



Open Source License Expectations

1	<b>Contents</b>	
2	<b>Apertis goals</b>	<b>2</b>
3	<b>Licensing constraints</b>	<b>2</b>
4	GPL-3 and derivatives . . . . .	2
5	Original 4 clause BSD license . . . . .	3
6	<b>Ensuring continuous maintenance of open source licence documen-</b>	
7	<b>tation</b>	<b>3</b>
8	<b>Apertis Licensing expectations</b>	<b>4</b>
9	General rules of the Apertis project and their specific constraints . . .	4
10	Apertis Repository component specific rules . . . . .	4
11	target . . . . .	4
12	hmi . . . . .	5
13	sdk . . . . .	5
14	development . . . . .	5
15	<b>Auditing the license of a project</b>	<b>6</b>
16	<b>Documenting exceptions</b>	<b>6</b>
17	<b>Future improvements</b>	<b>7</b>
18	<b>Appendix</b>	<b>7</b>
19	The Debian Free Software Guidelines (DFSG) . . . . .	7
20	The license is an important element in open source projects as the license define	
21	acceptable use cases, user rights, and contribution guidelines. There are different	
22	ways to identify the license from the project source code such as SPDX headers,	
23	the LICENSE file, and the COPYING file. However an open source project may	
24	contain files from other projects and may use different licenses for different files.	

## 25 **Apertis goals**

26 Apertis aims to accomplish the following goals:

- 27 • Ensure that all the software shipped in Apertis is open source or at least
- 28 freely distributable, so that downstreams are entitled to use, modify and
- 29 redistribute work derived from our deliverables.
- 30 • Ensure that Apertis images targeting devices (such as target and mini-
- 31 mal), are not subject to licensing constraints that may conflict with the
- 32 regulatory requirements of some intended use cases.

33 In order to reach these goals, the below assumptions are made:

- 34 • **Licenses declared by open source projects are correct:** The soft-  
35 ware authors correctly document the licensing of their released software  
36 sources and that they have all the rights to distribute it under the docu-  
37 mented terms.
- 38 • **Licenses verified by the Debian project are correct:** The package  
39 distributors (that is, Debian maintainers and the FTP Masters team)  
40 check that the licensing terms provided by the software authors are  
41 open source using the definitions in the [Debian Free Software Guide-](#)  
42 [lines](#)<sup>1</sup> and ensure those terms are documented in a canonical location  
43 (debian/copyright in the package sources).

## 44 Licensing constraints

45 Apertis currently limits the usage of the licenses below:

- 46 • GPL-3.0 and derivatives (LGPL-3, AGPL-3)
- 47 • BSD-4-Clause

### 48 GPL-3 and derivatives

49 [Version 3 of the GPL license](#)<sup>2</sup> was created to address the concern of users who  
50 were prevented from running modified code on their device, when the device was  
51 shipped with open source software. A common method for preventing users to  
52 run their own code is by using signature verification. This practice is known as  
53 [Tivoization](#)<sup>3</sup>. Those licensing rules are a constraint because in some application  
54 domains, it is a regulatory (or safety) requirement to ensure that the hardware  
55 runs verified software.

### 56 Original 4 clause BSD license

57 The [BSD-4-Clause](#)<sup>4</sup> license still contains the problematic advertisement clause  
58 that was dropped in later versions and is thus to be avoided in Apertis.

59 The original authors of the license retroactively deleted the problematic clause  
60 on the software under the University of California copyright, leading to the [BSD-](#)  
61 [4-Clause-UC](#)<sup>5</sup> variant which resolves the issue on the original software, but not  
62 on software with different copyright holders.

---

<sup>1</sup>[https://www.debian.org/social\\_contract#guidelines](https://www.debian.org/social_contract#guidelines)

<sup>2</sup><https://spdx.org/licenses/GPL-3.0-or-later.html>

<sup>3</sup><https://en.wikipedia.org/wiki/Tivoization>

<sup>4</sup><https://spdx.org/licenses/BSD-4-Clause.html>

<sup>5</sup><https://spdx.org/licenses/BSD-4-Clause-UC.html>

## 63 **Ensuring continuous maintenance of open source** 64 **licence documentation**

65 Maintaining the open source licenses documentation is an incremental process:

66 At the time of rebase, licenses are checked manually for all packages involved in  
67 the rebase. This covers the whole archive.

68 During the development, updates are monitored. The integration of a new  
69 project in Apertis and the update of source code are the operations that can  
70 result in the update of a license. New projects can be integrated at any time in  
71 Apertis. If new sources for a project already in Apertis are received: the license  
72 of the project can change, or the licensing for some distributables within this  
73 project can differ from the prevalent license.

