 지금도 여전히 사용되고 있으며, 자유 소프트웨어 라이선스 중⁵ 가장 널리 사용되고 있다. GPL 버전 2의 전문(前文)은 주로

⁴ 본 문서에서 제시한 자료 대부분은 FSF에서 제공하는 권위 있고 매우 유용한 'GNU 라이선스 관련 자주 묻는 질문' (<https://www.gnu.org/licenses/gpl-faq.html>)에서 확인할 수 있다.

⁵ <http://www.blackducksoftware.com/resources/data/top-20-open-source-licenses>.

라이선서의 사회적 의도를 다시 언급하고 있으며, 제1조와 함께 사용자가 자신의 권리에 대한 실제 공지를 보유할 수 있도록 각 저작물의 소스 또는 바이너리 버전 배포판에 라이선스 전문(全文)을 포함하도록 규정하고 있다. 해당 의무가 컴플라이언스 소송의 상당 부분을 차지하고 있는데, 이는 주로 GPL 프로그램의 존재와 사용자가 구매한 실질적 제품 내 해당 라이선스 조항에 대한 필수 정보를 사용자에게 제공하지 않았기 때문이다.

컴플라이언스 엔지니어링(compliance engineering)의 가장 효과적인 방법은 필수 라이선스 본문을 소프트웨어의 컴파일링, 패키징, 배포 시 ‘메이크 타겟(make target)’으로 취급하여, 제품 스택 내 소프트웨어에 대한 라이선스 본문과 기타 ‘부산물(collateral)’이 바이너리 자체가 생성, 테스트, 패키징되는 것과 동일한 단계에서 같은 방식으로 제작되고 검증되도록 하는 것이다. SFLC는 GPL 소프트웨어가 포함된 제품을 만들거나 배포하는 다양한 하청업체들을 거느리고 있는 글로벌한 여러 기업들과 함께 일하면서 이러한 해결 방안을 가지고 해당 단계와 이와 유사한 단계에서 완벽한 컴플라이언스를 달성할 수 있었다.

GPL 버전 2 전문(前文)에서 규정한 또 다른 컴플라이언스 의무는 라이선스 본문 자체에 대한 수정 금지 의무이다. 배포자가 GPL에서 단어를 삭제하거나 교정 또는 ‘보조’ 조항을 추가하기로 했다는 이유만으로 컴플라이언스 문제가 발생하기도 한다. 다른 GNU 계열 라이선스와 마찬가지로, GPL 버전 2는 FSF의 저작물이며, 오직 원본을 그대로 유지한 라이선스 복제만을 허용한다.

제1조 프로그램 원본 복제 및 배포

본 조항은 GPL 프로그램 수령 후 발생하는 해당 GPL 프로그램 원본의 복제 및 배포권에 대해 다루고 있으며, 다음을 의무화한다.

- 적절한 저작권 공지의 명확한 공표
- 보증서를 제공할 수 있으나 그렇지 않은 경우, 보증의 제한 유지 또는 추가
- GPL 버전 2와 관련된 모든 공지 유지
- 라이선스 사본 제공

1978년까지 저작권 보호를 위해 저작권 공지를 의무화했던 미국의 통념과는 반대로, 베른 협약(Berne Convention)에서는 저작권 보호를 위한 저작권 공지와 등록이 의무 사항은 아니었다. 미국과 전 세계 대부분 나라에서 개발자는 코드를 작성하는 순간 저작권을 보유하게 된다. 그러나 미국의 경우 저작권 공지 및 등록에 법적 효력이 일부 남아있다. 거의 모든 자유 소프트웨어에서 저작권 공지 추가 및 유지를 의무화하고 있다. 저작권 공지 유지에 대한 자세한 모범 사례는 SFLC의 '자유 소프트웨어 프로젝트 내 저작권 정보 관리'에서 찾아볼 수 있다.⁶

제2조 프로그램의 개작 및 개작 버전 배포

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

제2조는 GPL 버전 2 카피레프트의 핵심이다. 제2조는 개작 버전이 동일한 라이선스 하에 배포되어야 한다고 명시하고 있다. 개작의 권리는 네 가지 기본권 중 하나이므로, GPL 버전 2는 개작을 제한하기보다는 보호하고 있다. 하지만 제2조에서 언급하듯, GPL 프로그램의 개작 버전은 이와 같은 사실 때문에 ‘GPL 프로그램을 기반으로 한 저작물’이며, 오직 GPL 버전 2의 조건에 따라서만 다른 이에게 배포될 수 있다.

제2조 a항은 모든 개작 버전에 대해 수정 부분, 수정 날짜, 개작자의 신원 정보 일부 표시를 의무화하고 있다. 적절한 형태로 해당 정보를 담고 있는 소스 코드의 표시를 통해 컴플라이언스를 달성할 수 있다. 소스 코드 버전 컨트롤 시스템의 모든 가용 정보를 제공할 필요는 없으며, 프로젝트 레벨의 체인지로그 또는 유사 파일의 대체물을 제공할 필요도 없다. 특정 소스 파일을 사용하거나 검토하는 프로그래머는 적절한 컴플라이언스를 통해 소스 파일로부터 프로젝트나 프로그램 버전을 확인할 수 있고 최근에 있었던 실제 개작 이력을 추적할 수 있다.

또한, 다른 이의 법적 고지에 대한 추가 보호로써, 제2조 c항은, 만약 개작 전의 프로그램이 ‘일반적인 실행 형태가 대화형 구조로 명령어를 읽는 방식’이고 법적 정보를 표시하거나 출력하는 경우, 모든 저작권 고지, 보증의 제한, 개작 표시, 라이선스 본문 위치가 대화형 구조로 표시되고 출력되어야 한다고 명시하고 있다. GNU 디버거, gdb, 혹은 GNU zile을 예로 들 수 있다.

GPL 버전 2 제2조는 카피레프트의 의무 사항을 다룰 뿐만 아니라, 의무사항이 아닌 부분에 대해서도 다루고 있다. 제2조의 나머지 세 단락에서는 개념 및 운영과 관련해

카피레프트의 범위를 한정하고 있다. 제2조는, 비(非) GPL 라이선스 저작자에 의해 작성된 GPL 소프트웨어와 유용하게 결합할 수 있는 분리되고 독립된 저작물의 코드에 대해, GPL 버전 2가 카피레프트를 확장하지 않음을 분명히 하고 있다.

분리되고 독립된 저작물이 별도로 전송되거나, DVD나 USB 스틱과 같은 소프트웨어 배포 매체에 있는 GPL 프로그램과 함께 ‘집합저작물’ 형태로 전송되는 경우, GPL 버전 2는 해당 코드에 대한 카피레프트 의무 부여를 하지 않는다. 새로운 ‘프로그램을 기반으로 한 저작물’이 이전에 분리된 저작물과 GPL 코드의 결합을 통해 개발되었거나 해당 저작물이 GPL 코드와 결합하지 않으면 실행될 수 없는 독립적인 저작물이 아닌 경우, 카피레프트 의무가 새롭게 개발된 저작물에 부과될 수 있다.⁷ 집합저작물의 경우, GPL 버전 2는 각 프로그램의 개별 라이선스가 보장하는 권리를 사용자가 행사하지 못하도록 하는 포괄적 라이선스 하에 집합저작물이 출시될 수 없음을 분명히 하고 있다.

제3조 오브젝트 코드, 원본, 또는 개작 버전 프로그램 복제 및 배포

GPL 버전 2 제3조는 저작물의 실행 버전이나 바이너리 버전의 배포에 관한 조건을 명시하고 있다. 제2조가 카피레프트의 핵심이기는 하지만, 제3조는 가장 중요한 컴플라이언스 의무의 핵심을 구성하고 있다. 사용자는 구축 가능한 소스 코드 없이는

⁷ 이는 컴플라이언스 불이행에 대한 주요한 위반 책임 범위 전체를 다루고 있다. 이차적인 저작권 책임은 다른 방법으로 덧붙일 수 있다. 만약, 예를 들어, GPL 저작물을 사유화하고 사유화한 저작물의 가용성 및 해당 저작물과 원본 GPL 저작물의 결합저작물을 다운스트림 사용자가 개발하고 재배포할 수 있음을 광고하며 별도로 제공하는 경우, 이러한 일차적 위반 행위가 발생하면, GPL 라이선스 저작권자는 해당 저작권 침해를 유도하고 기여했다는 이유로 소송을 진행할 수 있고 해당 소송에서 승소할 수 있다. MGM 스튜디오(MGM Studios) 대 그록스터(Grokster), 545 U.S. 913 (2005) 참조.

프로그램을 연구, 이해, 개작, 공유할 수 없다. 이에, 제3조에서는 제공해야 할 소스 코드를 명시하고 있다. 제3조의 주요 세 가지 컴플라이언스는 다음과 같다.

1. GPL 실행 프로그램이 인터넷을 통해 복제해 사용 가능한 경우, 완전하고 부합하는 소스 코드는 ‘동일한’ 장소에서 복제할 수 있도록 한다.
2. GPL 실행 프로그램이 실제 매체를 통해 배포되었을 경우, 소스 코드는 동일한 매체나 DVD 또는 USB 메모리 스틱과 같은 매체 군의 다른 장치를 통해 복제본과 함께 제공한다.
3. 앞의 방식 또는 다른 방식으로 배포된 실행물에, 적어도 3년간 유효한, 완전하고 부합하는 소스 코드 제공에 대한 서면 제안이 포함되어야 한다. 소스 코드 대신에 서면 제의가 포함된 GPL 코드의 패키징을 단순히 재배포한 비영리 목적의 재배포자의 경우, 자신이 받은 서면 제의를 함께 전달할 수 있다. 소스 코드에 관한 규정을 준수한 작성 예는 부록 I에서 찾아볼 수 있다.

‘완전하고 부합하는 소스 코드’는 무엇으로 구성되는가? 우선, 소스 코드는 ‘저작물 개작을 위해 선호되는 형태’이다. 셸 스크립트, 펄, 자바스크립트, APL에 이르는 번역된 컴퓨터 언어에 대해, 저작물 개작을 위해 선호되는 형태는 아마도 프로그램 형태일 것이다(하지만 압축, 최소화, 모호화, ‘비(非)소스 형태’를 생성할 수 있는 번역된 코드의 수정과 관련해 주의를 기울여야 한다).

제3조에 명시된 것처럼, 완전하고 ‘부합하는’ 소스는 배포된 실행물과 정확하게 동일한 프로그램 버전을 구축하고, 소스 코드를 개작하고, 개작 버전을 구축하기 위해 필요한 소스 코드, 구축 스크립트(build script), 생성 파일(makefiles), 설정 파일(configuration

files)과 기타 자료를 의미한다. 배포된 프로그램과 동일한 버전을 구축하는데 필요한 정보가 부족하다는 점은 배포된 소스 코드가 불완전하며 ‘부합하는’ 코드가 아님을 의미한다. 개작 및 개작 버전 재구축에 적합하지 않은 ‘부합하는’ 소스 코드, 예를 들면, 실제로 개작하기에 너무 모호한 코드는 ‘완전한’ 코드로 간주되지 않는다.

완전성 의무 사항의 예외가, 시스템 라이브러리에 포함되어 있는데, 이는 제3조에서 다음과 같이 정의되어 있다.

해당 컴포넌트가 자체 실행물이 없는 경우, 실행물이 구동되는 운영 시스템의 주요 컴포넌트(컴파일러, 커널 등)와 함께 (소스 또는 바이너리 형태로) 일반적으로 배포되는 것

이 예외의 목적은, 다운스트림 사용자의 권리 행사에 필요치 않은 소스 코드 배포를 해야 할 수도 있는 배포자의 의무를 덜어 주려는 것인데, 그 이유는 해당 코드가 이미 사용자가 이용하고 있을 것이라고 추측되는 운영 환경의 일부로써 이미 제공되었을 수 있기 때문이다.

가장 심각한 의무 불이행 상태는 완전하고 부합하는 소스 코드의 제공 및 제공에 대한 서면 제안을 이행하지 못하는 것으로, 이는 컴플라이언스 분쟁으로 이어진다. 모든 상업적 배포자에 대한 커뮤니티 소송 사건들은, 유효한 소스 코드의 제공 요청에 대해 완전하고 부합하는 소스를 제공하지 못한 것과 관련이 있었다. 사실상 컴플라이언스 시행 시 가장 부담스럽고 어려운 부분이 컴플라이언스 요청에 따라 제공된 소스 코드가 사실상 완전하고 부합한다는 것을 확인하는 것이다. 이와 같은 문제들은 모두 부적절한

거버넌스가 원인이다. 당사자들은 자신의 제품에 탑재된 바이너리 소스 코드를 유지하지 않는다. 그리고 해당 소프트웨어 컴포넌트를 실제로 구축하여서 그와 같은 구축법을 알고 있는 직원들은 회사를 떠난다. 훗날, 개작하고 재배포한 저작물의 저작권자가 소스 코드를 찾게 된다면, 당사자인 회사는 이를 어떻게 제공해야 할지 모르거나 불완전하거나 불충분한 코드를 제공하게 될 수도 있으며, 법적 해명 시, 무력함과 무지함을 드러내게 된다.

이 중 어느 것도 일어나서는 안 된다. 만약 완전하고 부합하는 소스 코드의 타르볼(tarball)이 소프트웨어 개발 프로세스에서 ‘구축 타겟(build target)’이어서, 완전하고 부합하는 소스 코드를 만들지 않는 이상 바이너리 구축을 성공할 수 없다라고 한다면, 이런 경우 그 어떠한 소스 코드도 잃어버리지 않게 된다. 컴포넌트들을 리스트로 구축하여 남겨놓게 되면, 이를 구축한 엔지니어링 지식은 계속 회사에 남게 된다. 웹상에서 일반에 공개된 소스 코드의 프로비저닝은 제품 배송 또는 RTM(release to manufacturing)의 자동화된 부분일 수 있다. 전 세계 여러 기업과 함께 일하면서, 우리는 소프트웨어 개발 업무에 상대적으로 간단하면서도 큰 비용이 안 드는 변화를 줌으로써 훨씬 많은 비용이 드는 ‘컴플라이언스 툴(compliance tools)’의 정밀 검사로도 발견하지 못하는 컴플라이언스 불이행 사례를 완전히 근절할 수 있음을 반복적으로 확인했다.

소스 코드 제공을 서면으로 제안하는 것으로 소스 코드 제공 의무를 잠시 미룰 수 있지만, 이는 장기적으로 컴플라이언스 비용을 인상시키게 된다. 소스 코드 제공 서면 제안은 제품의 배포가 중단되었을지라도 최소 삼 년 동안 유지되어야 한다. 이를 통해 요청자가 바이너리를 받지 못했을지라도 ‘특정 당사자’가 완전하고 부합하는 소스 코드 사본을 요구할 수 있다.

다른 이가 제작한 소프트웨어를 단순히 재배포하는 경우에도, GPL 프로그램의 상업적 배포자는 GPL 조건의 제약을 받는다. 이렇게 제약을 받는 경우, 배포자는 업스트림 라이선서 또는 공급자로부터 받은 소스 제공 서면 제안을 자신의 사용자에게 전달하기만 하면 자신의 공급자가 해당 사용자에게 소스 코드를 전달할 거라 여겨선 안 된다. 이러한 경우, 사용자 권리를 행사하여 될 수 있으면 자동화 프로세스를 통해 공급자가 제공한 모든 GPL 프로그램에 대한 C&CS를 요청할 것을 권장한다. 해당 C&CS를 받은 후 제품과 함께 전달하면 해당 라이선스의 컴플라이언스가 확실하게 이뤄진 것이다.

GPL 버전 3 제6조가 이에 해당하며, 여기서 비(非)소스 형태의 소스 코드 전달(conveyance)을 다룬다.

제4조

제4조는 직접 ‘추가 제한 금지’ 개념을 설명한다. 또한, GPL 버전 2가 아닌 다른 라이선스로 2차 라이선싱 하는 것을 금한다. 이 금지 사항은 고지의 의무가 있다. 보통 GPL이 적용된 코드 배포 시에만 카피레프트 의무가 적용된다고 명시하고 있지만 정확히 그렇지 않다. 2차 라이선싱 또한 카피레프트 의무가 발생하는 별도의 행위이다.

