Does GPLv2 Include an 'Installation Information' Obligation? A Textual & Historical Analysis

P. McCoy Smith [a]

(a) Founding Attorney; Lex Pan Law & Founder, Opsequio

DOI: *10.5033/jolts.v12i1.149*

Abstract
One of the features included in version 3 of the GNU General Public License (GPLv3) was a requirement, in certain circumstances, to provide 'Installation Information.' This was a new addition to the licence to address a 'loophole' that existed in version 2 of the licence (GPLv2); a loophole that was perceived as being exploited, at the time, by certain device vendors. Recently, it has been asserted that this requirement was inherent, or explicitly called for, in GPLv2. This paper examines the historical record around the time that the 'Installation Information' requirement was proposed, and eventually ratified, in GPLv3, to show that this requirement was understood to be both new, and not a part of GPLv2. A textual analysis of GPLv2 yields an identical result.

Keywords
Law; information technology; Free and Open Source Software; GPLv2; GPLv3; 'Installation Information'; Installation 'Scripts'; 'Authorization Keys'

Introduction

The release of GNU General Public License, version 2[1] (GPLv2), first promulgated by the Free Software Foundation in 1991, began the now-widespread adoption of copyleft (or reciprocal) licensing for free and open source software ('FOSS'). By specifying requirements for when, and how, source code was required to be supplied, and further requiring that the same licence be used with further distribution of software licensed under its terms, GPLv2 was considered at the time – and to many is still considered to this day[2] – to be the best vehicle to ensure that software remains 'free': freely shared modifications, free of restrictions on what the user could do with the code, and free to make any changes to the code that a user desired.[3]

[1] GNU Operating System, 'GNU Library General Public License, version 2.0,' (June, 1991) https://www.gnu.org/licenses/old-licenses/lgpl-2.0.html (accessed March 8, 2021).

[2] Although GPLv3 was designed to eventually supplant GPLv2, in the 14 years since GPLv3 was published, the use of GPLv3, by some measures, is roughly equal in measure to the use of GPLv2; GPLv3's relative use is also declining while GPLv2 remains steady state. Johnson, Patricia, 'Open Source Licenses in 2021: Trends and Predictions,' WhiteSource (January 28, 2021) https://resources.whitesourcesoftware.com/blog-whitesource/open-source-licenses-trends-and-predictions (accessed March 30, 2021).

[3] *See* GNU Operating System, 'What is free software? The Free Software Definition,' https://www.gnu.org/philosophy/free-

Nevertheless, by at least 2005, the FSF recognized that certain changes to the GPLv2 licence were desired to address issues in the law[4], and in technology[5], that were not anticipated at the time GPLv2 was released.[6] As a result, beginning in 2006,[7] and extending well into 2007, the FSF launched a large, collaborative, multinational effort to create a new version of the GPL – resulting in the launch of GPLv3 on June 29, 2007.[8]

The 'Installation Information' Requirement of GPLv3

Although numerous features were added in GPLv3 to address issues or concerns that had been raised during the 15 years during which GPLv2 had been in widespread use, the most notable – and possibly the most controversial[9] – addition to GPLv3 were the provisions defining 'Installation Information,' and specifying the circumstances when that information was required to be provided upon 'convey[ing]'[10] software licensed under GPLv3. To understand to what extent the 'Installation Information' requirement in GPLv3 includes elements within, and without, those required in GPLv2, a detailed review of the language, and history, of both licenses is required.

GPLv3, in Section 6 (specifying the obligations when 'Conveying Non-Source Forms' of GPLv3 code), defines a class of information, designated 'Installation Information,' to which certain specific disclosure obligations apply:

> *"'Installation Information' ... means any methods, procedures,* ***authorization keys, or other information*** *required to install and execute modified versions of a covered work ... from a modified version of its Corresponding Source. The information must suffice to ensure that the continued functioning of the modified object code is in no case prevented or interfered with solely because modification has been made."*[11]

Notable in the definition of 'Installation Information' in GPLv3 is the specific recitation of 'authorization keys' and 'other information,' which, as described in more detail below, was included as part of the 'Installation Information' requirement to address a particular use case of GPLv2 software that concerned the FSF at the time the GPLv3 drafting process was launched.[12]

The detailed requirements of the 'Installation Information' obligation of GPLv3, and how, and when,

sw.en.html (accessed March 8, 2021).

