GPL-3 Deltas Assessment

Contents

Apertis the distribution is derived from Debian, from which it takes its philosophy, tools, workflows and packages. This robust, friendly and mature distribution provides a solid base on which to build an offering to suite the needs of very demanding markets such as the automotive industry.

One big difference between Apertis and Debian is that Apertis avoids certain licenses[1], in order to allow its target market to avoid legal issues. Several licenses are considered unsuitable in parts of Apertis, GPL-3 being the most important one. As a consequence of this, Apertis adopts a number of strategies to ensure packages meant to be installed on target devices comply with these license restrictions.

Several documents already cover specific cases or scenarios, which present the biggest licensing challenges:

- GPL-3-free replacements of coreutils[2]
- License-compliant TLS stack for Apertis targets[3]
- GPL-3-free replacements of GnuPG[4]

Besides the topics covered by the above documents, Apertis implements different strategies to avoid such problems. In the cases where package license changed from GPL-2 to GPL-3, Apertis continues shipping the last license friendly ver-

---

[1]https://martyn.pages.apertis.org/apertis-website/policies/license-expectations/
[2]https://martyn.pages.apertis.org/apertis-website/concepts/coreutils-replacement/
[3]https://martyn.pages.apertis.org/apertis-website/concepts/tls-stack/
[4]https://martyn.pages.apertis.org/apertis-website/concepts/gnupg-replacement/

sion of the package, appending the suffix `-gplv2` if it is needed to differentiate from the latest version

- readline5
- cpio-gplv2
- diffutils-gplv2
- findutils-gplv2
- grep-gplv2
- gzip-gplv2
- sed-gplv2
- tar-gplv2

In other cases, where the license issues was not in the package itself, but in one of its dependencies, Apertis tries to avoid the problem by either using a different equivalent dependency or using the last suitably licensed version of it. In those cases where the functionality provided by the dependency is not really required, Apertis opts for removing or disabling such functionality and in that way dropping the dependency.

Impact

As discussed in the introduction, depending on the situation the impact of a delta is different. Based on the type of delta we can enumerate the following scenarios:

- Delta causes outdated package to be shipped
- Delta causes alternative package dependency to be used when compared to Debian
- Delta causes functionality to be disabled

Additionally the following aspects should be taken into account:

- Possibility of delta increment across time
- Number of packages in the dependency change

Delta causes outdated package to be shipped

Since Apertis derives from Debian, generally it ships the same version, but as mentioned, in some cases it keeps shipping a specific version of a package for the `target` component, while keeping the latest in the `development` suite.

In general the impact of this kind of delta is high, since Apertis carries an old version of a package without updates and security bugfixes. For this reason deltas under this category should be examined closely, specially taking into account the aspects previously mentioned.

Below is a list of packages that are frozen at a specific version previous to the license change and the packages that depend on them in the `target` component.

3

- readline5 (version 5.2)
  - bluez
  - connman
- cpio-gplv2 (version 2.8)
  - initramfs-tools-core
- diffutils-gplv2 (version 2.8.1)
- findutils-gplv2 (version 4.2.31)
- grep-gplv2 (version 2.5.1a)
- gzip-gplv2 (version 1.3.12)
- sed-gplv2 (version 4.1.2)
- tar-gplv2 (version 1.17)
  - dpkg

From the list above it clear that `readline5 cpio-gplv2` and `tar-gplv2` are the package with higher impact in the system as they are used by other packages.

Delta causes alternative package dependency to be used

When it is possible to find an alternative to a package without license issues which provides similar functionality and it is present in Debian, the approach used is to switch to it, causing a delta. However, since the functionality is kept, the impact of the delta is considered lower than previous cases.

Delta causes functionality to be disabled

Under some circumstances, Apertis chooses to disable functionality to avoid a license issue. This approach is only valid if the functionality is not important, which requires an evaluation. Once it has been decided that the functionality is not a strong requirement a delta is introduced to disable it and drop dependencies which use unfriendly licenses. This generally only introduces a minor delta with respect to the package in Debian and is easy to maintain and port forward with updates in Debian.