금세기 들어 20년이 지날 때까지 2차 라이선싱을 통한 GPL 버전 2의 컴플라이언스 불이행은 보기 드문 이론상으로나 가능한 것이었다. 하지만 더 이상 그렇지 않다. 유틸리티 컴퓨팅 기반으로 판매되는 가상화 서버 스택 내 ‘서비스로서의

소프트웨어(software as a service)’의 ‘클라우드’ 제공을 통해 컴플라이언스 위반이 상당히 발생했다. 이러한 가상화 서버 인스턴스(instances)를 제공하는 기업은 사용자와 계약 체결 시 자사의 가상 서버 스택 내 소프트웨어 ‘라이선스’를 주장하는데, 여기에 항상 GPL 적용된 OS 컴포넌트가 포함되어 있어 GPL 라이선스가 GPL과 호환되지 않는 조항에 따라 2차 라이선싱 된다. 이러한 상황을 해결하기 위해 기술적 조치나 엔지니어링 조치가 필요한 것은 아니다.

GPL 버전 3은 모든 2차 라이선싱을 금한다 (20쪽 참조). GPL 버전 3 소프트웨어의 ‘클라우드’ 또는 ‘서비스로서의 소프트웨어’ 배포가 기술적으로 충분히 라이선스 위반이 아니지만, 잘못된 계약의 결과로 라이선스 위반이 될 수 있다.

GPL 버전 2 제4조는 라이선스 조항을 위반하면 자동으로 라이선스가 종료된다고 규정하고 있다. 해당 규정은 GPL 버전 3 제8조에 수정되어있다. 저작권 위반 시 자동으로 라이선스가 종료되기 때문에 배포 재개(GPL 버전 3하에서 자동으로 재개 가능)에 대해 저작권자로부터 배포권에 대한 확실한 재승인을 받아야 한다. 권리의 복구에 근거해 위반자가 배포 재개를 요청할 경우, 커뮤니티 저작권자가 이에 대해 일관되게 협조적이었을 지라도 저작권을 상품화하려는 영리 기구에 GPL 버전 2 저작권이 더욱 집중되는 결과는 피할 수 없으며, 이로 인해 앞으로 자동 종료로 인한 권리 재개 시 상당한 비용이 들 수 있다. 왜냐하면, 자동 종료에 대해 법원 명령이 즉각적으로 뒤따르기 때문에, GPL 버전 2 라이선스 저작권자를 모으는 비(非)커뮤니티는 자동 종료로부터 불공정한 영향력을 잠재적으로 확보하게 된다. 이는 이제 상업적인 재배포자가 자신의 업스트림 기술 제공자로부터 GPL 버전 2 라이선스보다 버전 3을 받는 것을 선호할 수밖에 없게 된 주된 이유 중 하나다.

제5조 라이선스의 수락 불필요

영국의 관습법은 전통적으로 지난 500여 년간 ‘라이선스’를 보유자가 미(未)보유자에게 부여하는 재산권 사용에 대한 일방적 허가로 정의하고 있다. 전통적으로 법률대학원의 재산권 수업 과정에서, 라이선스의 예로 저녁 초대를 든다. 만약 어떤 사람이 저녁 초대를 받았고 자신을 초대한 사람의 집 문지방을 넘었다는 이유로 그로부터 고소를 당했다면, 초대받은 사람의 방어 수단이 그곳에 있도록 한 일방적 허가인 ‘라이선스’가 된다. 전 세계적으로 법을 분석할 때, 라이선스를 계약으로 혼동하는 데 이는 고질적인 문제이다. 왜냐하면, 로마법에서 유래한 다양한 법체계에서, 라이선스를 계약이 아닌 일방적 의무 개념을 인식하지 못하고 있기 때문이다 (물론 누구도 자신이 초대한 사람을 고소하지는 못하지만.....). GNU 라이선스들은 관습법적 측면에서 라이선스에 해당한다. 일정 한도 내에서 일방적인 저작권을 허여한다. GPL 버전 2 제5조는 이와 같이 자주 논쟁의 소재가 되기는 하지만 분명한 법적 구분이 있는 이러한 개념들에 대해 설명하고 있다. 라이선스 수락은 의무가 아니며 이를 요청하지도 않는다. 클릭랩(clickwrap) 또는 다른 조치를 통한 수락 요청이 의무화돼 있지도 않을뿐더러 이는 해당 라이선스가 부여한 허여 조건에서 금하는 “추가 조건의 부여”가 된다. 따라서 제4조 및 제5조에서는, 계약 양식, 수락 양식, 감시를 통한 승인 등, 일방적인 허가를 쌍무 계약으로 변경하고자 하는 여러 가지 활동들을 금하고 있다.

제6조 자동 다운스트림 라이선싱

이는 GPL 버전 2 ‘자동 다운스트림 라이선싱’ 조항에 관한 것이다. GPL이 적용된 프로그램을 재배포할 때마다 수신인은 각각의 원(原) 라이선서로부터 라이선스 프로그램을 복제, 배포, 개작하기 위한 라이선스를 자동으로 받게 된다. 위에서 언급했듯이, 라이선스 조건과 관련하여, 다운스트림 수신인으로부터 수락을 받기 위해 어떠한 조치를 하도록 규정하고 있지는 않다. 이는 각 저작권자가 코드의 하향 사슬에 있는 다운스트림 재배포자와 법률적 당사자 관계나 직접적인 당사자 관계를 맺도록 하고 있는 것이다. 여기에 다음 두 가지 법적 효과가 뒤따르게 된다. 첫째, 제6조에서 밝히듯이 라이선스 규정을 준수한 수신인은, 자신의 업스트림 소프트웨어 공급자가 라이선스 위반 행위로 라이선스가 종료되었을지라도, 개작 및 재배포를 포함한 모든 행동에 대해 유효한 라이선스 허가를 유지한다. 이는 수신인의 라이선스 권리가 자신의 업스트림의 컴플라이언스에 의한 것이 아닌 저작권자로부터 직접 발급되는 것이기 때문이다. 둘째, 대체 공급자로부터 추가 복제본을 확보할지라도 자동 종료된 라이선스는 복구되지 않는다. 그 이유는 라이선스 허가는 원(原) 라이선서로부터 나오는 것이고, 만약 해당 라이선서가 허가를 자동으로 종료했다면, 중간 단계의 라이선스 보유자가 어떠한 조치를 할지라도 종료된 권리가 복구될 수는 없다.

제6조에서 명시했듯이, 라이선서는 제삼자 수신인 또는 배포자에 의한 컴플라이언스 집행에 대해 책임을 지지 않는다. 모든 라이선스 보유자는 자신의 행위를 기반으로 원(原) 라이선서로부터 허가권을 득(得)하거나 잃을 수 있다.

제7조

본 조항은, 간섭 의무(interfering obligations)로 인해 라이선스가 보장하는 자유의 전달을 금지당한 라이선서와 관련된 문제를 다룬다. 제7조를 준수한다는 것은, 라이선스의 의무 사항과 상충하는 조건이 부과되었을 때, 다른 이의 코드를 재배포하기 위해 라이선스를 사용할 수도 없고 사용해서도 안 됨을 의미한다. 저작권자로부터 추가적인 허가를 받지 않았거나 다른 법의 의무 사항을 준수해야 하는 경우, GPL 프로그램이나 해당 프로그램을 기반으로 한 저작물을 재배포할 수 없다.

이러한 관점에서 ‘부과된 조건’이란 용어는 GPL 버전 2 제7조의 해석에 있어 매우 중요하다. 만약 관련된 법원 혹은 합의를 통해 조건이 시행되면 조건이 ‘부과’된 것이다. 그래서, 예를 들어, 제품 한 개당 로열티나 다운스트림 재판매를 금하는 조항이 포함된 강제적인 라이선스나 법원 명령으로 인한 합의를 통해서 제7조 의무 사항의 효력이 발행한다. 특허 라이선스를 수락하거나 권리의 제약이 있는 여타 계약을 체결하는 경우도 동일한 효력을 가진다.

따라서 GPL 버전 2 제7조 컴플라이언스는 기술 또는 엔지니어링 소송 절차라기보다 법적 검토의 문제이다. 우리가 아는 한, 제7조 위반으로 인해 소송이나 적대적 소송 절차가 진행된 적은 없다. 하지만 실무적으로, SFLLC는 여러 기업과 함께 제7조의 문제를 해결하기 위하여 해당 기업이 체결한 특정 협정을 무효화해야 하는지 혹은 수정해야 하는지에 대해 종종 논의를 해왔다.

GPL 버전 2 제7조는 특허 소송에서 되도록이면 별도의 합의를 보는 행위를 하지 않도록 하고 있다. 합의 당사자만 특허로부터 안전한 GPL 소프트웨어를 배포하도록

허가하는 것은, 다운스트림 당사자에 의한 재배포를 금지하지 않으면 효과적인 제약이 될 수가 없다. 만약 합의 당사자에게만 배포권을 허가하는 경우, 이는 GPL에 의해 양도된 권리를 제한하는 것이므로, 제7조 위반이 된다.

GPL 버전 3

18개월의 공론 후 2007년 발표된 GPL 버전 3은 특정 상황에 따른 컴플라이언스 의무가 추가로 포함되어 있으나, 주요 컴플라이언스 의무의 이행을 더욱 용이하게 하고 의도치 않은 위반으로 인한 라이선스 자동 취소 위험을 상당히 제거했다.

GPL 버전 3에서 미국에 국한되거나 어느 한 국가의 법률 시스템에 국한된 용어 사용을 피하는 것을 원칙으로 삼았는데, 이는 어느 한 국가의 시스템에 국한한 법적 추정을 피하기 위함이다. 일례로, GPL 버전 2의 ‘배포’와 ‘2차적저작물’은 분석상 혼동을 일으켰다. 1991년 이후 전 세계에 GPL 라이선스 소프트웨어의 사용이 급증한 상황에서, FSF와 그 변호사들은 혼란에서 오는 위험을 개선하고자 어느 한 시스템에 국한된 어휘사용을 피하기로 결정했다.

GPL 버전 3을 살펴보면, 국제적으로 사용되는 저작권 표현과 비교했을 때 이질적인 용어가 두 가지 언급되어 있다. 라이선스 저작물을 ‘확산시킨다(propagate)’는 것은 어떠한 활동을 의미하는 데, 해당 활동을 하기 위해선 그 활동을 수행하는 국가의 법체계에 따라 저작권자의 허가가 필요하다. 개인 사용 또는 개인 사용을 위한 개작은

해당 국가의 저작권법과 상관없이 ‘확산(propagation)’ 활동에서 완전히 제외된 활동인데, 이는 해당 국가의 저작권법이 네 가지 자유 중 0번에서 2번까지 침해하는 것을 방지하기 위함이다. 더 나아가, ‘확산’ 활동에 특정 저작권 제도 하에 부여된 독점권 일체가 포함되므로, 독점권을 유효한 라이선스의 요건으로 인정하는 제도도 또한 자동적으로 준수하게 된다.

다른 당사자의 저작물 수령 또는 복제를 가능케 하는 확산 활동을 ‘전달(conveying)’이라 한다. 일반적으로 전달은 카피레프트 의무의 효력을 발생시키는 활동이다. GPL 버전 3과 AGPL 버전 3을 구분하는 ‘복제본의 양도(讓渡) 없이 컴퓨터 네트워크를 통해 사용자와 쌍방향 대화를 하는 것’은 ‘전달(conveying)’ 활동에서 명백히 제외 된다(33쪽 참조).

해당 용어의 사용을 통해 GPL 버전 3은 해당 국가의 법 지식 및 기술적 요소에 기반을 둔 권위 있는 라이선스로 해석될 수 있다. 국가별 저작권 조항에 따른 GPL 버전 3의 컴플라이언스 여부가 궁금하다면 두 가지를 질문하면 된다. 하나는 법률적 질문이고 다른 하나는 사실에 기반을 둔 것으로 아래와 같다.

1. 본 활동을 위해 해당 국가 저작권법에 따른 라이선스가 필요한가?
2. 자신의 활동을 통해 다른 이가 프로그램을 복제하거나 그 복제본을 받을 수 있는가?

해당 국가의 저작권 변호사가 첫 번째 질문에 대해 아니라고 답변을 할 경우, 관련 활동은 ‘확산’ 활동이 아니며 GPL 버전 3만으로 허가를 받는다. 두 질문 모두에 긍정적 답변을 받은 경우, 해당 활동은 ‘전달’ 활동이므로 아래 제4조부터 제6조에서 부과하는

카피레프트 의무가 적용된다.

GPL 버전 3은 저작물의 전달자(conveyor)가 지원 라이브러리, 컴파일레이션 스크립트(compilation scripts) 등을 포함한 프로그램 구축에 필요한 모든 소스를 제공하도록 의무화하고 있다. ‘부합하는 소스’는 GPL 버전 3 제1조에서 아래와 같이 정의하고 있다.

오브젝트 코드를 생성, 설치하고, (실행 저작물에 대해) 오브젝트 코드를 실행하고 저작물을 개작하는데 필요한 소스 코드 일체로, 여기에 해당 저작물을 제어하기 위한 스크립트가 포함된다. 그러나 저작물의 시스템 라이브러리나 범용 툴 또는 해당 활동 수행 시, 개작 없이 사용하나 저작물 일부가 아닌 일반적으로 사용 가능한 자유 프로그램은 여기에 포함되지 않는다. 예를 들어, ‘부합하는 소스’에는 저작물에 대한 소스 파일과 관련된 인터페이스 정의 파일, 공용 라이브러리 소스 코드, 밀접한 데이터 통신 또는 서브프로그램과 저작물의 다른 부분 간 제어 흐름과 같은 저작물에 필요한 동적 연결 서브프로그램의 소스 코드가 포함된다.

GPL 버전 모두 완전하고 부합하는 소스 코드 개발 의무 사항에서 ‘시스템 라이브러리’의 소스 코드를 제외시키는데, 이는 ‘시스템 라이브러리 예외’라 불린다. 위 인용문에서 볼 수 있듯이, ‘시스템 라이브러리 예외’는 사용자가 OS 복제본과 함께 받을 수 있는 소스 코드에서부터 사용 불가능한 OS 시스템 소스 코드나 프로그램 개발에 사용되는 일반적으로 구할 수 있는 프로그래밍 툴의 소스 코드까지 받을 수 거라 여기고 이러한 소스 코드를 요청함으로써 해당 소스 코드를 개발해야 하는 상황이 발생하는 것을

방지하기 위한 것이다.

제2조 기본적 허용들

본 라이선스는 개작하지 않은 프로그램의 실행을 제한 없이 허용함을 명백히 밝히고 있다. 개작만을 하거나 아니면 전달 활동을 하지 않는 저작물조차도 라이선스 조항을 준수하기만 하면 확산 활동을 할 수 있다. 사용자의 라이선스가 종료된 경우, 그 사용자는 새로운 개작 버전을 만들어서 해당 복제물을 계속해서 서비스 할 수 없다.

제2조는 2차 라이선싱을 금하고 있다. 앞서 언급했듯이, ‘서비스로서의 소프트웨어(Software as a service)’에 작성된 부주의한 계약 또는 모호한 맥락은 이 금지 조항에 저촉되거나 저촉 될 수 있다.

제4조 원본의 전달

제4조에 따라 누구나 프로그램 소스 코드 원본을 제한 없이 전달할 수 있다. 전달자 준수 의무는 저작권 고지 큐레이션(curation)과 소스 코드와 함께 라이선스 본문을 제공하는 것으로 제한된다. 저작권 고지가 누락된 경우 이는 반드시 추가되어야만 한다. 아래 제7조에 따라 귀속 및 추가 허가에 대한 고지는 반드시 보존해야 하며(신규 귀속 및 허가가 생기는 경우는 이를 추가해야 한다), 모든 업스트림 보증의 제한을 반드시 보존해야 한다. 자동화된 소프트웨어 거버넌스 프로세스는 제4조의 모든 의무 사항을

많은 비용을 들이지 않고 간단하게 충족시킬 수 있다.

제5조 개작한 소스 버전 전달

제5조에 따른 고지 의무는 제4조의 의무와 동일하며, 이에 더해 개작에 대한 고지, 날짜, 앞서 언급한 GPL 버전 2에 따른 개작자의 신원 일부 제공과 같은 추가 의무가 존재한다.

제5조는 다음과 같이 GPL 버전 2의 집합저작물 조항을 다시 언급하고 있다.