74 From a project perspective, Apertis teams tries to do a full scan on all projects  
75 at each release cycle.

76 Open source software shipped with devices that users buy adds significant li-  
77 censing constraints to the software stack of preview and product releases. These  
78 constraints do not affect development releases, and it is possible to save some  
79 work on those releases.

80 In an ideal situation, regular checks of the whole archive would be automated to  
81 ensure nothing escaped the manual checks. While the [Apertis maintainers](#)<sup>6</sup> are  
82 already manually checking packages, the automated whole-archive checks are  
83 not currently implemented. [Future improvements](#) presents a possible solution.

## 84 **Apertis Licensing expectations**

### 85 **General rules of the Apertis project and their specific con-** 86 **straints**

87 The [Debian Free Software Guidelines](#)<sup>7</sup> defines expectations for the licenses of  
88 the projects that are integrated in Debian. They serve as a base for Apertis  
89 policy. The DFSG can be read in the [Appendix](#) section of this document.

90 On top of the DFSG expectations, Apertis defines additional rules for specific  
91 sections of its package repository which are described in [Apertis specific rules](#).  
92 In particular, the sections in the Apertis package repository are meant to group  
93 the packages that are installed on images for target devices and should thus be  
94 free of [licensing constraints](#).

95 Debian packages in a repository are organized in components. A component is  
96 a group of packages sharing a common policy. A single image can incorporate

---

<sup>6</sup><https://jwd.pages.apertis.org/apertis-website/policies/contributions/#the-role-of-maintainers>

<sup>7</sup>[https://www.debian.org/social\\_contract#guidelines](https://www.debian.org/social_contract#guidelines)

97 packages from different components.

## 98 **Apertis Repository component specific rules**

99 The canonical source of Licensing information is this document. Each repository  
100 is listed here, with the rules that apply.

101 Each component contains several source packages, and each source package can  
102 generate multiple binary packages. For example, in a client server project, it's  
103 possible for a source package to generate two binary packages: one for the server  
104 side of a project, and one for the client side. Each binary package can have a  
105 different license.

106 For current apertis releases, the following components exist:

- 107 • target: contains packages for the final devices,
- 108 • hmi: contains user interfaces packages,
- 109 • sdk: contains packages specific to SDK
- 110 • development: contains packages useful for developers

111 The license expectations for each of those components are defined below. Any  
112 package outside these expectations should be documented as [license exceptions](#)<sup>8</sup>.

### 113 **target**

114 This component ships source packages producing binary packages used in images  
115 deployable on target devices. For a file in a binary package to be considered an  
116 artifact, the file must have been generated/compiled/translated from a source  
117 package. An artifact can be an executable, a library, or any other file that is  
118 subject to a license. Specifically, the binary packages installed on those images  
119 should not be affected by licensing constraints. This does not mean that every  
120 source or binary package in the component must be completely unrestricted:

- 121 • source packages may contain restricted build scripts, provided that the  
122 license does not affect generated artifacts
- 123 • source packages may contain restricted tests or utilities, provided that  
124 they are not shipped in the same package as the unrestricted artifacts  
125 installed on target images
- 126 • binary packages may contain restricted artifacts, provided that they are  
127 built from a source package also producing unrestricted packages that are  
128 shipped on target images
- 129 • binary packages may contain restricted artifacts with added exceptions.  
130 The [GCC Runtime Library Exception](#)<sup>9</sup> covering `libgcc` is the main exam-  
131 ple. Those exceptions should be documented as [license exceptions](#)<sup>10</sup>.

---

<sup>8</sup><https://jwd.pages.apertis.org/apertis-website/policies/license-exceptions/>

<sup>9</sup><https://www.gnu.org/licenses/gcc-exception-3.1-faq.html>

<sup>10</sup><https://jwd.pages.apertis.org/apertis-website/policies/license-exceptions/>

132 **hmi**

133 This component has the same usage and constraints as the `target` component.

134 **sdk**

135 This component ships source packages producing binary packages suitable for  
136 images deployable on SDK images. Since the packages hosted in this component  
137 are only meant for development purposes, no further requirement is imposed  
138 other than the DFSG ones.

139 **development**

140 This component provides the packages needed to build the packages in the `target`  
141 repository component but that are not meant to be installed on target devices.  
142 Build tools like GNU binutils, the GNU Autotools, or Meson are hosted in this  
143 component.

144 Dependencies of packages in the `target` component that are not meant to be  
145 installed on target images are also hosted in this component. For instance, many  
146 source package in the `target` component also build a binary package containing  
147 their tests which are not intended to be part of the target images: the extra  
148 dependencies required by the test package but not by the main package are  
149 hosted in the development component.