Package summary

The table below shows the packages which have a license related delta with respect to Debian. They are split into the following categories based on the scenarios described above:

- DF0: Disable functionality
- DF1: Disable minor functionality
- OP: Outdated package
- AP0: Use alternative outdated package
- AP1: Use alternative package

Package	Category	Information
base-files	DF0	Remove license information for GPL-3 LGPL-3 and MPL-1.1
bind9	DF0	Disable libidn2
bluez	AP0	Use of libreadline-gplv2-dev
connman	AP0	Use of libreadline-gplv2-dev
coreutils-gplv2	OP	Outdated GPL-3 free version
cpio-gplv2	OP	Outdated GPL-3 free version
curl	DF0	Disable libidn2 librtmp
cyrus-sasl2	DF0	Disable saslfinger libdes and krb4
diffutils-gplv2	OP	Outdated GPL-3 free version
findutils-gplv2	OP	Outdated GPL-3 free version
flatpak	DF0	Disable gpg
glib-networking	AP1	Use openssl instead of gnutls
glibc	AP1	Avoid bashisms
gnupg2	XXX	Need to be moved to development
gpgme1.0	AP0	Use of gunpg, drop libassuan
grep-gplv2	OP	Outdated GPL-3 free version
gstreamer1.0	DF1	Disable libdw
gtk+3.0	DF1	Disable cups
gvfs	DF0	Disable trashlib
gzip-gplv2	OP	Outdated GPL-3 free version
initramfs-tools	AP0	Use coreutils-gplv2
libblockdev	DF0	Disable parted
libcanberra	DF0	Disable tdb
liboauth	AP1	Use curl openssl instead of curl gnutls
mesa	DF0	Disable libefl
mktemp	XXX	Empty package, implemented in coreutils
openjpeg2	AP1	Use curl openssl instead of curl gnutls
openldap	AP1	Use curl openssl instead of curl gnutls
ostree	DF0	Disable libgpgme
pam	DF0	Replace pam-auth-update, disable NIS
pipewire	DF0	Disable libsdl2
pulseaudio	DF0	Disable libtdb
readline5	OP	Outdated GPL-3 free version
sed-gplv2	OP	Outdated GPL-3 free version
systemd	DF0	Disable libdw, gnutls, libmicrohttpd
tar-gplv2	OP	Outdated GPL-3 free version
totem-pl-parser	DF1	Disable libquvi
tumbler	DF0	Use curl openssl instead of curl gnutls
udisks2	DF0	Disable parted
util-linux	DF1	Disable parse_date
v4l-utils	DF1	Disable gettext
webkit2gtk	DF1	Disable libenchant-2
wpa	AP1	Use internal line edit instead of readline

Required Action

We believe that the following actions are required to reduce the impact of these deltas. We have proposed different strategies depending on the impact of the delta, focusing on those which cause outdated packages to be shipped.

For the remaining cases, the impact is only related to drop functionality, which have little value for Apertis, in consequence we believe that the best approach is to keep the delta.

The strategies relies in find the best possible alternative, taking into account

- License: The replacement should meet Apertis license expectations[5] in order to be consider as a valid one
- Debian support: The Debian support guarantees a community support on the package and a easy adoption in Apertis
- Compatibility: The replacement should provide the functionality required by Apertis on target images. Since the focus is on embedded devices, this is usually a small subset of the functionality provided by a fully featured tool, designed to be used by a user from a command line. For example, several alternative command line tools may use different arguments to provide functionality, for which existing users can be trivially altered or lack certain options, but in many cases these options will have little or no value when used in Apertis.

Delta causes outdated package to be shipped

This type of delta is the most problematic and requires immediate action as these packages are currently not receiving security updates and thus present a security risk.