[4] One example of a change in the law that the authors of GPLv3 felt needed to be addressed in that license was the adoption in 1996 of the WIPO Copyright Treaty (WCT), and the passage in 1998 of its counterpart in the United States, the Digital Millennium Copyright Action (DMCA), particularly the provisions against circumvention of 'technological protection measures', *See* WCT Article 11; 17 U.S.C. § 1201 (1998). GPLv3, § 3 directly addresses these additions to copyright law.

[5] The technology in TiVo's devices, preventing reinstallation of modified binaries on devices running GPLv2 software, was one example of technology developed long after the GPLv2 licence was drafted that was of concern to the drafters of GPLv3. Subsequent to the release of GPLv3, millions, if not billions, of devices continue to be distributed with a GPLv2-licensed Linux kernel that prevent the reinstallation of modified binaries. GPLv3 also addressed the outmoded language around distribution of source code in GPLv2, and GPLv3 – in Section 6 – added several additional mechanisms for fulfilling source code obligations more consistent with current mechanisms for software distribution. *See* GPLv3, § 6(d)-(e).

[6] Free Software Foundation, 'Rationale for 1st discussion draft,' http://gplv3.fsf.org/gpl-rationale-2006-01-16.html (accessed March 22, 2021).

[7] Irish Free Software Organization, 'Transcript of Opening session of first international GPLv3 conference,' (January 16th 2006) http://www.ifso.ie/documents/gplv3-launch-2006-01-16.html (accessed March 22, 2021).

[8] GNU Operating System, 'GNU General Public License, version 3,' ('GPLv3') (June 29, 2007) https://www.gnu.org/licenses/gpl-3.0.html (accessed March 22, 2021).

[9] Burnette, Ed, 'Tivo and GPL: Beauty and the Beast?,' ZDNet, (October 2, 2006) https://www.zdnet.com/article/tivo-and-gpl-beauty-and-the-beast/ (accessed March 29, 2021).

[10] 'Convey' is the activity defined in GPLv3 as triggering source code disclosure obligations. GPLv3, n. 6, §§ 4-6.

[11] GPLv3, n. 6 above, § 6.

[12] *See* 'Transcript of Opening Session of First International GPLv3 Conference,' (January 16th 2006) http://www.ifso.ie/documents/gplv3-launch-2006-01-16.html (accessed May 5, 2021) at 0h 03m 59s

GPLv3 requires that information to be provided, are beyond the subject matter of this article;[13] a general understanding of that obligation, and the history behind its adoption, is nevertheless necessary to understand where there may be analogues to that obligation in GPLv2 – and where that obligation is unique to GPLv3. As such, it is important to understand the particular historical context that led to the addition of the 'Installation Information' obligation in GPLv3 – as well as the specific language added to GPLv3 – and how that language differs from the obligations recited in GPLv2.

The Historical Background of the 'Installation Information' Requirement: 'Tivoization'

At the time a new version of GPL – GPLv3 – was being considered and proposed, circa 2006, the FSF evidenced a concern about a practice that they felt could potentially detract from their concept of 'software freedom.' This practice was labelled 'Tivoization'[14] by the FSF, as the digital video recorder (DVR) company TiVo had begun engaging in a practice, at around the time that GPLv3 was being contemplated, that the FSF felt was contrary to user freedom.

Certain TiVo DVR hardware devices of the mid 2000s had an installed Linux kernel, licensed under GPLv2. Embedded in that TiVo hardware – either in hardware logic itself, or in firmware installed in a programmable read-only memory (PROM) in the hardware – was a mechanism to validate the version of the Linux kernel to be installed on the TiVo hardware device. This validation mechanism used a checksum, or cryptographic hash function, to compare against the version of the kernel being installed on the device; if the version intended to be installed did not match the checksum or cryptographic hash[15], the mechanism would refuse to install that version of the Linux kernel. In this way, the TiVo device would allow only TiVo (the manufacturer of the hardware and thus the only entity with information necessary to match the embedded checksum or hash) to install their authorized version of the Linux kernel on the device. Users of the TiVo device (for example, customers who purchased the device), if they wished to take the source code for the kernel installed on the device, modify that kernel, and reinstall it, would be unable to reproduce the checksum or hash in that modified kernel, and thus would find that the modified kernel would fail to reinstall or execute.[16]

Therefore in 2006, the inability to reinstall a modified version of GPLv2 software on a device upon which that software was initially installed was, and still is, considered to be a significant detraction

[13] Perhaps the most notable feature of the 'Installation Information' requirement, and an important feature in understanding how that requirement differs from the source code obligations in GPLv2, is that the 'Installation Information' requirement of GPLv3 applies only to a specified subset of products – 'User Products' upon which GPLv3 might be installed. See GPLv3, n. 6 above, at § 6.