150 The development component also host development tools that are not part of  
151 the target images by default, but that may be useful to install manually on  
152 target devices during development. Tools like `strace`, `tcpdump` or `bash` belong to  
153 this category.

154 Since those packages are exclusively intended for a development purpose within  
155 the Apertis development team no further requirement is imposed other than the  
156 DFSG ones.

## 157 **Auditing the license of a project**

158 Auditing the license of an imported package depends of the type of the project.

159 For debian packages, the Debian licensing informations gives a good indication  
160 if a project can be integrated in Apertis. Debian maintainers take extreme  
161 precaution to ensure that what they redistribute is redistributable. Using the  
162 Debian licensing informations provides many benefits:

- 163 • vetting licensing terms to ensure they are open source (in particular, as  
164 defined in the DFSG)
- 165 • ensuring that non DFSG-compliant items are excluded from the source  
166 code
- 167 • a standardized location for the licensing information (that is, `de-`  
168 `bian/copyright` in the package source)

169     • an ongoing effort to make the provided licensing information machine-  
170         readable (DEP-5)

171 Some projects may not be packaged by Debian. In this case, the project source  
172 code should contain a document stating the license. Any project that do not  
173 provide license information should not be redistributed.

## 174 Documenting exceptions

175 For Apertis, the list of exceptions should mention:

- 176     • The project location in Apertis mainly gitlab or OBS.
- 177     • The project source package name
- 178     • The project component
- 179     • The rule the project does not meet that requires the exception
- 180     • The reason behind the exception
- 181     • The date at which the exception was made
- 182     • The name of the person who validated the exception

183 The canonical source of Licensing exceptions is the [license exceptions](#)<sup>11</sup> docu-  
184 ment.

185 Apertis derived projects should provide an equivalent location for their specific  
186 exceptions.

## 187 Future improvements

188 Manually checking licenses will not scale and may not be done in a deterministic  
189 way. Introducing automation is a key.

190 FOSSology is a license reporting tool. It is described in the [Automated License  
191 Compliance](#)<sup>12</sup> document along with an approach to enable end-to-end tracking  
192 of licensing information. Although we trust the developer to check license, the  
193 use of FOSSology could help ensure correct identification.

## 194 Appendix

### 195 The Debian Free Software Guidelines (DFSG)

#### 196 1. Free Redistribution

197 The license of a Debian component may not restrict any party from selling or  
198 giving away the software as a component of an aggregate software distribution  
199 containing programs from several different sources. The license may not require  
200 a royalty or other fee for such sale.

---

<sup>11</sup><https://jwd.pages.apertis.org/apertis-website/policies/license-exceptions/>

<sup>12</sup><https://jwd.pages.apertis.org/apertis-website/concepts/automated-license-compliance/>

201       2. Source Code

202       The program must include source code, and must allow distribution in source  
203       code as well as compiled form.

204       3. Derived Works

205       The license must allow modifications and derived works, and must allow them  
206       to be distributed under the same terms as the license of the original software.

207       4. Integrity of The Author's Source Code

208       The license may restrict source-code from being distributed in modified form  
209       only if the license allows the distribution of "patch files" with the source code  
210       for the purpose of modifying the program at build time. The license must  
211       explicitly permit distribution of software built from modified source code. The  
212       license may require derived works to carry a different name or version number  
213       from the original software. (This is a compromise. The Debian group encourages  
214       all authors not to restrict any files, source or binary, from being modified.)

215       5. No Discrimination Against Persons or Groups

216       The license must not discriminate against any person or group of persons.

217       6. No Discrimination Against Fields of Endeavor

218       The license must not restrict anyone from making use of the program in a specific  
219       field of endeavor. For example, it may not restrict the program from being used  
220       in a business, or from being used for genetic research.

221       7. Distribution of License

222       The rights attached to the program must apply to all to whom the program is  
223       redistributed without the need for execution of an additional license by those  
224       parties.

225       8. License Must Not Be Specific to Debian

226       The rights attached to the program must not depend on the program's being  
227       part of a Debian system. If the program is extracted from Debian and used  
228       or distributed without Debian but otherwise within the terms of the program's  
229       license, all parties to whom the program is redistributed should have the same  
230       rights as those that are granted in conjunction with the Debian system.

231       9. License Must Not Contaminate Other Software

232       The license must not place restrictions on other software that is distributed  
233       along with the licensed software. For example, the license must not insist that  
234       all other programs distributed on the same medium must be free software.

235       10. Example Licenses

236       The "GPL", "BSD", and "Artistic" licenses are examples of licenses that we  
237       consider "free".