저장 또는 배포 매체 내에, GPL 저작물의 확장 형태도 아니고 더 큰 프로그램을 구성하기 위해 GPL 저작물과 결합한 것이 아닌, 분리되고 독립적인 저작물과 GPL 저작물의 편집저작물(compilation)'을 '집합저작물(aggregate)'이라 하며, 이러한 경우 해당 편집저작물과 그로 인한 저작권이 개별 저작물이 허용하는 범위를 넘어 해당 편집저작물을 사용하는 사람의 접근권 및 법적 권한을 제약하지 말아야 한다. 집합저작물에 저작물이 포함되어 있다고 해서 본 GPL 라이선스가 집합저작물의 다른 부분에도 적용되는 것은 아니다.

제6조 비(非)소스 형태 전달

제6조는 프로그램의 '비(非)소스 형태' 배포에 대한 컴플라이언스 의무를 기술하고 있다.

비(非)소스 형태란 개작 시 선호되는 형태를 제외한 모든 형태를 의미한다. 바이너리 또는 실행물뿐 아니라 애매하고, 최소화되고, 압축되거나, 수정 시 선호되지 않는 형태가 비(非)소스 형태에 포함된다. 웹상에서 자바스크립트의 배포 시 GPL 버전 3을 준수하지 않는 경우가 종종 발생하고 있다.

비록 아직은 이로 인한 논쟁이 발생하지는 않았지만, 해당 분야에 대해 더욱 세심한 컴플라이언스 활동이 없는 현 상황을 봤을 때, 앞으로 관련 논쟁이 있을 것으로 예측할 수 있다.

제6조에 따른 완전하고 부합하는 소스 코드 제공 의무는 앞서 언급한 GPL 버전 2 제3조와 같으나 현대 기술 상황에 맞춰 컴플라이언스 프로비저닝을 쉽게 하도록 고안된 변경 사항이 추가되었다. GPL 버전 2처럼 소스 코드가 온라인 배포로 제공되지만, GPL 버전 2 제3조의 ‘동일한 장소’ 요건이 약간 완화되었다.

오브젝트 코드 복제 장소가 네트워크 서버일 경우, 이를 찾을 수 있는 장소를 오브젝트 코드 옆에 명확하게 제시한다면, 부합하는 소스는 동등한 복제 기능을 지원하는 다른 서버(본인 또는 제삼자에 의해 운영)에 위치할 수도 있다. 부합하는 소스의 호스트 서버가 어떤 서버인지 상관없이, 배포자는 이러한 의무 사항이 충족될 때까지 해당 서버를 사용할 수 있도록 해야 한다.

해당 조항에서 처음으로 상업적 배포 시 완전하고 부합하는 소스 코드의 제삼자 제공을 고려하고 있다. 비(非)소스 배포자는 제삼자의 소스 코드 프로비저닝 서버 위치를 명확하게 제시하고(비(非)소스 다운로드 ‘옆’) 해당 제삼자 서버가 의무로 명시된 기간

동안 운영될 수 있도록 해야 한다.

제6조는 바이너리나 기타 비(非)소스가 비트토렌트(BitTorrent)와 같은 P2P 프로토콜로 배포될 때 그와 같은 서버를 통해 소스가 제공되는 것을 허용한다. 위와 마찬가지로, 유일한 의무 사항은 서버에 소스 코드 위치 정보를 각 피어(peer)에 효과적으로 전달하는 것이다.

제6조에서 소스 코드 제공에 대한 서면 제안 관련 규칙이 약간 수정되었다. GPL 버전 2 제3조 b항에 따르면 제공에 대한 서면 제안은 반드시 3년간 유효해야 한다. GPL 버전 3 제6조는 서면 제안의 유효성이 3년 이상이 되어야 하고, 해당 프로그램이 임베드된 실제 제품과 관련해 예비 부품 및/또는 고객 서비스가 해당 기간까지 유효해야 한다고 규정하고 있다.

제1조의 부합하는 소스에서 제외된 ‘시스템 라이브러리’ 정의는 이미 언급한 GPL 버전 2에서의 정의와 상당히 다르다. 새로운 정의는 아래와 같다.

실행 저작물의 ‘시스템 라이브러리’에 저작물 전체가 아닌, (a) 주요 컴포넌트(Major Component) 패키징의 형태로 포함되어 있으나 그 주요 컴포넌트의 일부는 아닌 것과 (b) 주요 컴포넌트와 저작물을 사용할 수 있게 돕거나, 해당 구현이 일반에게 소스 코드 형태로 공개된 표준 인터페이스를 구현하도록 돕는 것이 포함되어 있다. 이 맥락에서 ‘주요 컴포넌트’는 실행 저작물을 구동할 수 있는 특정 운영체계의 필수 컴포넌트(커널, 윈도우시스템 등)이거나 그 저작물을 개발하는데 사용된 컴파일러, 또는 저작물을 실행시키기

위해 사용되는 오브젝트 코드 인터프리터(object code interpreter)를 의미한다.

본 정의를 통해 프로그램의 타깃 OS 배포에 포함된 언어 프로세서(컴파일러와 인터프리터)가 부합하는 소스 코드 일부가 아님을 명확히 하고 있다. GPL 버전 2에서 OS의 주요 컴포넌트에 첨부된 라이브러리 또는 기타 권한 부여(enabling) 코드나 보조(ancillary) 코드 또한 제외되는데, 이는 일반에 공개된 표준 인터페이스를 구현하기 위한 소스 코드가 이미 공개되어 있기 때문이다.

GPL 버전 3에서 가장 광범위하게 논의되고 있는 신규 의무 사항을 제6조에서 확인할 수 있다. 해당 의무는 ‘사용자 제품’ 내 GPL 버전 3 코드의 ‘설치 정보’ 제공이다. 제6조의 ‘설치 정보’ 제공을 통해 사용자가 자신이 소유하고 개인적으로 사용하는 제품에 내장된 GPL 버전 3의 코드를 개작하고 개작 버전을 사용할 권리를 보호하고자 한다. 정보를 제공 하지 않고 법적 소유자가 소프트웨어의 개작을 막기 위해 장치에 ‘록다운(lockdown)’을 하게 되면 이는 GPL 라이선스 원저작자가 사용자에게 부여하고자 한 자유를 해칠 수 있다. 만약 프로그램이 임베드된 장치 모두에 록다운이 구현되어 있다면, GPL에 의해 보장된 자유가, 미 대법원이 판시했듯, “희망을 짓밟는 공약(空約, 헛된 약속)”이 될 수 있다. *에드워드(Edwards) 대 캘리포니아(California)*, 314 U.S. 160, 186 (1941) (Jackson, J. 동의)

‘사용자 제품’은 다음과 같다.

- (1) 개인, 가족, 가계의 목적을 위해 일반적으로 사용되는 유형(有形)의 개인 소유물인 ‘소비재’ 또는
- (2) 주택에 불박이 형태로 있도록 설계되거나 팔린

일체의 것

GPL이 적용된 프로그램의 비(非)소스 형태가 소유권의 영구 양도나 장치의 제어 또는 일정 기간 대여를 통해 ‘사용자 제품’에 내장되어 사용자에게 전달된 경우, 부합하는 소스에 충분한 기술 정보, 방법, 절차, 권한 키, 또는 부합하는 소스의 개작 버전으로부터 해당 사용자 제품 내의 저작물 개작 버전의 설치, 실행 시 필요한 기타 정보를 반드시 추가해야 한다. 개작이 이뤄졌으므로 개작된 오브젝트 코드의 기능이 방해받지 않고 지속해서 이뤄질 수 있도록 해당 정보가 충분히 제공되어야만 한다.

만약 개작 버전의 설치 및 실행을 막기 위해 기기에 기술적 조치를 취하지 않았다면, 해당 요건은 크게 문제되지 않는다. 2007년 이후 자사의 제품을 잠그려는 여러 제조업체는 GPL 버전 3 소프트웨어를 피하는 경향을 보였다. SFLC가 알고 있는 바로는, GPL 버전 3의 공표 후 설치 정보 의무 컴플라이언스 불이행에 대한 적대적인 조치는 발생하지 않았다.

제7조 추가 조건

GPL 버전 2에서 추가 제약 조건을 허용하지 않지만, 버전 3은 일부 제한적 변형을 허용함으로써 다른 라이선스와 광범위하게 호환할 수 있도록 버전 2의 엄격한 카피레프트에 변화를 주었다. 제7조는 본 라이선스의 조건을 보완하기 위해 사용된 추가적인 허가(permissive) 또는 불허(不許) 조건으로 구성되어 있다. 저작물에 기여한 저작권자는 제7조에 명시된 조건 그 이상을 추가할 수 있다. GPL 버전 2와 달리 GPL

버전 3은 추가된 허가 전체를 서면상에 명시해야 한다고 규정하고 있다. GPL 기안자들은 GPL 버전 2하의 비공식적이거나 성문화되지 않은 추가적 허가의 제공이 불필요한 의혹을 발생시켰다고 보고 있다. 업스트림 저작자가 기존에 부여한 서면으로 작성된 추가 허가는 전달하는 측이 원할 경우 라이선스 프로그램 복제본의 전달 전에 삭제될 수 있다.

제7조 a항

보증의 제한: 저작권자는 제15조 및 제16조와 다른 방식으로 보증 책임을 부인하거나 법적 책임을 제한할 수 있다. GPL의 국제화에 따라, GPL 기안자들은 법률의 국가별 차이에 대처하기 위해 보증의 제한 및 책임 한계에 대한 조건을 변경하도록 허용했다.

제7조 b항

적절한 법적 고지 유지. 제7조는 법적 고지 유지에 대한 추가 조건을 허용하고 있으며 여기에 저작권의 실행 결과물도 포함이 된다.

제7조 c항

본 조항은 원본 자료에 대한 잘못된 설명을 금하는 것을 고려하고 있으며, 개작 버전을 GPL 자체의 정확한 표시 요건과 달리 ‘적절한’ 방법으로 표시하도록 의무화하고 있는 퍼미시브 라이선스와 GPL 버전 3을 호환하도록 하고 있다.

제7조 d항

홍보 목적의 라이선서 이름 사용 제한. 본 조항은 개작 버전의 라이선서 이름의 사용 및 광고 권리를 금하는 라이선스와의 호환성을 위해 추가되었다. 오랜 기간 때때로 문제를

일으켰던 ‘BSD 광고 조항’과의 비호환성이 본 조항을 통해 사라진 예라 할 수 있다.

제7조 e항

상표권법에 따른 권리 양도 금지. 본 조항은 퍼미시브 라이선스의 ‘상표권 양도 금지’와 관련하여 비슷한 기능을 갖는다.

제7조 f항

본 조항은 저작권자와 라이선서의 배상 요청을 허가함으로써 아파치 소프트웨어 라이선스 버전 2.0과의 비호환성 문제의 주요 원인 두 가지 중 한 가지를 제거했다.

제10조는 앞서 언급한 조항 이외의 허가를 제한하는 추가 조항을 ‘더 추가된’ 제한으로 보고 이를 금한다. 만약 제7조에 따라 추가 조항을 포함시켰다면 관련 소스 파일 내 해당 조항의 위치를 확실히 하거나 추가 조항을 발견할 수 있는 장소에 대해 명확한 고지를 해야 한다. 제7조에 명시된 컴플라이언스 의무는 기술 또는 엔지니어링의 비중이 없는 법적 검토이다.

(예) GNU 컴파일러 컬렉션(GCC) 런타임 라이브러리 예외

GCC 런타임 라이브러리 예외(이하 “예외”)는 GPL 버전 3 제7조에 따라 제공된 추가적인 허가이다. 본 예외의 목적은 본 예외가 적용되는 헤더 파일과 런타임 라이브러리를 사용하고 편집 프로세스의 일환으로 프로그램의 오브젝트 코드 내에 컴파일러를 통해 임베드된 카피레프트 툴체인(toolchain)에서 나온 코드를 포함하는

비(非)GPL(특허 포함) 프로그램의 편집저작물을 허용하기 위해서이다. GCC 런타임 라이브러리 예외는 예외 적용을 언급한 라이선스 헤더 내 공지가 있는 파일에 적용된다. 여기에 libgcc, libstdc++, libfortran, libgomp, libdecnumber, libgcov, 및 GCC와 같이 배포된 기타 라이브러리가 포함된다.

제8조 종료

위반 시 자동 종료에서 자체 시정 기간 및 첫 위반자의 자동 권리 복구로 해당 조항이 변경되었는데, 이는 GPL 버전 2에서 GPL 버전 3으로의 개정 작업 중 컴플라이언스 기본 규칙에 나타난 가장 중요한 수정 사항 중 하나라 할 수 있다.

그러나 만약 [자신의 라이선스 조항 위반 발견 후] GPL 라이선스에 대한 일체의 위반 행위를 멈췄다면, 그러면 특정 저작권자로부터 받은 라이선스가 (a) 저작권자가 분명하게 최종적으로 라이선스를 취소하지 않는 한, 그리고 취소할 때까지 조건부로 회복되며, (b) 만약 저작권자가 위반 행위 중단 후 60일 이내에 합당한 수단을 통해 당신이 위반하였다는 사실을 당신에게 공지하지 못하는 경우 영구적으로 회복된다.

더 나아가, 만약 저작권자가 합당한 수단을 통해 위반 사실을 통보하고, (어떤 저작물에 대한) 해당 라이선스의 위반 사실을 해당 저작권자로부터 통보 받은 일이 처음 있는 일이며, 해당 공지를 받은 후 30일 내에 위반 행위를 시정한 경우, 특정 저작권자로부터 받은 라이선스를 영구히 회복할 수 있다.

GPL 버전 2과 비교하여 봤을 때, 문제를 발견하고 시정한 것에 대한 인센티브를 재조정할 부분이 GPL 버전 3의 큰 장점이라 하겠다. 배포권 상실을 피하기 위해 저작권자에 의한 확실한 권리 복구가 필요한 자동 종료의 경우, 문제를 바로잡기보다 이를 숨기는 것을 장려한 셈이 되었다. ‘전체 배포 위반’은 디바이스에 임베드된 OS 배포판의 모든 카피레프트 프로그램에 대해 부합하는 소스 코드를 제공하지 못하는 경우를 일컫는다. 이 경우, 수십만 개의 저작권을 동시에 위반하게 된다. 권리 복구를 하지 않고는 자신의 제품을 배포할 수 없는 위반자에게 있어 이는 매우 큰 걸림돌로 작용 한다. 최근에 SFCL는 컴플라이언스 엔지니어링(compliance engineering)에 상대적으로 경험이 적은 중소기업 중 하나를 감사했다. 해당 기업의 과거 제품이 ‘전체 배포 위반’을 하게 되었고, 이로 인해 이 기업은 규정을 준수해 구축한 새 모델의 배포권을 유지하기 위해 GPL 버전 2 하에서 라이선스를 받은 100여 개 패키지의 저작권자 모두에게 연락을 취해야만 했다. 왜냐하면 초기 제품에서 새 모델의 프로그램과 동일한 프로그램이 규정을 위반하여 배포되었으므로 기업이 인지하지 못한 사이 프로그램의 라이선스가 자동으로 종료 되었기 때문이다.

제 10조 다운스트림 수신인 자동 라이선싱

이 조항은 본 문서의 다른 부분에서 이미 설명한 추가 제한 금지 및 자동 다운스트림 라이선싱 원칙을 기술하고 있다. 자신이 배포한 프로그램의 수신인에 대해 특허 청구권을 집행하는 것은 추가 제한에 해당하고, 제10조에서 이를 금하고 있다.

인수 합병 제10조에 따르면, 자산 매각 또는 경영권 양도를 통한 기업 인수 시, 인수 당사자는 합병 당사자의 다운스트림이 된다. 이를 통해 합병 기업의 모든 개작물, 현재 다운스트림 인수자를 위한 소스 코드 포로비저닝 권한이 업스트림 저작권자로부터 인수자에게 새롭게 자동으로 다운스트림 되게 된다.

SFLC가 경험한 바에 따르면, 대부분의 인수 합병 시 자동 라이선싱 관련 사안들을 조정하는 프로세스는 비용이 많이 들고 비효율적이다. 인수를 종료하기 전, 인수 당사자가 합병한 법인에 대한 GPL 컴플라이언스 관련 소송 일체를 포기하고 공개를 하는 경우 해당 문제가 해결된다. 인수 법인이 적절한 소프트웨어 거버넌스 시스템을 보유하고 있는 경우, 법인 인수 중 취득한 모든 소프트웨어를 이를 위한 표준 거버넌스 프로세스에 입력하여 새롭게 합병한 법인으로부터 다운스트림 된 컴플라이언스를 자동으로 관리한다.