[14] The Computer Language Company, 'Tivoization,' The Free Dictionary by Farlex https://encyclopedia2.thefreedictionary.com/Tivoization (accessed April 2, 2021).

[15] Checksums and cryptographic hashes are techniques used to determine whether a received binary file is identical to, or deviates from, an expected binary file. Various techniques are used to generate a numerical value associated with the digits in the expected file to generate a value; that value is then compared at the receiving end to a stored representation of the same value. In this way, any changes to the binary file, even so much as changing one bit from '0' to '1' or vice versa, will produce a different value which will not match the stored value, thus indicating at the received binary file is not identical to the expected binary file. *See* Fisher, T., 'What Is a Checksum?' Lifewire (June 14, 2021) https://www.lifewire.com/what-does-checksum-mean-2625825 (accessed June 14, 2021).

[16] Miller, Todd, 'Using large disks with TiVo,' Sudo Project (2008) https://web.archive.org/web/20120206023943/http://www.gratisoft.us/tivo/bigdisk.html (accessed April 2, 2021) ('it is not possible to replace the kernel on a Series2 TiVo since the PROM requires that the kernel be cryptographically signed with a key from TiVo'). Note that although most of the commentary about the Series 2 TiVo devices of the mid-2000s indicate that they would not allow modified GPLv2 binaries to install or execute, at least one commentator has stated that that device allowed such binaries to be installed and run, but only prevented execution of non-GPLv2 proprietary code on that device. *See* Kuhn, Bradley & Webster, Behan, 'Safely Copylefted Cars: Reexamining GPLv3 Installation Information Requirements,' Linux Foundation Events (2017) at 13 https://events19.linuxfoundation.org/wp-content/uploads/2017/11/Safely-Copylefted-Cars-Reexamining-GPLv3-Installation-Information-Requirements-ALS-Bradley-Kuhn-Behan-Webster-1.pdf (accessed April 9, 2021)

from the FSF's concept of the freedoms that a user should have to software. The FSF has not been shy in describing this practice in highly pejorative terms:

> *"A **tyrant** is a malicious device that refuses to allow users to install a different operating system or a modified operating system. These devices have measures to block execution of anything other than the 'approved' system versions."*[17]

Historical Analysis: GPLv3's 'Installation Information' Obligation and Its Relation to GPLv2

Although the FSF has long had an objection to the practice of 'Tivoization' – preventing the reinstallation of modified binary code licensed under one of the FSF's family of licences – it, through the statements of its President, its lead attorney during the drafting of GPLv3, and its Executive Director, also made abundantly clear that this practice was permissible under GPLv2:

> *"[T]he Tivo itself is the prototype of [T]ivoisation. The Tivo contains a small GNU/Linux operating system, thus, several programs under the GNU GPL[v2]. And, as far as I know, **the Tivo company does obey GPL version 2**. … [T]he trouble begins because the Tivo will not run modified versions, the Tivo contains hardware designed to detect that the software has been changed and shuts down."*[18]

> *"TiVo is a provider of hardware and software …. Our concern with them is that they have rights as users, but they should respect the rights of the users to whom they sell. Having a personal video recorder … which won't run software if you modify the box … is not user-respecting conduct. **(TiVo) complied with GPL 2 by the skin of its teeth.**"*[19]

> *"TiVoization is described by Peter Brown [Executive Director of FSF in 2006-07 during drafting of GPLv3] as circumventing GPL2 'in spirit, **not technically**."*[20]

The distinction between what was being proposed in GPLv3 to address 'Tivoization' (and other similar techniques for validation of GPL binaries), and what was allowed under GPLv2, was a crucial distinction for Linus Torvalds, author of the Linux kernel, in preferring to retain "GPLv2 only" for the kernel, rather than moving to GPLv3:

> *"'The FSF is trying to make some things **no longer permissible under the GPLv3 that the GPLv2 left open**, and I just happen to think that those things were better off being left open."*[21]

> *"'I don't think the GPL v3 conversion is going to happen for the kernel, since I personally don't want to convert any of my code.' … **'I think it's insane to require people to make their private signing keys available**, for example. I wouldn't do it,' [Torvalds] said."*[22]

[17] GNU Operating System, 'Proprietary Tyrants,' https://www.gnu.org/proprietary/proprietary-tyrants.html (accessed April 2, 2021).

[18] Stallman, Richard, 'Transcript of Richard Stallman at the 5th international GPLv3 conference,' (November 21, 2006) https://fsfe.org/activities/gplv3/tokyo-rms-transcript#tivoisation (accessed April 2, 2021).