Package readline5

Source: https://tiswww.case.edu/php/chet/readline/rltop.html

The `readline5` package ships version 5.2 of `GNU readline`. It provides a set of functions for use by applications that allow users to edit command lines as they are typed in. This same functionality can be provided by:

- libedit[6]: This is an autotool- and libtoolized port of the NetBSD Editline library (libedit). This Berkeley-style licensed command line editor library provides generic line editing, history, and tokenization functions, similar to those found in GNU Readline.
    - License: BSD-3-Clause
    - Debian: Present
    - Apertis: Present (target)

---

[5]https://martyn.pages.apertis.org/apertis-website/policies/license-expectations/
[6]https://www.thrysoee.dk/editline/

- replxx[7]: A small, portable GNU readline replacement for Linux, Windows and MacOS which is capable of handling UTF-8 characters. Unlike GNU readline, which is GPL, this library uses a BSD license and can be used in any kind of program.

    – License: BSD-3-Clause
    – Debian: Not present
    – Apertis: Not present

Conclusion

Since `libedit` is a mature package, based on NetBSD Editline library and is already present in Apertis, it is the primary candidate as a replacement. The approach in this case is to add support for it as alternative for `readline` in the packages which depend on it (`bluez` and `connman`).

Package tar-gplv2

Source: https://www.gnu.org/software/tar/

Package `tar-gplv2` ships version 1.17 of `GNU tar` which provides the ability to create and manipulate tar archives. There are the following alternatives with the same functionality:

- libarchive[8]: Multi-format archive and compression library, which includes the `libarchive` library, the `bsdtar` and `bsdcpio` command-line programs, full test suite, and documentation.

    – License: BSD-2-clause
    – Debian: Present
    – Apertis: Present (target)
    – GNU compatibility: Medium, basic set of features

- busybox tar[9]: BusyBox combines tiny versions of many common UNIX utilities into a single small executable, `tar` among them.

    – License: GPLv2
    – Debian: Present
    – Apertis: Present
    – GNU compatibility: Low, only minimum set of features

- tar-rs[10]: Rust library to manage TAR archives.

    – License: Apache
    – Debian: Not present
    – Apertis: Not present

---

[7]https://github.com/AmokHuginnsson/replxx
[8]https://www.libarchive.org/
[9]https://busybox.net/
[10]https://github.com/alexcrichton/tar-rs

Conclusion

The package `libarchive` is mature and already in Apertis. It provides `bsdtar` which gives a good basement to build a replacement for `tar`. The approach in this case is to test the use case of interest for `target` images, to install packages with `dpkg`.

Initial tests replacing `tar` with `bsdtar` or `busybox tar` and installing a package

```
$ sudo apt reinstall libc6
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
  libcolord2 libegl1-mesa libsys-cpuaffinity-perl libxdelta2 pbzip2 pixz xdelta xdelta3
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 1 reinstalled, 0 to remove and 0 not upgraded.
Need to get 2,831 kB of archives.
After this operation, 0 B of additional disk space will be used.
Get:1 https://repositories.apertis.org/apertis v2022dev2/target amd64 libc6 amd64 2.31-
9apertis2bv2022dev2b1 [2,831 kB]
Fetched 2,831 kB in 3s (887 kB/s)
debconf: unable to initialize frontend: Dialog
debconf: (No usable dialog-like program is installed, so the dialog based frontend cannot be used. at /usr/share
debconf: falling back to frontend: Readline
Preconfiguring packages ...
-x -f - --warning=no-timestamp
-x -f -
bsdtar: Option --warning=no-timestamp is not supported
Usage:
  List:    bsdtar -tf <archive-filename>
  Extract: bsdtar -xf <archive-filename>
  Create:  bsdtar -cf <archive-filename> [filenames...]
  Help:    bsdtar --help
dpkg-deb: error: tar subprocess returned error exit status 1
dpkg:    error    processing    archive    /var/cache/apt/archives/libc6_2.31-
9apertis2bv2022dev2b1_amd64.deb (--unpack):
 dpkg-deb --control subprocess returned error exit status 2
Errors were encountered while processing:
 /var/cache/apt/archives/libc6_2.31-9apertis2bv2022dev2b1_amd64.deb
E: Sub-process /usr/bin/dpkg returned an error code (1)
```

After omitting the argument the process finish without issues.