제 11조 특허권

GPL 버전 3은 두 종류의 특허권 의무를 부여한다.

1. **수신인에 특허 소송 진행 금지:** GPL 버전 3 10조는 특허 라이선스 수락을 요구하거나 라이선스 수신인에게 로열티 지불을 요청하는 것을 금지하는 추가 의무를 명시하고 있다. 본 조항은 특정 시스템이나 지역의 특허와 관계 없는 GPL 프로그램 관련 특허 소진에 대한 확실적인 규칙을 수립하고 있다.

2. 기여자 버전에 대한 라이선스 권한 부여: 제11조는 GPL 프로그램에 코드를 제공한 기여자의 확실한 특허 청구권 부여를 소개하고 있다. 해당 조항은 자신이 개작한 코드를 사용하는 사용자에게 커먼즈 내의 당사자들이 공격적으로 특허를 주장하지 못하도록 하기 위해 만들어졌으며, 이렇게 함으로써 커뮤니티 ‘내부자’에 의한 커먼즈 배신 행위를 방지한다. 개작물 결합을 통해 개발된 프로그램 버전에 대해 필요에 의해 침해된 기여자의 특허 청구권은 모든 후속 사용자, 해당 프로그램 개작자, 또는 해당 프로그램을 기반으로 한 프로그램에 부여된다. 다른 당사자에 의한 후속 개작물의 특허 침해에 관해서 청구권이 부여되지 않는다. 해당 버전을 통해 어쩔 수 없이 특허를 위반한 ‘기여자 버전’의 개발 후 획득한 특허 청구권도 청구권 획득 또는 완성 시점에 본 조항을 통해 부여된다. 그와 같은 청구권을 다수 보유한 회사가 프로그램 개작자를 인수하면, 모든 보유 청구권 또는 인수자가 그 이후 취득한 청구권 전체는 자동으로 본 조항에 따라 라이선스 된다. 마이크로소프트의 노키아 인수를 예로 들어보면, 해당 인수로 인해 노키아가 개작한 GPL 버전 3 프로그램의 기여자 버전에 대해 현재 보유하고 있거나 인수 이후 취득한 모든 마이크로소프트 청구권은 자동 라이선스 되었다. 노키아 인수로 발생한 GPL 버전 3 프로그램에 대한 마이크로소프트사의 특허 청구권 일체 상실은 지금까지 업계에서 잘 알려지지 않았다.

제11조는 또한 차별적 특허 라이선스 집행을 도모한 당사자에 의해 만들어진 GPL이 적용된 일반적 권리를 약화시키는 잠재적 활동을 다루고 있다. 만약 A가 A의 고객 또는 수신인보다 A에게만 혜택을 주는 특허를 B로부터 받았다면, A는 B의 특허로부터 오는 위험을 자신이 감수하는 것이 아니라 다른 이에게 부과하게 되는 것이다. 여러 정황을

고려해 봤을 때, 이는 수용할 수 있는 결과이다. 그러나 만약 A가 해당 프로그램의 유일한 가용 소스라면, 그와 같은 라이선스를 받고 해당 라이선스에 의지해 배포를 함으로써, A는 B의 ‘해당 프로그램 사유화(私有化)’를 사실상 돕는 것이다. 제11조는 라이선스에 대한 권리를 포기하거나 당사자에게 해당 범위를 보장하는 것과 같은 A가 할 수 있는 다양한 해결 방안을 제시한다. 그러나 가장 가능성 있고 비용이 가장 적게 드는 컴플라이언스 행위는 제삼자를 통해 해당 프로그램 소스 코드를 영구적으로 사용 가능하도록 하는 것으로, 위험을 감수하더라도 해당 코드를 개발하거나 이를 상업적으로 배포(deploy) 또는 재배포 하길 원하는 당사자는 그렇게 할 수 있도록 하는 것이다.

‘다운스트림 보호’ 조항에 더해, 제11조는 차별적 특허 라이선싱 전략(tactics)의 과도한 조작을 막기 위해 두 가지 조치를 취한다.

만약 단일 거래 또는 협의에 따라, 혹은 이와 관련하여, 저작물의 전달 또는 이를 알선하여 확산하고, 저작물을 받은 당사자에 저작물의 특정 복제물을 사용, 확산, 개작, 또는 전달할 수 있는 권한을 부과하는 특허 라이선스를 부여한다면, 부여한 특허 라이선스는 모든 저작물과 저작물을 기반으로 한 저작물의 수신인에게 자동적으로 확대 적용된다.

만약, 예를 들어, B사가 A사로부터 A사에 의해 배포된 GPL이 적용된 프로그램의 권한을 제삼자에게 주는 쿠폰을 매입하고 자사로부터 구매한 쿠폰이나 선물 받은 쿠폰을 통해 A사 제품을 취득한 이들에게 특허프로그램 권한을 제공하면, 본 조항을 통해 다른 모든 사용자에게 특허 라이선스가 자동 확대 적용되어 B사는 해당 거래를 통한 금전적 가치를 박탈당하게 된다. 경험상 유추해 봤을 때, 이런 경우 B사는 발급된 쿠폰은 해당

쿠폰이 적용되는 소프트웨어 집합물(collection) 내의 그 어떤 GPL 버전 3 프로그램에도 적용되지 않는다고 소급해서 주장 하게 된다. 하지만 이러한 주장은 효과가 없다.

어떤 특허 라이선스가 본 GPL 라이선스가 부여하는 한 개 이상의 특정 권리를 그 적용범위 내에 포함하지 않거나 그 권리 실행을 금하는 경우, 혹은 그 권리의 불이행을 조건으로 내세우는 경우, 해당 특허 라이선스는 ‘차별적’인 것이다. 만약 어떤 이가 소프트웨어 배포 사업을 하는 제삼자와 저작물을 전달하는 활동 범위에 따라 그 제삼자에게 돈을 지불하고, 그 제삼자가 저작물의 수신인에게 (a) 자신이 전달받은 저작물의 복제본(또는 해당 복제본의 복제본)과 관련된, 혹은 (b) 저작물을 포함하는 특정 제품이나 집합저작물위주로 그리고 이와 관련하여 차별적인 특허 라이선스를 부여토록 계약을 맺었다면, 그는 저작물을 전달할 수 없다. 단, 본 조항은 제삼자와의 협정 또는 특허 라이선싱이 2007년 3월 28일 이전에 이뤄진 경우 적용되지 아니한다.

본 조항은 GPL 버전 3 채택을 주도한 FSF에 의한 공론 과정 중 기업 간 이뤄질 수 있는 특정 트랜잭션(transaction) 형태가 확산되는 것을 방지하기 위해 채택되었다. SFLC가 인지하는 한, 본 조항과 관련한 컴플라이언스 문제는 현재 존재하지 않는다.

특허 행사 및 보복 GPL 버전 3은 자신이 배포하고 기여한 라이선스 저작물에 대해 특허 소송을 하고자 하는 이들에 대한 특허 보복 제한 조치를 시행하고 있다. 제8조, 제10조, 제11조에서 특허 소송 방어를 위한 특허 중지 및 종료 조건이 명시되어 있다. 제8조 종료 조항은, 만약 라이선시가 GPL 조건을 위반할 경우, 기여자가 라이선시에 부여한 저작권 허가와 함께 동 라이선시에게 제11조에 따라 부여한 특허 라이선스도

종료할 수 있도록 명시하고 있다. 따라서, 기여자는 특허 침해 소송을 개시하여 제10조를 위반한 다운스트림 라이선스에 대해 자신이 부여한 특허 라이선스를 종료할 수 있다.

제12조 타인의 자유 보장

GPL 버전 3 제12조는 기능적인 측면에서 GPL 버전 2 제7조와 동일하다. GPL 버전 2의 조항에 명시된 설명으로 인해 더 많은 혼란이 발생하므로, 해당 부분이 삭제되었다. 컴플라이언스 결과를 포함한 모든 측면에서 살펴봤을 때, 앞서 언급한 두 조항은 동등한 효력을 가진다(18쪽 참조).

제13조 GNU Affero GPL과 병행사용

본 조항은 GPL 버전 3에 따라 라이선스 된 저작물과 AGPL 버전 3 하에 라이선스 된 저작물의 연결 또는 결합을 허가하고 있다. 해당 허가에 대한 컴플라이언스 결과는 아래 AGPL 버전 3 관련 부분에서 다루기로 한다.

연결 및 결합을 통해 만들어진 저작물은 GPL 버전 3의 조건에 따라 관리되나, 해당 저작물이 네트워크상에서 사용되는 경우, AGPL 버전 3의 조건이 해당 저작물에 적용된다.

사용자가 GPL 버전 3의 조건을 선호하거나 그 반대로 선호하지 않을 수 있으나, GNU AGPL 하에 출시된 코드의 경우, 사용자가 이를 수용, 전달, 또는 개작을 할 수 없다. 그러나 단일 프로젝트 내에서 각 라이선스 하에 출시된 분리된 모듈 또는 소스 파일의 경우, 사용자가 이를 결합할 수 있다.

결합저작물이 GPL 버전 3이 적용된 코드와 AGPL 적용된 코드의 연결을 통해 만들어 졌다면, 어느 한 쪽의 카피레프트가 다른 라이선스에까지 확대 적용되지 않는다. 다시 말해, 그와 같은 결합저작물에서, 아페로(Affero)의 의무 사항은 아페로 라이선스 하에 결합을 한 부분에 대해서만 적용이 된다. 해당 결합저작물을 수령하였고 아페로 의무 사항이 적용된 코드의 사용을 원치 않는 경우, 결합저작물에서 아페로가 적용된 부분을 삭제할 수 있다.

GPL 특별 예외 라이선스

GNU 클래스패스 (Classpath)와 GPL 연결 예외

GNU 클래스패스 (Classpath)는 자바 프로그래밍 언어를 위한 GPL이 적용된 필수 라이브러리의 집합이며, GPL과 예외 라이선스는 GNU 클래스패스 (Classpath)를 허가하고 있다.

이러한 라이브러리를 정적 또는 동적으로 다른 모듈과 연결함으로써 해당

라이브러리에 기반을 둔 결합저작물이 만들어진다. 따라서 GNU GPL 조건이 결합저작물 전체에 적용된다.

특별 예외로, 각각 연결된 독립적 모듈에 대해 사용자가 해당 모듈의 라이선스 조건을 준수할 경우, 그러한 독립적인 모듈의 라이선스 조건과 관계없이, 라이브러리의 저작권자는 사용자가 실행물을 생성하기 위해 라이브러리를 독립적인 모듈과 연결하고, 그 결과로 만들어진 실행물을 또한 사용자가 정한 조건으로 복제 및 배포할 수 있도록 허가한다. 독립적인 모듈은 해당 라이브러리에서 파생되거나 해당 라이브러리에 기반을 두지 않은 모듈을 의미한다. 사용자가 라이브러리를 수정했다면, 이 예외를 자신이 수정한 라이브러리 버전에도 확대 적용할 수 있으나, 의무 사항은 아니다. 이 예외 확대 적용하기를 원치 않을 경우, 사용자는 자신의 버전에서 해당 예외 문구를 삭제하면 된다.

AGPL

자유 소프트웨어 커뮤니티는 네트워크 상에서 자유 소프트웨어를 통해 제공되는 서비스에까지 카피레프트 개념을 확대할지 여부에 대해 오랫동안 논의해왔다. 첫 번째 자유(Freedom zero)는 모든 사용자가 어떠한 목적으로든 프로그램을 실행할 수 있도록 허가하고 있으며, 여기에는 물론 타인에게 컴퓨팅 서비스를 제공하는 것도 포함된다. 세 번째 자유(Freedom two)는 개인의 개작 권리 존중을 의무화하고 있다. 이 두 가지

정의에 따라 타인에게 컴퓨팅 서비스를 제공할 목적으로 누구나 자신이 개작한 GPL 프로그램을 실행할 수 있다.

그러나 네트워크 서비스 제공을 위해 사용되는 소프트웨어를, 특히 웹 애플리케이션 인프라, 개발하는 프로그래머들이 자유 소프트웨어 개작 버전을 이용해 서비스를 제공하는 당사자가 원 개발자가 개작자에게 부여한 것과 같은 권리를 서비스 사용자들에게도 부여하도록 의무화하고자 하는 데는 타당한 이유가 있다. 왜냐하면 연합 또는 '자유' 서비스 플랫폼이 이러한 라이선싱 모델의 도움을 받고 있기 때문이다.

FSF는 하나의 라이선스 범위 내에서 문제를 해결하기 보다, GNU GPL과 부분적으로 호환 가능한 '카피레프트 서비스' 소프트웨어의 라이선스 기간을 연장하고자 하였다. 아키텍처는 서로 다르지만, 그 목적은 유사한 두 개의 라이선스 AGPL 버전 1과 AGPL 버전 3이 공표되어 사용되고 있다.⁸ AGPL에서 'A'는 FSF의 디렉터 헨리 풀(Henry Poole)이 시작한 소프트웨어 프로젝트의 이름인 '아페로(Affero)'를 의미한다. 특별히 아페로용으로 FSF가 작성한 AGPL 버전 1은 2003년에 발표되었다. AGPL 버전 3은 이러한 라이선스를 더 긴밀하게 통합하고 AGPL 버전 3이 적용된 프로젝트에서 GPL 버전 3 코드의 사용을 허가하기 위해 GPL 버전 3과 동시에 설계 및 고안되었다.

각 AGPL 라이선스의 공통적인 아키텍처 특징은 카피레프트 의무 사항(GPL 버전 2에 따른 '배포' 및 2차 라이선싱) 및 GPL 버전 3에 규정된 '전달'을 위한 트리거 (trigger: 자동 발동) 조항 외에, 별도의 '트리거' 조항이 있다는 점이다. AGPL 라이선스에 의거한

⁸ 2007년에 발표된 AGPL 버전 2는 과도기적 라이선스 '스터브(stub)'로 모든 라이선서들이 AGPL 버전 3에 따른 AGPL 버전 1 프로그램을 기반으로 한 저작물을 배포할 수 있다.

카피레프트의 범위는 각각의 GPL 버전에 의거한 카피레프트 범위와 동일하다. 소스 코드와 라이선스 본문 제공에 대한 의무 규정 관련 조건만 변경되었다.

AGPL 버전 1

AGPL 버전 1은 제2조 d항에 명시된 추가 규정 하나가 GPL 버전 2에 포함된 것이다.

수령한 AGPL 프로그램이 컴퓨터 네트워크를 통해 사용자와 쌍방향 대화를 하기 위해 고안되었고, 수령한 버전에서 해당 프로그램과 대화하는 모든 사용자가 프로그램의 완전한 소스 코드 전송을 요청할 수 있도록 허가한 경우, 수신인은 이 프로그램을 기반으로 본인이 개작한 버전 또는 저작물에서 해당 내용을 삭제해서는 안되며, 컴퓨터 네트워크를 통해 수신인의 프로그램과 대화하는 모든 사용자가 수신인의 개작 버전 또는 다른 2차적저작물의 전체 소스 코드를 HTTP로 즉시 전송 요청할 수 있도록 해야 한다.

이러한 규정은 GPL 버전 2 제2조 c항의 ‘쌍방향 대화 시 공지의 유지’와 관련한 의무 사항과 비슷한 것으로, 브래들리 쿤 (Bradley Kuhn)이 FSF의 운영책임자 (Executive Director)로 재직할 당시 고안한 것이다. 이 규정을 준수하는 것은 간단하다. 서버 측의 소스 코드 요청 기능이 ‘컴퓨터 네트워크를 통한 사용자 쌍방향 대화를 위한 프로그램’에 명시되어 있으면, 해당 기능을 삭제해서는 안 된다.

AGPL 버전 3

AGPL 버전 3은 GPL 버전 3에 의무를 하나 추가한 것으로, 이 추가 의무사항은 AGPL 버전 3 제13조에 포함되어 있다. AGPL 버전 1과 마찬가지로, AGPL 버전 3도 카피레프트의 어떠한 범위 내에서도 개작을 허용하지 않는다. AGPL 버전 3 제6조에 따라 제공해야 하거나 그 제공의 서면 제안을 해야 하는 ‘부합하는 소스’를 결정할 때, 실제 배포 매체에서 비(非)소스 버전(non-source version)으로 동일한 프로그램이 사용자에게 제공되었다면 어떤 소스 코드가 제공되어야 하는지 고려해 보면 충분히 알 수 있다.