[19] Shankland, Stephen, 'Defender of the GPL,' CNet (January 19, 2006) https://www.cnet.com/news/defender-of-the-gpl/ (accessed April 2, 2021).

[20] Byfield, Bruce, 'GPLv2 or GPLv3?: Inside the Debate,' Datamation (June 17, 2007) https://www.datamation.com/trends/gplv2-or-gplv3-inside-the-debate/ (accessed April 9, 2021).

[21] Bennett, Amy, 'Linux creator Torvalds still no fan of GPLv3,' Computerworld (July 28, 2006) https://www.computerworld.com/article/2820022/linux-creator-torvalds-still-no-fan-of-gplv3.html (accessed April 7, 2021).

[22] Shankland, Stephen, 'Torvalds rules out GPL3 for Linux,' ZDNet UK (January 27, 2006)

> *“[If] you can not **install** or **run** your changes on somebody else’s hardware … it in no way changes the fact that you got all the source code, and you can make changes (and use their changes) to it. **That requirement has always been there, even with plain GPLv2.** You have the source. The difference? The hardware may only run signed kernels. The fact that the hardware is closed is a **hardware** license issue. Not a software license issue. I’d suggest you take it up with your hardware vendor, and quite possibly just decide to not buy the hardware. Vote with your feet. … **[I]t’s important to realize that signed kernels that you can’t run in modified form under certain circumstances is not at all a bad idea in many cases.**’*[23]

Torvalds’ opinion on the ‘Installation Information’ requirement in GPLv3 was also shared by several important kernel developers,[24] remained a consistent position a decade later, and to this day is one of the reasons the Linux kernel remains licensed under ‘GPLv2 only.’[25]

> *“I give you source code, you give me your changes back; we’re even. … That’s my take on GPL version 2 and it’s that simple. … **Version 3 extended that in ways that I personally am really uncomfortable with.** Namely I give you source code, that means if you use that source code, you can’t use it on your device unless you follow my rules. And to me that’s a violation of everything version 2 stood for. And I understand why the FSF did it, because I know what the FSF wants, **but to me it’s not the same license at all.** So I was very upset, and made it very clear, and this was months before version 3 was actually published.’*[26]

Throughout the process of proposing, revising, and ultimately publishing GPLv3, the FSF also made clear that it was adding features to ensure that GPLv3, unlike GPLv2, would prevent ‘Tivoization’:

> *“There are several primary areas **where version 3 is different from version 2. One is in regard to [T]ivoisation.**’*[27]

> *“The Tivo includes some GPL-covered software. …[Y]ou can get the source code for that, as required by the GPL … and once you get the source code, you can modify it, and there are ways to install the modified software in your Tivo and if you do that, it won't run, period. Because, it does a check sum of the software and it verifies that it's a version from them and if it's your version, it won't run at all. **So this is what we are forbidding, with the text we have written for GPL version three.** It says that the **source code they must give you includes whatever signature keys, or codes that are necessary to make your modified version run.**’*[28]

The FSF – from the time GPLv3 was first proposed, and continuing to the date of publication of this article – has made clear that indeed GPLv3 has a more expansive definition of the obligations encompassed in the ‘Installation Information’ requirement than any requirement that GPLv2 contains:

https://web.archive.org/web/20080424051024/http:/news.zdnet.co.uk/software/0,1000000121,39249370,00.htm (accessed April 7, 2021).

[23] Barr, Joe, ‘Torvalds versus GPLv3 DRM restrictions,’ Linux.com (February 2, 2006) https://www.linux.com/news/torvalds-versus-gplv3-drm-restrictions/ (accessed April 8, 2021).

[24] Bottomley, James, et al., ‘Kernel developers' position on GPLv3,’ LWN.net (September 22, 2006) https://lwn.net/Articles/200422/ (accessed April 8, 2021). *See also* Bottomley, James, et al., 'The Dangers and Problems with GPLv3,' (September 15, 2006) https://lore.kernel.org/lkml/1158941750.3445.31.camel@mulgrave.il.steeleye.com (accessed May 27, 2021).

[25] Linux kernel licensing notice, https://elixir.bootlin.com/linux/latest/source/COPYING (accessed April 8, 2021).

[26] Deb Conf, ‘Linus Torvalds says GPL v3 violates everything that GPLv2 stood for,’ YOUTUBE (accessed May 5, 2021, at 0h 0m 34s) https://www.youtube.com/watch?v=PaKIZ7gJlRU.