Package cpio-gplv2

Source: https://www.gnu.org/software/cpio/

Package `cpio-gplv2` ships version 2.8 of `GNU cpio` which is used to copies files into or out of a cpio or tar archive. The archive can be another file on the disk, a magnetic tape, or a pipe. This same functionality can be provided by:

- libarchive[11]: Multi-format archive and compression library, which includes the `libarchive` library, the `bsdtar` and `bsdcpio` command-line programs, full test suite, and documentation.

  − License: BSD-2-clause
  − Debian: Present
  − Aperts: Present
  − GNU compatibility: Medium, basic set of features

- busybox cpio[12]: BusyBox combines tiny versions of many common UNIX utilities into a single small executable, `cpio` among them.

  − License: GPLv2
  − Debian: Present
  − Apertis: Present
  − GNU compatibility: Low, only minimum set of feature

- cpio-rs[13]: Rust library to manage CPIO archives.

  − License: MIT License
  − Debian: Not present
  − Apertis: Not present

Conclusion

The package `libarchive` is mature and already packaged in Apertis. This provides `bsdcpio` as a good base to build a replacement for `cpio`. In this case we need to test if it can successfully be used to build the initramfs used in Apertis.

Initial test, replacing `cpio` with `bsdcpio` and `busybox cpio` and running `update-initramfs`, was successful with no errors.

Package diffutils-gplv2

Source: https://www.gnu.org/software/diffutils/

Package `diffutils-gplv2` ships version 2.8.1 of `GNU diffutils`, a set of programs to find differences between files. Similar functionality can be obtained by:

- busybox diff[14]: BusyBox combines tiny versions of many common UNIX utilities into a single small executable, `diff` among them.

  − License: GPLv2
  − Debian: Present

---

[11]https://www.libarchive.org/
[12]https://busybox.net/
[13]https://github.com/jcreekmore/cpio-rs
[14]https://busybox.net/

- Apertis: Present
- GNU compatibility: Low, only minimum set of feature

- ccdiff[15]: Perl script to achieve same functionality than `diff` but improving the visual output with colors.

    - License: Artistic-2.0
    - Debian: Present
    - Apertis: Not present
    - GNU compatibility: High
    - Runtime dependencies:
        * libalgorithm-diff-xs-perl (not in Apertis - Artistic)
        * libalgorithm-diff-perl (development - Artistic)
        * libscalar-list-utils-perl (development - Artistic)

- colordiff[16]: The Perl script `colordiff` is a wrapper for `diff` and produces the same output but with pretty 'syntax' highlighting. Colour schemes can be customized.

- rust-diff[17]: A rust library to compute text diffs.

Conclusion

The most suitable replacement found is `busybox diff`, since it provides the basic functionality required on target images. Initial tests shows that `ccdiff` has same functionality, very similar arguments and similar output (adds colors) to `diff`. However, since it is a `perl` script it requires additional dependencies to be installed.

Additionally it was found that `diff` is used on package install by `dpkg` but the process runs smoothly with `busybox diff` and also with `ccdiff`. The features of `cmp`, `diff3` and `sdiff` are not supported, however there is not much value in `target` images.

Package findutils-gplv2

Source: https://www.gnu.org/software/findutils/

Package `findutils-gplv2` ships version 4.2.31 of `GNU findutils` a set of basic directory searching utilities. Alternatives to this package are:

- busybox find/xargs[18]: BusyBox combines tiny versions of many common UNIX utilities into a single small executable, `find` and `xargs` among them.

    - License: GPLv2
    - Debian: Present
    - Apertis: Present

---

[15]https://metacpan.org/pod/App::ccdiff
[16]https://www.colordiff.org/
[17]https://docs.rs/diff/0.1.12/diff/
[18]https://busybox.net/

- GNU compatibility: Low, only minimum set of feature
- uutils-findutils[19]: A rust implementation of `findutils`
  - License: MIT License
  - Debian: Not present
  - Apertis: Not present
  - GNU compatibility: High in mind, however it is in early stage of development

Conclusion

Currently the best approach to have a replacement for the set of utilities provided by `findutils` is to use `busybox find` and `busybox xargs`. These are already present in Apertis and their limited functionality is not impacting the existing limited usage by the packages which depend on them.