AGPL 버전 3의 추가 트리거 (trigger: 자동 발동) 조항은 다음과 같다.

프로그램 개작을 한 경우, 해당 프로그램의 개작자는 컴퓨터 네트워크 상에서 원격으로 해당 버전과 대화하는 모든 사용자가 소프트웨어 복제를 쉽게 할 수 있는 일부 표준 또는 기존 방법을 통해서 네트워크 서버(만약 해당 버전이 쌍방향 대화를 지원한다면)에 접속하여 무상으로 해당 개작 버전의 부합하는 소스 코드를 받을 수 있도록 해야만 한다. 이러한 부합하는 소스 코드에는 다음 문단에 따라 정립된 GNU GPL 라이선스 버전 3이 적용되는 모든 저작물의 부합하는 소스가 포함되어야 한다.

네트워크를 통해 원격으로 사용자와 대화할 수 있는 아키텍처를 포함하여 개작된 프로그램의 경우, 사용자가 네트워크를 통해 부합하는 서버 측의 소스 코드를 받을 수

있도록 의무화 해야 한다. 모든 카피레프트 저작물에 대해 제안한 바와 같이 해당 소스 아카이브 자체가 프로그램을 구축하는 과정에서의 ‘메이크 타겟(make target)’일 경우, 초기 바인딩(early-binding) 언어를 위한 컴플라이언스 기능을 구축하는 것은 관련 요청이 구동 프로그램에 전달되었을 때, 해당 소스 아카이브(그리고 동일한 시스템의 일부인 다른 구동 프로그램의 다른 소스 아카이브)를 상수 데이터로 하여 커뮤니케이션 스트림(stream)으로 출력하는 것만큼 간단하다. 초기 바인딩 컴파일 언어로 작성된 프로그램의 ‘카피레프트 범위’ 문제가 컴플라이언스에 대한 기술적 접근을 통해 해결된다. 구동된 프로그램의 컴포넌트가 메이크 타임(make time)에서 결정될 경우, 프로그램을 만드는데 사용된 컴포넌트에서 시스템 라이브러리와 다른 예외코드를 제외하면 부합하는 소스가 된다.

GPL 버전 3 제13조와 AGPL 버전 3 제13조 2항은 함께 GPL 버전 3과 APGL 버전 3이 적용된 코드를 결합하기 위한 호환성 구조를 구성한다. GPL 버전 3는 GPL 버전 3 조건에 의거한 코드가 AGPL 버전 3 조건에 의거한 코드와 결합되는 것을 허용한다. 각 부분은 각각의 해당 라이선스의 조건에 따라서 취급된다. AGPL 버전 3 코드는 각 저작권자로부터 추가 허가 없이 GPL 버전 3에 따라 다시 라이선스를 받거나 그 반대로 될 수 없다. GPL 버전 3에 따라 허가된 소스 코드는 이 두 가지 라이선스에 따라 코드를 결합하는 모든 저작물의 부합하는 소스 코드에 포함된다.

LGPL

약소 일반 공중 라이선스(Lesser General Public License, LGPL)의 조건은 다른 저작물 중에서 GNU C 라이브러리에 적용되며 ‘약한 카피레프트’ 라이선스로 표현되기도 한다. 왜냐하면, LGPL 조건이 적용되는 코드는 사유 라이선스(non-free license)가 적용된 코드와 결합할 수 있으며 또한 때때로 결합한 형태로 사용될 수 있기 때문이다. 그러나 LGPL의 두 개 버전이 현재 내용과 구조 측면에서 상당히 다르게 사용되고 있음에도 불구하고, 이 두 개의 버전은 광범위한 연결 예외를 포함하는 강력한 카피레프트 라이선스이다. 이 두 버전의 차이가 빈번하게 잘못 이해되므로, 이 라이선스들을 준수하는 것은 두 버전의 차이를 이해하는 것으로 결정된다. 비(非)카피레프트 라이선스(non-copyleft license)가 적용된 코드와 결합하였을 때, LGPL 코드는 신규 또는 대체 조건에 따라 절대로 입력될 수 없다. 결합은 허가되지만, 프로그램의 LGPL 컴포넌트에는 그 저작권자가 정한 조건이 계속해서 적용된다. 이는 이클립스(Eclipse) 라이선스와는 그 행태가 그리 유사하지 않다. 이클립스 라이선스는 파일 단계에서 카피레프트 조건에 따라 소스 코드 배포를 보장한다. 반면 바이너리는 재배포자의 재량에 따라 어떠한 조건에 하에서도 배포될 수 있다. 각 LGPL 버전은 소스 또는 비(非)소스 형태의 LGPL 코드로 구성된 허가된 결합저작물의 포션(potion)에서 다운로드 사용자의 권리를 보장하고자 한다.

LGPL 버전 2.1

LGPL 버전 2.1은 그 자체로 카피레프트 라이선스이며, 제6조에서 카피레프트가 아닌(non-copyleft) 결합저작물에 대한 예외를 명시하고 있다. 분석적인 특징을 보면 LGPL 버전 2.1 제6조의 예외에서 비롯된 카피레프트 라이선스는 GPL이 아닌 LGPL

버전 2.1 제1조에서 제5조에 해당하며 GPL 각 버전과 마찬가지로 어느 측면에서나 적용하지 않는다.

제0조 정의와 라이선스 범위 LGPL 버전 2.1 제0조는 라이선스 프로그램의 개작 버전을 생성하려면 LGPL이 적용된 프로그램의 다른 프로그래밍 언어 또는 루틴 라이브러리로 ‘바로’ 번역을 하도록 명시하고 있다. GNU GPL 버전 2와 GNU GPL 버전 3에는 이러한 정의가 존재하지 않는다.

제2조 제2조 a항은 저작물이 소프트웨어 라이브러리[‘실행물을 구성하기 위한 애플리케이션 프로그램(이러한 기능과 데이터를 사용하는 일부 프로그램)과 편리하게 연결하기 위한 소프트웨어 기능 그리고/또는 데이터의 집합’이라 제0조에서 정의]이고, 개작된 버전 자체가 라이브러리일 경우에만 개작 버전의 배포를 허가하고 있다. 따라서 LGPL 버전 2.1이 적용된 코드는 라이브러리 맥락으로부터 가져와 라이브러리가 아닌 개작 버전 또는 그 버전에 기반한 저작물에 입력될 수 없다. 제6조는 이러한 규칙에 대한 예외를 인정하지 않는다. 결합저작물은 다른 코드가 포함된 LGPL 라이브러리의 개작된 버전으로 구성되지만, LGPL 코드는 계속해서 라이브러리로 구조화되어야 하며, LGPL 라이선스 조건이 계속해서 라이브러리에 적용된다.

제2조 d항은 라이브러리 루틴의 개작 버전에서 라이브러리 기능을 작동시키기 위해 라이브러리 기능과 관련이 없는 비(非)아규먼트 글로벌 데이터(non-argument global data)를 요청해서는 안 된다고 기술하고 있다. 이는 애플리케이션이 라이브러리

루틴으로부터 서비스를 호출하고 제공받기 위해 입력된 키, 토큰, 테이블 또는 다른 비기능적 제한 데이터를 요구하여 라이브러리가 개작될 수 없음을 의미한다. 제6조는 이러한 의무 사항을 배제하지 않고 있다. LGPL 버전 2.1이 1999년 공표된 이래 그와 같은 개작을 피하도록 의무화하였고 이는 비교적 보편화되었다.

제2조의 이러한 의무 사항은 엄격하며, 단독으로 사용자의 자유에 밀접한 영향을 끼칠 수 있다. 그러나 LGPL 버전 2.1의 제3조는 GPL 버전 2 또는 다른 후속 버전 대신에 LGPL이 적용된 라이선스 조건에 따라 모든 저작물들이 복제되어 사용될 수 있도록 허가한다. 이 규정은 LGPL 버전 2.1의 조건에 따라 코드 사용을 원치 않는 사람들에게 본인의 재량에 따라, GPL 버전 2 또는 GPL 버전 3 하에서 코드를 사용하도록 허가하고 있다.

LGPL 라이선스의 제6조에 기능적인 안티록다운(anti-lockdown) 조항이 포함되어 있는데 특허 또는 비(非)카피레프트 저작물의 일부로 사용자에게 제공된 모든 LGPL 코드는 반드시 라이브러리 개작 버전의 재연결 또는 재설치를 수락해야만 한다.

GPL 버전 2에 기술된 바와 같이 제2조는 아래 내용을 반복해서 기술하고 있다.

본 의무 사항은 개작저작물 전체에 적용된다. 해당 저작물의 특정 부분이 라이브러리를 통해 생성되지 않았거나 그 자체로 독립적이거나 별도의 저작물이라고 합당하게 인정이 될 경우, 수신인이 해당 부분을 별도의 저작물로 배포할 때, 본 라이선스 및 그 조항은 적용되지 않는다. 그러나 라이선스 적용된 라이브러리를 기반으로 한 저작물의 일부로 동일한 부분이 배포가 될 때는 전체

저작물 배포는 LGPL 라이선스의 조건을 준수해야 하며, 다른 라이선시를 위한 본 라이선스의 허가는 전체 저작물에 확대 적용된다. 따라서 누가 해당 저작물을 작성했는지와 상관없이 각각 모든 부분에 적용된다.

제2조의 내용을 고려할 때, 이 조항은 제6조에 따라 결합저작물의 예외를 제한하는 방식으로 해석되지 않는다. 그러나 LGPL 저작물 자체의 개작 버전과 관련하여, 제2조는 제6조가 적용되지 않음을 재확인한다.

제4조 제4조는 저작물 원본의 비(非)소스 형태의 배포와 제2조에 따른 개작저작물을 다루고 있다. 해당 조항을 준수할 경우, 앞서 언급한 GPL 버전 2도 같이 준수하는 것이 된다.

제5조 제5조는 소프트웨어 저작권법의 적용에 있어 모호한 점을 명확히 기술하고 있다.

해당 라이브러리의 일부에서 파생되지 않았지만, 컴파일 되거나 연결되어 해당 라이브러리와 함께 작동하도록 설계된 프로그램을 ‘해당 라이브러리를 사용하는 저작물’이라고 칭한다. 이러한 저작물은 단독으로는 라이브러리에서 파생된 2차적저작물이 아니기 때문에, 본 라이선스 범위에 적용되지 않는다.

그러나 ‘라이선스 적용된 라이브러리를 사용하는 저작물’과 해당 라이브러리의 연결을 통해 실행물이 생성되는데, 이는 ‘라이선스 적용된 라이브러리를 사용하는 저작물’이라기보다 라이브러리에서 파생된(라이선스 적용된 라이브러리의 일부를 포함하기 때문) 2차적저작물이다. 따라서 본 라이선스가

적용이 되며, 제6조에서 그와 같은 실행물의 배포 조건을 기술하고 있다.

‘해당 라이브러리를 사용하는 저작물’이 해당 라이브러리의 일부인 헤더 파일의 자료를 사용하는 경우, 소스 코드가 아니라도 그 저작물의 오브젝트 코드는 해당 라이브러리의 2차적저작물로 간주된다. 이는 저작물이 해당 라이브러리 없이 연결될 수 있는지 저작물 자체가 라이브러리인지 여부에 따라 판단할 수 있다. 2차적저작물 판단 기준은 법으로 명확하게 정의되어 있지 않다.

그렇기 때문에 모호한 상황의 발생을 피하기 위해, 제5조는 다음과 같이 기술한다.

오브젝트 파일이 숫자 파라미터, 데이터 구조 레이아웃과 접근자(accessors), 작은 매크로와 인라인 함수(10행 이하)로만 구성 되었을 경우, 법적으로 2차적저작물인지 여부와 관계없이 해당 오브젝트 파일의 사용은 제한되지 않는다(실행물이 이 오브젝트 코드와 라이브러리의 일부를 포함하고 있을 경우에는 제6조가 계속해서 적용된다).

그러나 저작물이 라이브러리의 2차적저작물일 경우에는, 제6조에 따라 그 저작물의 오브젝트 코드를 배포해야 한다. 라이브러리와 직접적으로 연결되었는지 여부에 관계없이 해당 저작물을 포함하는 모든 실행물에도 제6조가 적용된다.

본 조항의 목적은 제6조에 기술된 예외에 따라 결합물 수준까지 확장된 모호한 카피레프트에 대한 권리를 포기하기 위함이다. 우리가 아는 한, LGPL 버전 2.1 실행과

관련, 제5조에 명시된 분야에 대해 소송이 제기된 적은 한번도 없었다.

제6조 제6조는 ‘약한 카피레프트’가 아님에도 빈번하게 그렇게 잘못 간주되는
결합저작물의 예외를 기술하고 있다. 이 조항은 다음을 허가하고 있다.

만약 라이선스 조건에 따라 고객 본인을 위한 개작과 개작을 디버깅하기 위한
리버스 엔지니어링(reverse engineering)이 허용한다면, 해당 라이브러리의 일부를
포함하는 저작물을 생성하고 그 저작물을 본인이 정한 조건에 따라 배포하기
위해 라이선스 저작물 또는 그에 기반한 저작물의 결합저작물이 허용된다.

라이선스가 적용된 결합저작물과 관련된 이러한 의무 사항에 따라, 선택된 라이선스는
저작물 전체에서 고객의 개작과 이러한 개작저작물을 디버깅하기 위한 리버스
엔지니어링을 금해서는 안 된다. 실제로 과거 사례를 살펴보면, 선택된 라이선스 조건은
결합저작물에 포함된 LGPL 코드에서 개작과 개작저작물의 디버깅을 위한 리버스
엔지니어링을 금지하지 않고 있다.

제6조는 또한 다음 사항을 규정한다.

해당 라이브러리가 사용된 저작물의 각 복제본 마다 해당 라이브러리와 그
라이브러리의 사용은 본 라이선스의 적용을 받는다는 내용을 명확히 표시해야
한다. 그리고 본 라이선스의 사본을 제공해야 한다. 저작물을 실행할 때 저작권
공지가 표시되었다면, 그 중에 라이브러리의 저작권 공지와 본 라이선스의

사본의 위치를 사용자가 확인할 수 있는 레퍼런스를 포함시켜야 한다.

이는 GPL 버전 2 및 GPL 버전 3의 의무 사항과 동일하지 않다. 대화형 프로그램에서 ‘크레딧(credits)’ 또는 ‘라이선스’ 또는 ‘어바우트(about)’ 화면과 관련해 상이한 컴플라이언스 의무가 존재한다.

또한 제6조에 따라 결합저작물에서 LGPL 적용 부분에 소스 코드가 제공되거나 ‘공유 라이브러리’ 메커니즘이 사용되어 개작 버전에 의해 라이선스 저작물의 동적인 교체가 가능하도록 해야 한다.

만약 소스 코드가 제공되었다면, GPL 버전 2와 GPL 버전 3에서 벗어나는 규정을 기술하는 제6조 a항이 적용된다.

저작물에 사용된 모든 변경사항을 포함하여 컴퓨터가 판독할 수 있는 전체 소스 코드를 저작물과 함께 제공해야 한다(저작물은 상기 제1조와 제2조에 따라 배포). 저작물이 라이브러리와 연결된 실행물일 경우, 컴퓨터가 판독할 수 있는 전체 ‘라이선스 적용된 라이브러리를 사용한 저작물’ 전체와 함께 오브젝트 코드 그리고/또는 소스 코드로 제공한다. 그렇게 되면 사용자가 라이브러리를 개작하고 개작된 라이브러리가 포함된 개작 실행물을 생성하기 위해 라이브러리를 재연결 할 수 있다.

완전하고 부합하는 소스 코드가 전부 제공되어야 할 뿐만 아니라, 사용자에게 원래 제공되었던 버전을 대체하는 LGPL 저작물의 개작 버전을 가지고 전체 결합저작물의

개작버전이 재생성 될 수 있는 방식으로 제공되어야 한다. 예를 들면, LGPL 코드가 정적으로 비(非)카피레프트 실행물과 연결되었을 경우, 필수 소스 코드에는 배포된 실행물을 분리하고 라이브러리의 개작된 버전을 재연결(relink) 할 수 있는 충분한 자료가 포함되어야 한다.

제6조 c항에 따라, 해당 소스 코드는 직접 제공되지 않고 서면을 통해 그 제공 의사를 표시할 수 있다. 또한 제6조 d항의 조건에 따라 네트워크를 통해 배포 될 수도 있다.