[27] Stallman, Richard, ‘Transcript of Richard Stallman at the 3rd international GPLv3 conference,’ (June 22, 2006) https://fsfe.org/activities/gplv3/barcelona-rms-transcript.en.html#tivoisation (accessed April 2, 2021).

[28] Stallman, Richard, ‘Transcript of Richard Stallman speaking on GPLv3 in Torino,’ (March 18, 2006) https://fsfe.org/activities/gplv3/torino-rms-transcript.en.html#drm (accessed April 2, 2021).

> *"**GPLv2 did not address the use of technical measures to take back the rights that ... GPL[v2] granted**, because such measures did not exist in 1991 [when GPLv2 was written], and would have been irrelevant to the forms in which software was then delivered to users. … **GPLv3 must address these issues: free software is ever more widely embedded in devices that impose technical limitations on the user's freedom to change it**."*[29]

> *"Does GPLv2 have a requirement about delivering installation information?...*
>
> *"GPLv3 explicitly requires redistribution to include the full necessary 'Installation Information.' GPLv2 doesn't use that term, but it does require redistribution to include scripts used to control compilation and installation of the executable with the complete and corresponding source code. **This covers part, but not all, of what GPLv3 calls 'Installation Information.' Thus, GPLv3's requirement about installation information is stronger.**"*[30]

In his plea to software developers to "upgrade" to GPLv3 to address existing problems in GPLv2, Richard Stallman cited the new Installation Information requirement as the first reason developers should move to GPLv3:

> *"Keeping a program under GPLv2 won't create problems. **The reason to migrate is because of the existing problems which GPLv3 will address**.*
>
> *One major danger that GPLv3 will block is tivoization. Tivoization means computers (called "appliances") contain GPL-covered software that you can't change, because the appliance shuts down if it detects modified software. The usual motive for tivoization is that the software has features the manufacturer thinks lots of people won't like. The manufacturers of these computers take advantage of the freedom that free software provides, but they don't let you do likewise.*[31]

The Source Code Obligation of GPLv2

One of the most notable features of copyleft licensing, as exemplified by the release of GPLv2 in 1991, is the obligation imposed upon any person or entity which 'distribute[s]'[32] code licensed under GPLv2's terms, to provide 'source code.'[33] GPLv2, Section 3, specifically defines what constitutes the 'source code' which must be provided in the event that GPLv2 licensed code is distributed in object or executable code form:

> *"The source code for a work means **the preferred form of the work for making modifications to it**. For an executable work, complete source code means all the source code for all modules it contains, plus **any associated interface definition files**, plus **the scripts used to control compilation and installation of the executable**."*[34]

[29] Free Software Foundation, 'Opinion on Digital Restrictions Management,' (August, 2006) http://gplv3.fsf.org/drm-dd2.html (accessed March 17, 2021).

[30] GNU Project, 'Frequently Asked Questions About the GNU Licenses,' https://www.gnu.org/licenses/gpl-faq.html#InstInfo (accessed April 7, 2021)

[31] Stallman, Richard M. 'Why Upgrade to GPL Version 3,' (May 31, 2007) http://gplv3.fsf.org/rms-why.html (accessed May 6, 2021).

[32] GPLv3 uses the term 'convey,' n. 8 above, whereas GPLv2 uses the term 'distribute,' to articulate acts that trigger, among other things, obligations to provide source. Although there are subtle differences between the two terms, they are intended to cover the same acts. GNU Project, 'Frequently Asked Questions About the GNU Licenses,' https://www.gnu.org/licenses/gpl-faq.html#ConveyVsDistribute (accessed March 29, 2021).

[33] Brown, Neil, 'GNU GPL 2.0 and 3.0: obligations to include licence text, and provide source code,' JOLTS vol. 2, no. 1 (2010) DOI: 10.5033/ifosslr.v2i1.31 (accessed March 30, 2021).

[34] GPLv2, n. 1 above, § 3.

Although the description of the obligation to provide source code is primarily tied to the general understanding of what is ‘source code’ in computer programming, i.e.:

> *“Source Code: … The form in which a computer program (software) is written by the programmer. Source code is written in some formal programming language which can be compiled automatically into object code or machine code or executed by an interpreter.”*[35]

GPLv2 also includes two other items falling within that licence’s definition of ‘source code’: ‘associated interface definition files’ and ‘scripts used to control compilation and installation of the executable.’ An examination of the meaning of each of these provisions is necessary to understand how the disclosure obligations in GPLv2 differ from those in GPLv3.