The package `uutils-findutils` is being developed by the same community which develops `uutil-coreutils`, which has been chosen by Apertis as a replacement for `coreutils` based on its pros.

- High GNU compatibility
- High community support
- High community impact
- Portability in mind
- Ongoing development
- Implemented in a modern memory safe language

However, it is in an early stage of development and thus we recommend to wait for it to mature and then re-evaluate it as a replacement for `find`. Additionally it is important to mention that it does not yet support `xargs`.

Unfortunately initial tests with `busybox find` show that additional functionality is required by `update-initramfs`

```
find: unrecognized: -printf
find: unrecognized: -regextype
```

A similar issue is found with `uutils-findutils find` which triggers

```
find: unrecognized: -printf
```

These limitations needs to be addressed before switching to it.

Package grep-gplv2

Source: https://www.gnu.org/software/grep/

Package `grep-gplv2` ships version 2.5.1a of GNU grep, which searches one or more input files for lines containing a match to a specified pattern. By default, `grep` outputs the matching lines.

---

[19]https://github.com/uutils/findutils

- busybox grep[20]: BusyBox combines tiny versions of many common UNIX utilities into a single small executable, `grep` among them.

  - License: GPLv2
  - Debian: Present
  - Apertis: Present
  - GNU compatibility: Low, only minimum set of feature

- ugrep[21]: A grep alternative aim to be faster and with additional features.

  - License: BSD-3-Clause License
  - Debian: Present
  - Apertis: Not present
  - GNU compatibility: High
  - Runtime dependencies:
    * libbz2-1.0 (target)
    * libc6 (target)
    * libgcc-s1 (target)
    * liblz4-1 (target)
    * liblzma5 (target)
    * libpcre2-8-0 (target)
    * libstdc++6 (target)
    * libzstd1 (target)
    * zlib1g (target)

Conclusion

The goal to provide the required features for `target` images can be accomplish by using `busybox grep` without adding additional packages, making it the best option. Initial tests booting an image and installing packages don't show any issues.

It is worth mentioning that in cases where higher compatibility with GNU is required, the `ugrep` package is already in Debian and all its dependencies are already in `target`, making it a viable alternative.

Package gzip-gplv2

Source: https://www.gnu.org/software/gzip/

- busybox gzip[22]: BusyBox combines tiny versions of many common UNIX utilities into a single small executable, `gzip` among them.

  - License: GPLv2
  - Debian: Present
  - Apertis: Present
  - GNU compatibility: Low, only minimum set of feature

---

[20]https://busybox.net/
[21]https://github.com/Genivia/ugrep
[22]https://busybox.net/

362   • flate2-rs[23]: Rust library to manage ZIP archives.

363   – License: Apache
364   – Debian: Not present
365   – Apertis: Not present

366 **Conclusion**

367 In order to replace `gzip` the best alternative is to used `busybox gzip`, which even
368 with the its limitations it is enough for the requirements in target images

369 **Package sed-gplv2**

370 **Source**: https://www.gnu.org/software/sed/

371 Package `sed-gplv2` ships version 4.1.2 of `GNU sed` a non-interactive command-line
372 text editor.

373   • busybox sed[24]: BusyBox combines tiny versions of many common UNIX
374   utilities into a single small executable, `sed` among them.
375   – License: GPLv2
376   – Debian: Present
377   – Apertis: Present
378   – GNU compatibility: Medium, only minimum set of feature, but there
379   are not much difference

380 **Conclusion**

381 In order to provide a replacement for `sed-gplv2` the use of `busybox sed` is recom-
382 mended since no other package depends on and the basic functionality provided
383 by `busybox sed` covers most common use cases.

384 Initial tests

385 Besides of the partial tests done when analyzing each package, as part of the
386 initial test the following actions have been done

387   • Boot target image with tools replaced
388   • Reinstall all the packages in target image

389 These rest were passed successfully which shows that the suggested approach
390 is viable. Despite this promising results further testing should be conducted to
391 assure a smooth transition.

---

[23]https://github.com/rust-lang/flate2-rs
[24]https://busybox.net/