제6조 e항에 따라 사용자가 이미 소스 코드를 수령했는지 또는 적어도 배포자가 이미 이 사용자에게 관련 소스를 전송했는지 ‘확인’ 해야 한다. 이는 ‘전체 배포’를 할 때 중복으로 소스 코드를 전송하지 않기 위해서이다.

결합저작물의 배포자가 LGPL 컴포넌트의 소스 코드를 배포하거나 소스 코드 제공을 서면으로 제안할 의도가 없다면, LGPL 저작물은 별도로 다음의 ‘공유 라이브러리’ 메커니즘으로 배포(분리 배포의 경우 소스 코드 배포 의무 대상이 됨) 및 패키징 되어야 한다. 이는 다음을 의미한다.

- (1) 공유 라이브러리 메커니즘은 라이브러리 기능을 실행물에 복제하는 것이 아니라, 사용자의 컴퓨터 시스템에 이미 설치되어 있는 라이브러리의 복제본을 런타임에 사용하는 것을 의미하며, (2) 개작 버전이 저작물을 만들 때 사용되었던 라이브러리 버전과 인터페이스 호환이 될 경우, 사용자가 라이브러리를 설치하면 개작된 라이브러리 버전과 정상적으로 구동된다.

전체적으로, 이 조항이 의미하는 바는 다음과 같다.

1. LGPL 버전 2.1에 따라 별도로 배포되는 저작물과 공유 라이브러리 메커니즘을 통해 연결되는 프로그램을 만들었다면, 수신인이 선택한 라이선스에 따라 해당 프로그램을 배포할 수 있으며, LGPL 저작물인 소스 코드를 전달할 필요가 없다. 라이브러리를 프로그램과 함께 배포하거나, 다른 맥락이나 제품으로 라이브러리를 별도로 배포하는 경우, 수신인의 선택에 따라 LGPL 버전 2.1 또는 GPL 버전 2+의 조건하에서 부합하는 소스를 배포해야 한다.

2. LGPL 저작물과 정적으로 연결되거나 수신인의 프로그램이 LGPL 저작물과 결합될 경우, LGPL 컴포넌트에 LGPL 조건이 계속해서 적용된다는 것을 고려할 때, 사용자의 개작, 리버스 엔지니어링 및 디버깅에 관한 라이선스 조건이 제한사항을 충족한다면, 수신인은 저작물에 대한 라이선스를 직접 선택할 수 있다. 수신인은 LGPL 컴포넌트에 대한 완전하고 부합하는 소스 코드를 제공해야 한다. 제공된 소스 코드 자료는 LGPL 컴포넌트의 사용자 개작 버전으로 결합저작물을 재생성 하기에 충분해야 한다.

LGPL 버전 2.1은 상대적으로 잘 지켜지지 않고 있다. 아마도 읽으면서 쉽게 간과되는 조건에 대해 잘못된 인식을 하고 있는 경우가 많기 때문일 것이다. 우리는 사용자를 대신해서 자신의 권리는 실행하는 일부 라이선서를 변호한다. 공공 분쟁은 빈번하게 발생하지 않는다. 왜냐하면 라이선스 조건을 잘 준수해야 하는 당사자들은 정중하지만 단호하게 요청 받았을 경우 라이선스 조건을 잘 준수하기 때문이다.

LGPL 버전 3

LGPL 버전 3은 GPL 버전 3 결합 예외인 LGPL 버전 3에서 카피레프트 라이선스를 제작하여 GNU 계열 라이선스 아키텍처 계획을 수정하기 위해 고안되었다. 따라서 LGPL 버전 3은 상기 GPL 버전 3 제7조에 명시된 조건에 대한 추가적인 허가를 의미한다.

제0조 추가 정의 제0조는 '라이브러리'를 정의한다. 라이브러리는 '애플리케이션'에 의해 '사용' 될 수 있는 하나 이상의 인터페이스가 있는 저작물에 적용된다. 클래스 상속은 인터페이스를 사용하는 것으로 '간주된다(deemed).' 하나 이상의 '라이브러리' 인터페이스를 사용하는 다른 프로그램 코드인 '애플리케이션'은 '결합저작물'을 구성하기 위해 사용된 다른 쪽 인터페이스의 코드와 결합될 수 있다.

제1조 GNU GPL 제3조의 예외 제1조는 GPL 버전 3의 제3조에 명시된 다른 저작권이 적용된 저작물의 접근 제한에 대한 '효율적인 기술 방안'의 하나로 LGPL 버전 3 적용 코드 사용에 대한 간섭을 예외로 하고 있다.

제2조 개작 버전의 전달 제2조는, LGPL 버전 2.1 제2조 d항과 마찬가지로, 라이브러리가 키, 토큰, 테이블, 또는 기능과 상관 없는 기타 글로벌 비(非)아규먼트 데이터를 요청하도록 개작하는 것을 금하고 있다. 이는 '선의의 노력'에 대한 의무 사항으로 한번 더 기술하고 있는 것이다. 그러나 공지에 따라 제대로 조치하지 않는 것은 선의의 노력이 제대로 실천되지 않음을 보여주는 강력한 증거이다. 추가적 허가를 제외하고

GPL 버전 3 조건을 적용하는 것은 GPL 버전 3 제7조에 기술된 바와 같이 컴플라이언스의 대안이 된다.

제3조 라이브러리 헤더 파일 자료를 포함하는 오브젝트 코드 제3조는 저작권 범위(라이브러리가 프로그램에서 사용되고 있으며, GPL 버전 3과 LGPL 버전 3의 복제본이 저작물과 함께 제공됨을 공지) 내에서 사용자의 재량에 따라 적용되는 규정을 명시하고 있는 LGPL 버전 2.1 제5조에 따른 헤더 파일의 사용과 라이브러리 자료의 다른 형태의 사용에 있어 불분명한 부분을 배제하고 있다.

제4조 결합저작물 제4조는 LGPL 버전 3의 핵심인 결합저작물 허가와 관련해 기술한다. 이 조항에서 사용자가 라이브러리 코드를 개작하거나 그러한 라이브러리 코드의 개작을 위해 필요한 리버스 디버깅의 방법으로 디버깅하는 것을 금지해서는 안 된다는 것을 명확히 하기 위해 LGPL 버전 2.1 제2조의 라이선스 제한 조항을 다시 한 번 기술하고 있다.

제4조 d-0항은 최소한의 부합하는 소스를 제공할 것을 의무화 하고 있다. “이는 결합저작물의 부합하는 소스 코드이며, [라이브러리의] 연결 버전이 아닌 애플리케이션 기반의 독자적인 결합저작물의 일부분에 대한 소스 코드는 포함하지 않음을 의미한다.” 상기 LGPL 버전 2.1과 관련하여 기술된 제4조 d-1항에 따라 ‘공유 라이브러리’ 메커니즘 방식으로 배포하는 것은 소스 코드 제공의 대안이 된다.

또한 제4조 e항은 GPL 버전 3 제6조에 따라 ‘사용자 제품’에서 개작된 라이브러리 버전을 설치하기 위해 ‘설치 정보’ 제공을 의무화 하고 있다. 제4조 d-1항에 의거하여

라이브러리에 대해 최소한의 부합하는 소스 코드가 제공될 수 없다면, GPL 버전 3 제6조에 따라 제4조 e항은 계속해서 '설치 정보'가 규정에 따라 제공되어야 함을 재확인하고 있다.

GPL 버전 3의 다른 조항은 이전에 기술된 바와 같이 유효하며, LGPL 버전 3에 의해 부여된 추가 허가로 인해 예외 사항으로 간주되지 않는다.

컴플라이언스 책임에 대한 이해

상기 섹션에서 왜 자유 소프트웨어를 사용하는 개발자들이 GNU 계열 카피레프트 라이선스를 사용해야 하는지 설명하였다. 그리고 라이선서의 의도와 목적과 관련하여 해당 라이선스의 조항에 대해 검토한 바를 자세히 기술 하였다. 다음 섹션에서는 카피레프트 프로젝트의 '다운스트림' 당사자의 컴플라이언스에 대해 논의할 것이며, 사용자들의 소스 코드 제공 관련 문의와 저작권자의 컴플라이언스 소송에 대응하기 위한 일반적인 지침을 제공할 것이다.

컴플라이언스 의무 주체는 누구인가?

이미 설명했듯이 라이선스는 제약 없이 어떠한 목적을 위해서라도 프로그램을 실행하고, 복제하고, 수정하고, 공유할 수 있는 사용자의 권리를 보호하는데 그 초점을 맞추고

있다. 라이선스 기안자들과 라이선서는 라이선스를 사용하는 다운스트림 사용자의 권리를 보호하기 위해 최소한으로 필요하다고 간주하는 범위에서만 컴플라이언스를 엄격하게 의무화 하고 있다. 카피레프트는 그 시스템의 핵심이다. 업스트림에서 자유롭게 제공된 카피레프트 코드 기반 저작물 전체에는 ‘균등한 공유’ 의무가 부과된다. 이 라이선스의 보조 조항은 ‘균등한 공유’ 원칙이 약화되거나 제대로 준수되지 않는 것을 방지하기 위하여 작성된 것이다. GNU 라이선스는 기존의 경제 용어에서 ‘커먼즈(communs)’를 창조했는데, 이는 라이선스에 의해 부가된 제한이 어업 할당량 또는 토지사용 시 지하수 보존 규정과 같은 커먼즈를 위한 ‘거버넌스’ 또는 ‘관리’ 규칙을 의미한다. 커먼즈에 대한 당사자의 의무는 그 자원과 관련된 역할에 따라 각각 달라진다. 상세한 역할과 관련 의무는 다음과 같다.

1. 라이선스에 따라 코드 사용자는 의무를 지지 않고 권리만 가진다. 코드 사용자는 저작물의 특정 라이선스 범위에 이르는 모든 라이선스 코드를 읽고, 연구하고, 이해하고, 실행하고, 개작하고, 재사용하고 변형시킬 수 있으며, 의무 사항에 적용할 예외가 존재하면 이를 알린다. 이러한 권리를 행사할 때, 저작물의 완전하고 부합하는 소스 코드를 보유하고 사용할 수 있으며, 이러한 소스 코드는 이는 부수적인 스크립트, 생성 파일 등과 함께 가장 쉽게 개작할 수 있는 형태로 프로그램의 개작 버전과 모든 관련 설치 정보를 구축하기 위해 필요한 각각의 관련 컴퓨터 프로그램을 의미한다. 왜냐하면 사용자들이 이러한 권리를 알지 못하면 이 권리는 소용이 없기 때문에, 사용자들은 프로그램에 적용되는 조건에 대한 실제 공지를 받아야 한다.

2. 네트워크를 통한 서비스 제공자들은 – 네트워크를 통해 GNU AGPL에 따라

이러한 서비스를 제공하도록 특별히 설계된 컴퓨터 프로그램을 사용할 때 - 서비스 사용자에게 대해 상기 기술된 의무를 가지게 된다.

3. 라이선스 저작물의 배포자는 - 개작 버전을 배포하든지 원본을 판매하든지 디지털 복제본을 전송하든지 - 본인 또는 중간 배포자가 배포한 복제본을 받은 사용자에게 대해서는 적어도 의무를 지닌다. 라이선스 저작물을 직접적으로 배포 받지 않은 제삼자에게 대해서도 이러한 의무를 가지는지 여부는 관련된 특정 라이선스와 배포자가 선택한 소스 코드 배포 또는 제공 모드의 선택에 따라 결정된다. 또한, 배포자는 업스트림 당사자에게 대해서도 코드에 입력되어 있는 타당한 법적 공지를 보호하고 개작 버전을 적절하게 표시할 의무를 가진다.

4. 업스트림 당사자에게 추가적인 제약이 부과되는 것을 방지하기 위해 서비스 제공자와 배포자 모두 사용자 권리 보호 의무를 지닌다. 제공자 및 배포자는 '포괄적 라이선스' 및 EULA의 조건을 제한하거나 상기 기술된 다운스트림 사용자의 권리를 제약하는 2차 라이선스 적용해서는 안 된다. LGPL의 조건에 따라, 더 큰 비(非)LGPL이 적용된 저작물의 라이선스 컴포넌트를 교체하는 것을 금지하거나, LGPL이 적용된 컴포넌트에서 버그를 수정하고 향상시키기 위한 디컴파일이나 리버스 엔지니어링을 금지하는 라이선스 저작물에 기반한 저작물 관련 라이선스 조건을 사용해서는 안 된다.

5. 배포한 저작물을 읽을 수 있는 권리를 보유하고 있는 특허권자는 본인들이 배포한 저작물을 수령한 이들에 대해서 이러한 권리를 행사해서는 안 된다. 버전 3 계열 라이선스에 따라 저작물을 개작하고 배포한 특허권자는 배포한

버전뿐만이 아니라 이러한 권리를 실행하는 후속 버전 또는 그 후속 버전에 기반한 저작물에 대해 본인들이 배포한 버전을 읽을 권리를 행사해서는 안 된다.

6. 기술된 바와 같이, 사용자에게 대한 컴플라이언스를 저해하는 법적 행위, 법적 효력이 있는 조건에 따라 의무를 수용하거나 법이 이를 부여한 경우, 해당 당사자는 서비스 제공자나 라이선스 저작물의 배포자가 되어서는 안 된다. 당사자에게 이처럼 서로 상충하는 의무가 있는 경우, 해당 당사자는 라이선스 의무 수행을 하지 말아야 한다.

컴플라이언스 의무 충족 방법

GPL 준수 프로그램을 구축하면서 상거래 당사자들과 일한 경험과 GPL 라이선스 보유자들이 컴플라이언스 의무 불이행으로 인한 결과에 대응할 때 이들을 대변하는 변호사로서의 역할을 한 경험을 통해, 컴플라이언스에 대한 비즈니스의 예측과 뭔가 잘못되어 분쟁이 발생하고 이러한 분쟁을 해결하는 현실 간에는 상당한 불일치가 있음을 확인하였다. 지속적으로 지켜봤을 때, 법적 소송으로 이어질 수 있고 다른 부정적인 결과를 야기할 수 있는 리스크는 제대로 관리되지 않고 있는 반면, 과거에 발생한 적이 거의 없고 해결하는데 비용이 거의 들지 않는 리스크를 피하기 위해 막대한 경비를 들여 준비하는 기업을 흔히 보았다. 본 섹션에서는 기업들이 최소한의 노력과 비용으로 실제로 직면할 수 있는 컴플라이언스 관련 리스크를 예방하면서 관련 컴플라이언스 의무를 다하도록 지원하기 위한 활동에 대해 개괄적으로 기술하도록 한다.

경험을 통해, 실제 컴플라이언스 및 컴플라이언스 리스크 관리 사이의 불일치는 라이선서의 의도를 잘못 이해하여 발생한다는 것을 확인했다. 영리 추구 당사자는 카피레프트 프로젝트 커뮤니티가 저작권을 통해 수익을 창출하거나 카피레프트의 조건 하에 특허 소프트웨어에 다시 라이선스를 부여할거라 여기고 카피레프트 컴플라이언스를 다룬다. 라이선서가 로열티를 받기 위해 규정 불이행을 통해 이익을 얻으려는 의도가 있거나, 카피레프트에 따라 기업의 특허 제품이 배포되도록 강제할 의도가 있다는 가정 하에, 기업은 ‘우발적으로’ – 또는 감독 받지 않는 개인 프로그래머의 행동의 결과로 – 카피레프트 코드에서 규정을 준수하지 않은 코드 조각(snippet)을 자사의 특허 컴퓨터 프로그램에 복제하게 될 리스크를 관리하고 있다. 리스크 관리를 위해 우발적으로 이러한 코드가 포함되는 것을 방지하고자 고가의 특허 ‘코드 스캐닝’ 서비스를 구매한다. 자유 소프트웨어에 의한 ‘감염’을 예방하기 위해 특허 컴퓨터 프로그램이 어떻게 작성되었는지에 그 노력이 집중되어 있다.

그러나 사실, 카피레프트를 사용하는 개발 커뮤니티는 컴플라이언스 불이행을 규정을 더 잘 준수할 수 있는 기회로 간주한다. 프로젝트에서 모든 다운스트림의 컴플라이언스 불이행은 사용자의 권리 손실을 의미한다. 프로젝트는, 저작권자로서, 사용자의 권리를 보호한다. 또한 그들의 활동은 해당 권리를 보호하고 소프트웨어를 만들 때 프로젝트 참여자의 의도를 보호하기 위해 고안되었다. 컴플라이언스를 위한 프로젝트의 목표는 특허 및 자유 소프트웨어의 결합저작물이 생성되는 방식보다는 소프트웨어가 사용자들에게 제공되는 방식에 의해 더 자주 좌절된다. 특히 아래와 같은 경우가 그러하다.