Textual Analysis: GPLv3’s ‘Installation Information’ Obligation and the Source Code Obligations of GPLv2

As discussed above, GPLv3’s disclosure obligations upon distribution of executable code include both ‘Corresponding Source’:

> *“[A]ll the source code needed to generate, install, and (for an executable work) run the object code and to modify the work, including scripts to control those activities.”*[36]

and ‘Installation Information’:

> *“[A]ny methods, procedures,* ***authorization keys****, or other information required to install and execute modified versions of a covered work ... from a modified version of its Corresponding Source.”*[37]

As originally drafted, GPLv3 included the obligation to provide, e.g., authorization keys within the definition of “Corresponding Source”;[38] the FSF – in the face of objections to the conflation of data, like authorization keys, with source code, subsequently moved the authorization key requirement to a different section in response to those objections:

> *“We have moved the technical restrictions provisions from section 1, where they formed part of the definition of Corresponding Source, to section 6,* ***where they are presented as a condition on the right to convey object code works****. Some critics of the provisions in our earlier drafts focused on what they regarded as* ***an inappropriate equation of cryptographic keys with source code.*** *Placing the requirements in section 6 should make their purpose and reasonableness more evident.”*[39]

Thus, during the drafting stages of GPLv3, the Free Software Foundation recognized, and acknowledged in changes to the drafting of that license, that the ‘Installation Information’ requirements were a separate obligation beyond the Corresponding Source Code obligations extant in GPLv2 and ported into GPLv3.

[35] ‘Source Code,’ Computer Dictionary of Information Technology https://www.computer-dictionary-online.org/definitions-s/source-code.html (accessed March 30, 2021).

[36] GPLv3, n. 6 above, § 1.

[37] GPLv3, n. 6 above, § 6.

[38] Free Software Foundation, ‘GPLv3 First Discussion Draft,’ §1 (January 16, 2006) http://gplv3.fsf.org/gpl-draft-2006-01-16.html (accessed June 14, 2021).

[39] Free Software Foundation, ‘GPLv3 Third Discussion Draft Rationale,’ (March 28, 2007) http://gplv3.fsf.org/gpl3-dd3-rationale.pdf/download (accessed June 14, 2021).

GPLv2’s source code disclosure obligations recite only:

> *“For an executable work, complete source code means all the source code for all modules it contains, **plus any associated interface definition files**, **plus the scripts used to control compilation and installation of the executable**.”*[40]

If there is any correspondence between the ‘Installation Information’ requirement of GPLv3 within the ‘corresponding source code’ requirement of GPLv2, it is in the recitation of two separately articulated disclosure obligations – ‘any associated interface definition files’ and ‘scripts used to control compilation and installation of the executables.’

Although ‘interface definition file’ is an uncommonly used term in computer programming (and there is no more detailed definition of this term in GPLv2), it would likely be interpreted as a separate file, possibly including attributes and definitions for any programming interfaces for a particular piece of software.[41] This requirement of GPLv2 would seem in no way to impose obligations to provide keys or checksums or any other information required to allow installation or execution of modified binaries; instead it requires disclosure of information necessary to understand interfaces into the distributed binaries – that otherwise would not be revealed by disclosure of the source code to those binaries itself.

‘[S]cripts used to control compilation and installation of the executable,’ on the other hand, is, at least, directed to material related to the installation of an executable subject to GPLv2. Nevertheless, this requirement is directed to ‘scripts,’ a term with generally-understood meaning in computing:

> *“A computer script is a list of commands that are executed by a certain program or scripting engine. Scripts may be used to automate processes on a local computer …. Script files are usually just text documents that contain instructions written in a certain scripting language. … [W]hen opened by the appropriate scripting engine, the commands within the script are executed.”*[42]

> *“Script[:] … a sequence of instructions or commands for a computer to execute … especially … one that automates a small task (such as assembling or sorting a set of data).”*[43]

Installation scripts[44] are generally small, simple programs used to automate the process of installing a particular program on a particular device.[45]

It would appear unquestionable that GPLv2’s obligation to provide ‘scripts used to control … installation of the executable’ cannot be construed, as a matter of textual interpretation, to cover checksums, hashes, authorization, or signing keys, or other numerical data embedded in hardware or firmware that is used to validate the installation of GPLv2 executable code; that data would not fall within the common understanding of what is a ‘script.’ A more interesting interpretational issue

[40] GPLv2, n. 1 above, § 3.

[41] *E.g.*, Microsoft, ‘Interface Definition (IDL) File,’ Windows Developer Documentation (May 31, 2018) https://docs.microsoft.com/en-us/windows/win32/midl/interface-definition-idl-file (accessed April 8, 2021); de St. Germain, H. James, ‘Interfaces in Object Oriented Programming Languages,’ University of Utah Computing Department https://www.cs.utah.edu/~germain/PPS/Topics/interfaces.html (accessed April 8, 2021).

[42] Christensson, Per, ‘Script Definition,’" TechTerms. (2006) https://techterms.com/definition/script (accessed April 8, 2021).

[43] ‘Script,’ Merriam-Webster.com Dictionary, Merriam-Webster https://www.merriam-webster.com/dictionary/script (accessed April 8, 2021).

[44] GPLv2’s requirement to provide ‘compilation’ scripts are not analysed in this article; compilation is part the process of converting source code into executable code, and is not related to the subsequent activities of installing, or executing, that executable code.

[45] Arthur, Ty, ‘How to Write a Simple Script to Install a Program,’ Techwalla https://www.techwalla.com/articles/how-to-write-a-simple-script-to-install-a-program (accessed April 8, 2021)

would be if a hardware device included firmware running an installation program that – as part of its functionality – engaged in some form of validation of the executable and barred installation if the executable was determined to be invalid – for example, as the result of modification. Given the long-standing and consistent position of both the FSF and the Linux kernel developers during the drafting and release of GPLv3 that any sort of installation validation – including TiVo's validation using PROM-loaded information – was permissible under GPLv2, even a firmware-instantiated validation feature would be difficult to argue falls within the 'scripts used to … installation of the executable' requirement of GPLv2.

Backporting the Installation Information Requirement Into GPLv2 Would, Counter-Intuitively, Render GPLv3 Less 'Freedom'-Respecting Than GPLv2

By way of an illustrative example of why efforts to backport the full 'Installation Information' definition in GPLv3 into the source code obligations of GPLv2 yields ahistorical and textually incorrect results: imagine if the full and complete 'Installation Information' definition was imported into Section 3 of GPLv2. In doing so, a specific dilemma is quickly created. GPLv3's 'Installation Information' requirement is not a fully-intact, portable, requirement, but is instead specifically tied to a defined class of products to which the obligation applies: so-called 'User Products.'[46] In GPLv3, the requirement to provide 'Installation Information' applies only to defined 'User Products,' and not any other product:

> *"If you convey an object code work under this section **in, or with, or specifically for use in, a User Product** ... the Corresponding Source conveyed under this section **must be accompanied by the Installation Information.**"*[47]

In contrast, GPLv2 has no definitions, limitations or restrictions on the types of products to which its source code obligations apply – 'User Products' and non-'User Products' alike must provide the source code and other disclosure obligations of GPLv2. Thus, if GPLv3's full 'Installation Information' obligation definition were merely a reiteration or clarification of the existing disclosure obligations of GPLv2, GPLv3 would have narrowed the circumstances under which those disclosure obligations are extant. As a result, GPLv3 would be less 'software freedom' respecting than GPLv2 because its obligation would apply to a smaller subset of software. That result would be directly contrary to the stated goals of creating GPLv3 in the first place:

> *"As a free software license ... this license [GPLv3] intrinsically disfavours technical attempts to restrict users freedom to copy, modify, and share copyrighted works. Each of [the licenses] provisions shall be interpreted in light of this specific declaration of the licensor's intent. We wish courts all over the world to understand that our intent [in creating GPLv3] **is to maximise freedom**, not to restrict it, and that everything should be so understood when effect is given to its terms"*[48]

GPLv3 only expands and maximizes freedom if the 'Installation Information' obligation itself expands 'freedom' beyond the disclosure obligations of GPLv2. As GPLv2 obligations are not limited

[46] 'User Products' in GPLv3 are subject to a rigorous definition which excludes a large class of products which can, and currently do, use code licensed under one of the GPL family of licences: "A 'User Product' is either (1) a 'consumer product', which means any tangible personal property which is normally used for personal, family, or household purposes, or (2) anything designed or sold for incorporation into a dwelling. … A product is a consumer product regardless of whether the product has substantial commercial, industrial or non-consumer uses, unless such uses represent the only significant mode of use of the product." GPLv3, n. 6 above, at Section 6.

[47] GPLv3, n. 6 above, at Section 6.

[48] Transcript of Opening Session of First International GPLv3 Conference, see n.10 above, at 0h 23m 30s.

to any product subset, providing obligations in GPLv3 only for User Products would thus necessarily represent a reduction of 'freedom.'