- 구매한 제품에 입력된 카피레프트 프로그램 및 관련 라이선스 조건에 대한 필요 정보가 사용자에게 제공되지 않음
- 배포자가 알고 있는 라이선스 조건에 따라 사용했고 사용해야 하는 카피레프트 프로그램의 완전하고 부합하는 소스 코드를 사용자가 받지 못함
- 사용자가 공개된 주소로 비즈니스 의무사항을 준수할 것을 요청했는데 응답을 받지 못함

이러한 경우 및 이와 유사한 상황에서, 프로젝트의 목표는 다운스트림 사용자에게 대한 의무를 다함으로써 카피레프트 프로그램의 의도적인 사용을 통해 발생한 의무를 준수하는 것이다. 이러한 유형의 컴플라이언스 불이행은 스캐닝 프로그램이나 다른 유사한 서비스로는 발견되지 않으며 전세계 카피레프트 커뮤니티에 의해 모두 소송으로 이어졌다.

상업적 특허 제품에 자유 소프트웨어가 포함되는 경우가 발생하고 있다. 카피레프트를 사용하는 개발 커뮤니티를 대신하여 업무를 하면서, 빈번하지는 않지만 자주 이러한 문제에 직면하고 있다. 우리가 아는 한, 이러한 문제에 대해 커뮤니티는 한번도 컴플라이언스 불이행 소송을 제기한 적이 없다. 이러한 문제는 원만하게 상호협력을 통해 꾸준히 해결되고 있다.

컴플라이언스의 핵심, 거버넌스

‘소프트웨어 거버넌스’는 비즈니스가 수용·배포한 소프트웨어, 인바운드 및 아웃바운드 트랜잭션에 적용 또는 제공한 라이선스를 문서화하고 제어하는 프로세스를 위해 사용한다. 기업이 소프트웨어가 탑재된 실제 제품 또는 소프트웨어 제품 및 서비스를 판매하는지 간에, 훌륭한 소프트웨어 거버넌스는 최소 비용으로 컴플라이언스 의무를 충족시켜야 한다.

가상의 가전제품 제조사 Acme사를 예로 들어보자. Acme사는 안드로이드를 – 카피레프트 OS 커널(kernel), 리눅스 및 LGPL 라이브러리와 일부 GPL 유틸리티가 포함된 커스텀 애플리케이션 포함 – 일부 기기에 탑재 한다. Acme사는 커스텀 소프트웨어 스택을 – GPL 버전 2에 따른 카피레프트 커널, 개작된 리눅스 비지박스(busybox) 그리고 GPL 버전 3에 따른 GNU 유틸리티 묶음 포함 – 다른 기기에 탑재 한다. Acme사는 일부 자사 제품을 재판매한 중간업자에게 디지털 파일로 펌웨어 업그레이드를 제공한다. Acme사는 제품 패키지나 문서에 소스 코드 제공의 서면 제안을 포함시키는 대신에 제품과 함께 GPL 소스 코드가 들어있는 CD와 USB 메모리 스틱은 제공하지 않기로 결정했다. 그 결과, Acme사는 모든 요청자에게 자사 제품 전체의 카피레프트 프로그램 관련 완전하고 부합하는 소스 코드를 제공해야 하는 의무를 가지게 된다.

이러한 의무를 다하기 위해, Acme사의 거버넌스 프로세스는 각 제품의 소프트웨어 스택의 컴포넌트 파트 목록인 ‘제품 구성 사양(bill of material)’ 부터 시작한다.

훌륭한 FOSS 툴이 있는 프로세스인 ‘라이선스 분석’을 통해, 개발을 하기 전에 제품 구성 사양 내의 모든 아이템에 라이선스가 적용이 되어 있는 지 확인하고, 개발이

진행되는 중에 반복해서 제품 구성 사양을 수정한다. 제품 문서에서 다시 제작해야 하는 라이선스 본문은 라이선스 분석의 결과를 통해 구축되며 제품 패키징과 온라인 문서에 포함된다. Acme사의 모든 카피레프트 소프트웨어는(하드웨어 컴포넌트에 내장이 되었거나 제삼자 소프트웨어 공급자가 제공) 자동적으로 완전하고 부합하는 소스 코드를 요청한다. 사용자로서 권리를 행사함으로써, Acme사는 부여된 의무를 다하고 공급체인 내에서 법인의 컴플라이언스를 개선한다.

카피레프트 라이선스에 따라 Acme사의 모든 소스 코드는 구축 프로세스를 통해 실행, 수정 또는 그대로 사용된다. 첫 번째 시험 구축에서 출시를 위한 최종 구축 및 생산에 이르는 모든 단계에서, 카피레프트 바이너리에 대한 완전하고 부합하는 소스 코드는 구축 목표의 일부이다. 이러한 소프트웨어 스택을 포함하는 제품이 출시되었을 경우, 카피레프트 컴포넌트에 대한 완전하고 부합하는 소스 코드는 제품 모델 번호, 개정 코드 또는 해당 기기와 소프트웨어 버전의 임의 해시 코드로 구분되어 공공 웹사이트를 통해 제공된다. 이 ‘스택마크’는 또한 제품 문서에 인쇄되어 있으며 외부 제품 패키지에는 QR 코드가 표시되어 있다. 유사한 스택마크는 디지털 파일로 제공되는 펌웨어 업그레이드와 함께 판매 중간업자에 전달되어 그들의 웹사이트를 통해 제공된다.

제품 출시 전 법적 검토를 통해 Acme사는 자사가 의무를 잘 준수하고 있는 지 확인한다. Acme사가 자사 제품의 전체 소프트웨어와 자사 제품 관련 ‘서비스로서의 소프트웨어’ 제공을 위해 작성한 서비스 계약 일체를 이행하기 위해 적용한 EULA 또는 포괄적 라이선스를 확인하여 전체 제품 소프트웨어 스택 내 모든 FOSS 라이선스의 조건을 잘 준수하고 있는 지 살피고, 카피레프트의 위반을 초래하는 추가 제약사항이 부과되지는 않는지 확인한다.

소프트웨어 거버넌스 프로세스에서 이러한 단계를 실행함으로써, Acme사는 GNU 계열 라이선스에 따라 자신의 권리를 실행하고자 하는 모든 자사 제품 사용자가 심지어 박스를 열지 않고도 소스 코드를 요청하여 즉각적으로 받을 수 있음을 보장할 수 있다.

Acme사는 또한 자사 제품의 각 구매자가 관련 카피레프트 라이선스의 사본을 제공받고 자사 제품에 카피레프트 소프트웨어 관련 사용자의 권리를 침해하거나 그 권리를 무효화하는 특허 제품의 라이선스가 없다는 것을 최소의 노력과 비용으로 보장할 수 있다.

이러한 단계를 거친다고 해서 Acme사가 제품 스택의 일부로 배포한 특허 프로그램에서 카피레프트 코드가 우발적으로 포함되는 것을 반드시 피할 수 있는 것은 아니다. 그러나 Acme사는 커뮤니티 당사자가 항상 소송을 제기하는 이유인 컴플라이언스 불이행을 스스로 예방할 수 있다.

준비된 컴플라이언스의 원칙

물론 현실은 Acme사의 경우처럼 그리 간단하지 않다. 자동차에서 가전제품, 네트워크, 또는 엔터테인먼트 기기에 이르기까지 오늘날의 제품에는 카피레프트 FOSS, 다른 FOSS, 또는 특허 프로그램 등이 탑재된 소프트웨어 스택이 들어있는 수십 개의 컴퓨터가 포함되어 있다. 그러나 이러한 복잡한 영역에도 몇 가지 유효한 기본 원칙

존재한다.

1. 라이선스를 통해 전달된 권리를 행사하고자 하는 다운스트림 사용자의 포지션에서 컴플라이언스를 측정한다. 의도적으로 제품에 포함시킨 카피레프트 소프트웨어에 대한 공지를 사용자가 받았는가? 완전하고 부합하는 소스 코드와 관련 설치 정보를 가급적이면 자동적화된 수단을 통해 쉽게 사용자가 받을 수 있는가? 구축한 것만 측정하는 툴보다 배포(deliver)한 것을 측정하는 툴이 더 유용한가?

2. 가급적이면 전체 소프트웨어 거버넌스 시스템에 직접 입력하는 자동화된 프로세스를 통해 항상 소프트웨어와 소프트웨어에 임베드된 컴포넌트의 모든 제공자로부터 전체 카피레프트 저작물의 완전하고 부합하는 소스 코드를 요청할 권리를 항상 행사한다. 가능하다면, 요청을 해도 완전하고 부합하는 소스 코드가 제공되지 않거나, 소스 코드의 자동 프로비저닝을 위한 ‘스택마크(stack mark)’가 없는 카피레프트 소프트웨어가 포함되어 제공된 규정을 준수하지 않는 컴포넌트는 사용하지 않는다. 업스트림 제공자에게 소프트웨어를 의존하고 있는 경우, 단순히 자신이 배포한 소프트웨어를 타인이 패키징 했다는 이유만으로 GPL 컴플라이언스 의무를 무시할 수 없다.

3. 사용하고 있는 카피레프트 소프트웨어에 집중한다. 일부 프로그래머가 부주의로 특허 제품에 입력한 카피레프트 코드 조각을 통해 발생할 수 있는 리스크는 예전부터 발생 빈도도 낮았고 해결하기도 수월한 편이었다. 더 우려되는 리스크를 효율적으로 관리하고자 한다면, 2년 전 출시된 제품에

코어부트 부트로더(Coreboot bootloader), 리눅스 커널(Linux kernel), 비지박스(Busybox) 또는 GNU tar를 복제하기 위한 부합하는 소스 코드 및 생성 파일이 올바르게 제공됐는지를 확인해야 한다.

4. 코드 스캐너는 거버넌스를 향상시키기 위한 프로세스에서 너무 늦게 작동하고 배포와 포스트 세일 프로비저닝(post-sale provisioning)에서 문제를 파악하기 위한 프로세스에서 너무 일찍 작동하므로, 맹목적으로 의존하지 않는다. 코드 스캐너는 덜 중요한 부분을 처리하는 데 많은 비용이 들며 더 중요한 부분은 전혀 처리하지 않는다. 코드 스캐너는 거버넌스와 검증 프로세스 보완 목적으로 비용 효율적으로 사용될 수 있는 부분에서 사용해야 하며 리스크 관리를 위한 주요 툴로는 사용하지 말아야 한다.

컴플라이언스 관련 문의 처리

우리끼리 얘기지만, 저작자들이 GPL을 집행하기 위해 거의 30 인년(person-year)을 소비했다. 우리는 개별적 또는 집단적으로 SFLC를 통해 미국에서 소송이 제기된 모든 커뮤니티의 GNU 카피레프트 라이선스의 집행에 참여하고 있다. 또한 소송 단계까지 이른 수 십 건의 컴플라이언스 관련 분쟁을 해결해왔다.

이러한 과정에서 이러한 분쟁이 상호 오해로 인한 결과라는 사실을 확인하게 되었다. 커뮤니티 당사자는 규정을 준수하도록 하기 위해 컴플라이언스 불이행에 대한 소송을

제기한다. 기업은 컴플라이언스 관련 분쟁이 금전적 요구나 특허 소프트웨어와 관련 기업 비밀을 침해하고자 하는 시도라 생각하고 방어적으로 대응한다. FOSS의 소프트웨어 엔지니어링 업무방식에 익숙한 여러 커뮤니티 당사자는 때때로 자사 제품에 포함된 카피레프트 프로그램에 대한 완전하고 부합하는 소스 코드를 신속하게 생성하지 못하는 기업이 고의로 혼란을 일으키고 있다고 간주한다.

경험상 봤을 때, 초기 단계에서 당사자 간의 커뮤니케이션을 강화함으로써 이러한 오해가 확대되는 것을 방지할 수 있다. 몇몇 지침을 예전 사례와 함께 제시하면 도움이 될 거라 생각한다.

1. *책임자에게 전화를 연결한다.* 컴플라이언스 불이행 관련 소송을 성공적으로 처리하는 가장 중요한 하나의 방법은 커뮤니케이션을 계속하는 것이다. 내부 소프트웨어 거버넌스 메커니즘을 이해하는 사람에게 FOSS 컴플라이언스 관련 문제를 맡기고 이러한 문제가 발생했을 경우, 커뮤니티를 위한 주요 연락책으로 활용하는 것은 문제를 해결하는 가장 효율적인 방법일 것이다. 초기 커뮤니케이션 단계에서 협조적으로 대응하는 당사자에게 커뮤니티는 한번도 컴플라이언스 소송을 제기한 적이 없었다.

일례로, 소스 코드 관련 의무 이행 요청에 여러 달 동안 대응하지 못한 한 다국적 가전 제품 제조사는 사내 변호사가 소송을 제기한 저작권자를 대표하는 라이선스 원저작자와 개인적으로 한 전화통화에서 단순히 구두상으로 신속한 협조를 약속한 결과로 고소가 제기되기 몇 시간 전에 다수 피고를 대상으로 한 컴플라이언스 강제 집행 소송에서 제외되었다.

2. *고소인 입장에서 준비한다.* 카피레프트 컴플라이언스 불이행에 대한 고소를 제기한 여러 단체는(설립 순서에 따라, FSF, GPL-violations.org, SFLC, 소프트웨어 자유 보호협회(Software Freedom Conservancy)) 컴플라이언스 불이행 당사자에게 연락하기 전에 고소 내용에 대해 충분히 조사하고 검증한다. 원고는 소송의 근거가 되는 사실을 입증하기 위한 준비를 한다.

몇 년 전에 의도치 않게 코드가 포함된 사건에서, 한 글로벌 제조사는 주요 특허 소프트웨어 제품에서 필수적인 기능을 제공하기 위해 카피레프트 라이브러리 전체를 사용했다. 저작권자를 대신해서 이 제조사에 연락을 했을 때, 해당 기업의 FOSS 관련 담당 변호사는 그러한 사건이 발생할 수 있다는 것이나 우리 엔지니어가 해당 기업 제품에서 카피레프트 코드가 포함되어 있는 것을 분명히 확인한 사실을 반복적으로 부인했다. 우리가 해당 제조사의 엔지니어와 세 번이나 확인한 후에야 그러한 실수가 발생할 수 있다는 것을 비로소 인정했다. 사실에 대한 우리의 주장이 검증된 후 소송 및 금전적 손해배상 없이 이 사건은 신속하게 해결되었다.

3. *프로세스에 엔지니어를 참여 시킨다.* 경험에 비추어 봤을 때, 컴플라이언스 관련 문제를 해결하는데 있어 가장 많은 시간이 소요되고 어려운 부분은 소스 코드가 실제로 완전하고 부합하는 소스 코드인지 검증하는 것이다. 양측 소프트웨어 엔지니어간의 직접적인 연락 없이, 배포된 바이너리가 제공된 소스 코드를 통해 구축되었다고 증명하는 기술적인 문제를 해결하는 일은 힘들고, 비용이 들며, 잠재적으로 긴장을 야기하는 일이다. 변호사는 적대적인

당사자로부터의 질문에 고객사의 직원이 직접 답변하는 것을 당연히 꺼린다. 기술적인 문제에 대해 소프트웨어 엔지니어들이 커뮤니케이션을 하면서 주고받은 정보를 통해 문제 해결 시간을 단축할 수 있으며 문제 해결하기까지의 비용도 상당히 줄일 수 있다. 또한 상호 오해로 인한 불필요한 문제를 막아준다.

4. *규정을 준수한다.* 반복해서 위반을 한 경우를 제외하고, GPL 버전 3 계열 라이선스 하에서 컴플라이언스 불이행을 해결하기 위한 신속한 조치를 통해 라이선스 종료를 자동적으로 피할 수 있다. GPL 버전 2의 조건 하에서 라이선스가 자동적으로 종료되지만, 상거래에서 널리 사용되는 프로그램의 GPL 버전 2 라이선서의 대부분은 컴플라이언스 불이행 공지를 받았을 때 신속하게 컴플라이언스 조치를 취하는 당사자에게는 배포권을 즉각적으로 복구시켜주는 정책을 갖고 있다. 세부 사항에 대한 검증과 상호 합의 대상인 컴플라이언스에 대한 제안은 예외 상황을 제외하고 원만한 합의 기반을 구축하기 위한 모든 상황에서 이뤄진다.

소프트웨어 제품으로 배포한 특허 프로그램에 실수로 카피레프트 코드를 포함시킨 경우, 소송을 피하는 가장 주요한 방법은 원만한 합의를 통해 해당 카피레프트 코드를 신속하게 제거하고, 고객과 다른 연락 가능한 수신인에게 컴플라이언스 버전을 재배포하는 것이다.

5. *관계를 향상시키기 위해 컴플라이언스에 대한 논의를 한다.* 개발 커뮤니티는 모든 사용자에게 혜택을 주기 위해 소프트웨어를 만든다. 사용자가 카피레프트 커뮤니티 소프트웨어를 제품에 사용할 경우, 개발자 관점에서 해당 사용자는

커먼즈의 중요하고 귀중한 부분이 된다. 컴플라이언스 문제 해결 과정은 커먼즈와의 관계를 강화할 수 있는 기회이다, 이는 사용자가 자신의 엔지니어와 사용자의 비즈니스에 이용하는 결과물을 만들어낸 프로젝트 간의 커뮤니케이션을 확대함으로써 이를 수 있다.

경험상 봤을 때, 기업은 컴플라이언스 관련 문의를 통해 처음으로 만나게 되는 프로젝트 개발자와 아주 유용한 컨설팅 및 고용 관계를 구축해 나가고 있다. 경우에 따라, 회사의 제품 내 프로젝트 코드 사용 모드를 변경하기 위한 협업이 분쟁 해결의 중요 요소가 되기도 한다. 더 빈번하게, 문의 과정에서 구축된 커뮤니케이션 채널이 추후에 다른 용도나 더욱 생산적인 용도로 사용되기도 한다.

6. *문제가 해결되면, 비용을 지불한다.* 컴플라이언스 관련 소송을 조사하고 해결 방법을 확인하는 작업의 대부분은 수익이 없는 개발 커뮤니티를 대변하는 비영리 단체가 수행한다. 지대 추구 및 타인의 실수를 통한 이익 추구에 반대하는 커뮤니티 당사자의 윤리가 의미하는 바는 컴플라이언스 관련 문의에 대응하고 분쟁을 해결하기 위해서 해당 당사자가 들인 상당한 엔지니어링 노력이 그들의 얼마 되지 않는 수입을 통해 지급된다는 점이다. 컴플라이언스 관련 문제가 성공적으로 해결이 되었을 경우, 커뮤니티 측이 기술자 작업 시간에 대한 비용을 지불하는 것은 미래에도 계속해서 기술자와 좋은 관계를 유지하도록 해준다. 이러한 비용의 상환은 전적으로 타당하며 예측 가능하다. 신속한 조치는 분쟁이 심화되는 것을 방지하며, 이는 SFLC가 활동하며 자주

목격하게 되는 결과이다.⁹

첨부 1 소스 코드 제공의 서면 제안

실용 가이드에서 GPL 컴플라이언스까지¹⁰

다수의 배포자가 완전한 소스 코드 패키지보다 소스 코드 제공 서면 제안을 바이너리 배포와 함께 제공하는 것을 선호한다. 이러한 옵션은 소스 코드 배포 비용이 진정한 제품 단위 당 비용일 경우에만 가치가 있다. 예를 들어, 이 옵션은 CD가 아닌 매뉴얼 또는 기타 인쇄물과 함께 출시되는 내장형 제품의 영구 저장 공간이 소스 코드를 저장하기에 부족할 때 적합하다.

그러나 이 옵션으로 인해 의무 기간이 크게 증가하게 된다. 소스 코드는 마지막 바이너리 배포(GPL 버전 2의 경우), 혹은 마지막 바이너리 또는 예비 부품 배포(GPL 버전 3의 경우)로부터 만 3년 동안 제공되어야 한다. 소스 코드 요청 및 프로비저닝 시스템은 제품 수명주기보다 훨씬 더 오래가도록 설계되어야 한다.

또한, GPL 버전 2의 조항을 준수해야 하는 경우, 네트워크 서비스를 사용하여 소스 코드를 제공할 수 없다. GPL 버전 2의 경우, 소스 코드 제공은 실질적 매체를 통해서만

⁹ SFLC는 FOSS를 만들고 배포하는 비영리 단체에 무료 법률 자문을 제공한다. 컴플라이언스 문제 관련 업무 등을 진행하는 SFLC는 분쟁 해결을 통해 수익을 내기보다 분쟁 해결 서비스의 가치를 인식하는 기업과 개인의 기부를 통해 자금을 조달한다. SFLC는 SFLC가 대변하는 프로젝트를 대신해 컴플라이언스 불이행 당사자에 의한 금전적 분쟁에도 때때로 참여한다. 이러한 분쟁 해결을 통해 얻은 수익은 SFLC의 과거 수익에서 2% 이하를 차지하며, 한 번도 회계연도 수익의 10%를 넘어서지 않았다. SFLC는, 분쟁 합의 시, 고객 전체를 대변하는 데 들이는 노력에 대한 비용이나 분쟁 해결을 위해 경주한 SFLC의 노력에 대한 비용을 회수한 적이 없다.

¹⁰ <http://softwarefreedom.org/resources/2008/compliance-guide.html>.

이뤄질 수 있다. 이는 최신 '소스 코드 CD'를 제품의 수명 종료 시까지 지속해서 제작해야 함을 의미한다.

GPL 버전 2의 경우, 소스 코드 제공 시, 실제 매체를 통한 소스 코드 제공뿐 아니라 다운 가능한 소스 코드용 인터넷 링크를 포함할 것을 허용하고 권장한다. 이러한 방법을 통해, 빠른 네트워크 접속이 가능한 이들에게 소스 코드를 더욱 신속히 제공할 수 있고, 일반적으로 실제 매체를 이용한 소스 코드 제공에 관한 이행 요청의 수를 줄일 수 있다(GPL 버전 3 제6조 b항은 공용 네트워크 접근을 통한 소스 코드 제공만을 허용하고 실제 매체를 통한 소스 코드 제공을 허용하지 않는다. 본 섹션 마지막 부분에서 이를 자세히 다루고 있다).

다음은 각 바이너리 배포 시 인쇄물로 제공할 수 있는 GPL 버전 2에 따른 소스 코드 제공 서면 제안의 예이다(GPL 버전 3에도 적용 가능).

본 제품에 포함된 소프트웨어는 GPL 라이선스 소프트웨어를 포함하고 있습니다. 라이선스 사본은 본 문서의 X쪽에서 확인할 수 있습니다. 우편환(換) 또는 수표를 아래 주소로 보내주시면 2011년 8월 1일 이후 본 제품의 마지막 배송일부터 만 3년간 완전하고 부합하는 소스 코드를 받아 보실 수 있습니다.

미화 5불(\$5 to):

GPL 컴플라이언스 부(GPL Compliance Division)

회사 이름(Our Company)

회사 주소(Any Town, US 99999)

지금 메모에 '제품 Y의 소스'라 작성해 주십시오.

또는 <http://www.example.com/sources/Y/>에서 소스 코드의 복제본을 확인할 수 있습니다.

해당 소스 코드 제공의 서면 제안은 본 정보의 수신인 모두에게 해당됩니다.

소스 코드 제공 서면 제안과 관련하여 몇 가지 중요한 사항이 있다. 첫째, 복제 비용을 제공해야 한다. GPL 버전 2은 '소스 배포 수행 시 물리적으로 드는 비용'만을 허용하고 있다. 해당 비용은 반드시 합당해야 한다. 만약 CD를 복사하고 배송하는 비용이 미화 10불 이상인 경우, 보다 저렴한 CD 및 배송 방법을 찾아야 한다. 커뮤니티에 과잉 청구를 하지 말아야 한다. 위 조항을 남용하여 소스 코드 제공을 통한 영리 사업이 확인 될 경우 소송이 제기될 수 있다.

둘째, 마지막 줄은 소스 코드 제공 서면 제안이 소스 코드를 요청하는 모든 이에게 소스 코드 제공을 보장함을 보여준다. 이는 버전 2 제3조 b항이 부합하는 소스 코드의 복제본을 '제삼자에 제공'하도록 의무화하고 있기 때문이다. GPL 버전 3에도 '오브젝트 코드를 가지고 있는 모두'에게 소스 코드를 제공토록 명시하고 있다. GPL 버전 2 제3조 c항과 버전 3 제6조 c항에 명시된 의무 사항에 따라 비상업적 재배포자는 배포판에 해당 소스 코드 제공에 대한 서면 제안을 전달해야 한다. 소스 코드 서면 제안은 고객뿐만 아니라 그들로부터 바이너리 복제본을 받은 모든 이에게도 유효해야 한다. 상당수의 배포자가 이 의무 사항을 간과하고 직접적인 고객의 요청만 이행하면 된다고

생각한다.

이행 프로세스를 처리 할 장비를 갖춘 회사의 경우, 직접 소스 코드를 배포하는 것보다 소스 코드 제공에 대한 서면 제안을 하는 편이 특별한 혜택이 있다. GPL 버전 2 제3조 c항과 GPL 버전 3 제6조 c항은 소스 코드 제공 의무를 지닌 비영리, 비정기 재배포자가 자신이 받은 소스 코드 제공 서면 제안을 전달하는 것을 허용함으로써 이들의 부담을 덜어주고 있다.

상업적 재배포자는 옵션 (c) 예외를 이용할 수 없으므로, 소스 코드 제공 서면 제안은 GPL 버전 2하에 서면 제안을 받은 사람 또는 GPL 버전 3 하에 오브젝트 코드를 받은 사람 모두에게 적합해야 한다. 자사 제품을 상업적으로 재배포하는 이의 의무는 사라지지 않는다. 라이선스 조건은 원(原)배포자인지 아닌지 여부에 상관없이 GPL 소프트웨어를 배포하는 모든 사람에게 적용된다. 내장형 장치에 사용되는 GPL 소스 코드를 통해 소프트웨어 플랫폼을 개발한 판매 업체 V사를 예로 들어 보자. 제조 업체 M사는 자사의 장치에 V사의 소프트웨어를 펌웨어로 설치하기로 계약한다. V사는 규정을 준수한 소스 코드 제공 서면 제안과 함께 M사에게 소프트웨어를 제공한다. 이 경우, M사는 단순히 V사의 소스 코드 제공 서면 제안을 고객에게 전달할 수 없다. M사 또한 상업적으로 GPL 소프트웨어를 배포하므로, M사도 반드시 GPL을 준수하여 자사의 고객에게 소스 코드를 제공하여야 한다(또는 M사의 자체 소스 코드 제공 서면 제안).

이는 소스 코드 제공에 관한 서면 제안이 고객이 재배포할 가능성이 높은 제품과 관련해서 그리 좋은 선택은 아니라는 점을 보여준다. 만약 배포자가 제품에 소스 코드를 포함시켜 고객에 배포한 경우, 이는 규정을 준수한 것이고, 배포자의 고객이 자사의

고객에 개작하지 않은 원본을 배포하는 것도 규정을 준수하는 것이다. 왜냐하면 두 방법 다 소스 코드를 제공하고 있기 때문이다. 만약 소스 코드 제공에 관한 서면 제안만 포함하여 배포한 경우, 배포자는 규정을 준수하고 있지만 그 고객의 배포에는 해당 컴플라이언스가 ‘상속’되지 않는데, 이는 배포자의 고객이 배포와 관련해 자신의 소스 코드 제공에 관한 서면 제안을 배포에 포함시키지 않았기 때문이다.

GPL 버전 3 하에서 배포할 경우, 소스 코드 제공 관련 서면 제안 조건은 상당히 다르다. 버전 3을 따르는 경우, 제품 또는 관련 부품의 마지막 배포 후 만 3년 간 일반에 공개된 사용 가능한 네트워크 서버를 통해서만 소스 코드의 제공이 가능하다. 따라서, 인터넷만을 통한 배포를 통해 이행 의무를 완수 할 수 있다. 버전 3에 따른 배포의 경우, 이러한 의무 사항은 ‘소스 코드 제공’ 옵션 문제를 경감시켜, 상업적 재배포자의 컴플라이언스 부담을 완화시킨다. 그러나 인터넷 기반 이행 프로세스로 완전히 전환하기 전에 먼저 GPL 버전 3하에서 모든 소프트웨어를 배포할 수 있는 지 확인해야 한다. 일부 프로그램은 실제로 ‘GPL 버전 2 또는 후속 버전’(‘GPL 버전 2 또는 후속판’으로 축약 사용)에 의해 허가가 되므로 이러한 라이선싱은 GPL 버전 3 하에서 재배포할 수 있다. 그러나 몇몇 인기 프로그램은 ‘다른 후속 버전’이 아닌 GPL 버전 2(‘GPL 버전 2만 허용’)에 의해서만 허가가 되므로, 후자의 경우, 인터넷 기반 소스 코드 요청에 대한 이행만을 할 수 없다.

전 제품 내 모든 GPL 저작물을 GPL 버전 3으로 업그레이드하기로 결정했다면(또는 이미 GPL 버전 3으로 시작한 경우), 소스 코드 제공 서면 제안은 아래와 같이 간단하게 작성할 수 있다.

본 제품의 소프트웨어에 GPL 버전 3 하에서 라이선스 된 저작권 소프트웨어가 포함되어 있습니다. 라이선스 사본은 본 문서의 X쪽에서 확인하실 수 있습니다. 2011년 8월 1일 이후 제품 그리고/또는 예비 부품이 마지막 배송이 된 후 3년간 완전하고 부합하는 소스 코드를 당사 웹사이트 <http://www.example.com/sources/productnum/>에서 받아 보실 수 있습니다.

GPL 버전 2와 GPL 버전 3에 따라, 소스 코드 제공 관련 서면 제안은 배포 시 소프트웨어 카피 또는 하드 카피 형태로 라이선스 사본과 함께 제공되어야 한다.

참고 문헌

1. 자유 소프트웨어 재단, GNU 라이선스 관련 자주하는 질문.

<http://www.gnu.org/licenses/gpl-faq.html>, 2014년 10월. (2014년 10월 23일 접속).

2. 자유 소프트웨어 재단, 소프트웨어를 위한 GNU 라이선스 사용법.

<http://www.gnu.org/licenses/gpl-howto.html>, 2014년 4월. (2014년 10월 23일 접속).

3. 자유 소프트웨어 재단, 다양한 라이선스 및 해당 라이선스 관련 해설.

<http://www.gnu.org/licenses/license-list.html>, 2014년 10월. (2014년 10월 23일 접속).

4. 브래들리 쿤(Bradley M. Kuhn), 아론 윌리엄슨(Aaron Williamson), 캐런 샌들러(Karen M. Sandler), 실용 가이드에서 GPL 컴플라이언스까지.

<http://softwarefreedom.org/resources/2008/compliance-guide.html>, 2008년 8월. (2014년 10월 23일 접속).

5. 소프트웨어 자유 법률 센터, 자유 소프트웨어 프로젝트 내 저작권 정보 관리.

<http://softwarefreedom.org/resources/2012/ManagingCopyrightInformation.html>, 2012년 9월. (2014년 10월 23일 접속).

6. GPL 버전 3 개정 과정, 에벤 모글렌(Eben Moglen).

https://www.softwarefreedom.org/resources/2013/A_History_of_the_GPLv3_Revision_Proce

ss.pdf, 2013 7월. (2014년 10월 23일 접속).

7. 자유 소프트웨어 재단, GPL 버전 3 논의 1차 초안. <http://gplv3.fsf.org/comments/gplv3-draft-1>, 2006년 1월. (2014년 10월 23일 접속).

8. 자유 소프트웨어 재단, GPL 버전 3 논의 2차 초안. <http://gplv3.fsf.org/comments/gplv3-draft-2>, July 2006. (2014년 10월 23일 접속).

9. 자유 소프트웨어 재단, GPL 버전 3 논의 3차 초안. <http://gplv3.fsf.org/comments/gplv3-draft-3>, 2007년 3월. (2014년 10월 23일 접속).

10. 자유 소프트웨어 재단, GPL 버전 3 논의 4차 초안.

<http://gplv3.fsf.org/comments/gplv3-draft-4>, May 2007 5월. (2014년 10월 23일 접속).