Textual & Historical Revisionism of GPLv2

As outlined in detail above, both a textual analysis, and a review of the historical record, makes clear that the 'Installation Information' obligations of GPLv3 do not exist in, and can in no way be backported into, the source code obligations found in GPLv2. Despite these facts, there has been a recent effort to alter the historical record, and to reinterpret the requirements of GPLv2, to equate the source code obligations in GPLv2 with the source code and 'Installation Information' requirements of GPLv3:

> *"GPLv2 §3 requires that the source code include 'meta-material' like scripts, interface definitions,* ***and other material that is used to 'control compilation and installation'*** *of the binaries."*[49]

> *"GPLv2 included a clear obligation to provide 'the scripts used to control … installation' that function for the GPLv2'd works. GPLv2 assures, to the purchaser of an embedded product, their absolute right to receive* ***the information necessary to install a modified version of the GPLv2'd works****. … The GPLv2 was designed to assure bug-fixing. Furthermore, the drafters knew that, on embedded systems and devices, you need to know* ***how*** *to install those fixes. Scripts can be technical [artefacts] like shell scripts,* ***but can also be merely a recipe and/or guidance — written instructions that explain how to succeed at install****."*[50]

As evidenced by the statements above, efforts are now being made to import concepts from GPLv3's 'Installation Information' requirement – that 'information,' 'recipe[s],' 'guidance,' or 'instructions' for executables to be installed and run be provided – into the more constrained obligation in GPLv2 to provide installation 'scripts.' These efforts are both counter-textual to the actual requirements of GPLv2, and also ahistorical – given that the very 'drafters' of GPLv2 have conceded that the sort of information required to allow installation of modified executables on the TiVo device were **not** required to be provided under GPLv2.[51]

Conclusion

The text, and the historical record, of GPLv3 makes clear that the 'Installation Information' requirement in that licence was specifically designed to add new requirements that were not found in GPLv2. The historical record also makes clear that it was perfectly permissible to distribute code licensed under GPLv2 without providing information – such as authorization keys or other hardware-embedded information that might prevent the installation of modified versions of that GPLv2 code – and that only a fairly narrow class of information – installation scripts – were required in GPLv2. Efforts to backport the 'Installation Information' requirements of GPLv3 into GPLv2 are both ahistorical, and yield the counter-intuitive result that make GPLv3 less 'freedom'-preserving than GPLv2 – a result which was in no way the goal of creating GPLv3 in the first instance. This counter-intuitive result would have counselled software freedom-loving authors to prefer GPLv2 over GPLv3,

[49] Kuhn, Bradley, et al., 'Copyleft and the GNU General Public License: A Comprehensive Tutorial and Guide,' Copyleft.org at § 5.2 (2003-2018) https://copyleft.org/guide/comprehensive-gpl-guidech6.html#x9-460005.2 (accessed April 9, 2021).

[50] Gingerich, Denver, 'Understanding Installation Requirements in GPLv2,' Software Freedom Conservancy (March 25, 2021) https://sfconservancy.org/blog/2021/mar/25/install-gplv2/ (accessed April 9, 2021).

[51] *See* above nn. 17 and 22-23.

a result which would be contrary to the entire purpose of creating and launching GPLv3.

The extent to which an ahistorical and textually unsupported interpretation of GPLv2 is a mere theoretical debate, or is one which may eventually be tested in an interpretational tribunal as the result of compliance enforcement litigation, remains to be seen. Many of the statements made during the GPLv3 drafting process, and the actual language from GPLv2 – as outlined in detail above – will likely be relevant to any decision on scope of the source code obligations of GPLv2.

About the author

P. McCoy Smith *is Founding Attorney at Lex Pan Law (www.lexpan.law), a full-service intellectual property law firm in Portland, Oregon, U.S.A., that has a sub-speciality in free and open source licensing, as well as Founder at Opsequio (www.opsequ.io), an software licence compliance consultancy. As a member of GPLv3 Discussion Committee B, he was an active participant in the debate over, and revision of, the 'Installation Information' requirement in that licence.*

Licence and Attribution

This paper was published in the Journal of Open Law, Technology, & Society, Volume 12, Issue 1 (April 2021). It originally appeared online at https://www.jolts.world

This article should be cited as follows:

Smith, P. McCoy (2021) 'Does GPLv2 Include an "Installation Information" Obligation? A Textual & Historical Analysis', Journal of Open Law, Technology & Society, 12(1), pp 21 – 31

DOI: 10.5033/jolts.v12i1.149