

Architectural criteria of the reusing existing OSS

Naoto YAMAGUCHI

This document define to "reuse program management" process criteria. It presents the criteria for the selection of existing OSS for use in an AGL distribution for the Instrument Cluster.

This document aim to create common agreement for reuse existing OSS both community and industry.

1. License

1.1. Basis

Assessing the license of the OSS.

- The OSS must be released under a license included in the allow list in Table 1-1.
- When the OSS license is not included in the allowed list, it must be confirmed that it is not included in the deny list in Table 1-2.
- When the OSS license is not listed on both lists, this license must be judged by AGL Instrument Expert Group and accepted by SAT.

Table 1-1. Allow license list

No.	License name	License URL
1	GNU General Public License, version 2	https://www.gnu.org/licenses/old-licenses/gpl-2.0.txt
2	GNU Lesser General Public License, version 2.1	https://www.gnu.org/licenses/old-licenses/lgpl-2.1.en.html
3	Apache License 2.0	https://www.apache.org/licenses/LICENSE-2.0
4	3-clause BSD license	https://opensource.org/licenses/BSD-3-Clause
5	2-clause BSD license	https://opensource.org/licenses/BSD-2-Clause
6	MIT License	https://opensource.org/licenses/mit-license.php
7	Mozilla Public License 2.0	https://www.mozilla.org/en-US/MPL/2.0/
8	zlib/libpng License	https://opensource.org/licenses/Zlib
9	Boost Software License 1.0	https://opensource.org/licenses/BSL-1.0
10	GCC Runtime Library Exception	https://www.gnu.org/licenses/gcc-exception-3.1.en.html

Table 1-2. Deny license list

No.	License name	License URL
1	GNU General Public License, version 3	https://www.gnu.org/licenses/gpl-3.0.en.html
2	GNU Lesser General Public License, version 3	https://www.gnu.org/licenses/lgpl-3.0.en.html
3	GNU Affero General Public License version 3	https://opensource.org/licenses/AGPL-3.0

*The GPLv3 and GPLv3 like license does not allow **tivoization**. This is incompatible with embedded use cases.

1.2. Special case

Licensing restrictions should be relaxed for some use cases such as debugger, host tools and analysis tools. In this document, these use cases are calling the exception use cases.

The OSS used in the exception use case, that must block automatically to installing on the final target image using integration system.

Table 1-3 and Table 1-4 shows exception for Table 1-1 and Table 1-2. In excepted use-case can use license both Table 1-1 and Table 1-3. When the same license appears in more than one table, Table 1-3 is preferred over Table 1-2, Table 1-4 is preferred over Table 1-1.

Table 1-3. Special allow license list

No.	License name	License URL
-----	--------------	-------------

1	GNU General Public License, version 3	https://www.gnu.org/licenses/gpl-3.0.en.html
2	GNU Lesser General Public License, version 3	https://www.gnu.org/licenses/lgpl-3.0.en.html

***The GPLv3 and GPLv3 like license does not allow tivoization. When these software only to use debugging (not installing in final product), it's no problem.**

Table 1-2. Special deny license list

No.	License name	License URL

2. Health of the community

Assessing the health of the community that develops OSS. This requirement defines criteria for OSS selection in AGL Instrument Cluster Distribution.

- The OSS community must shall be established the rules shown in Table 2-1.

Table 2-1. Community check list

No.	Requirement	Example	Req. Level
1	Defining the coding rule or guideline	https://www.kernel.org/doc/html/latest/process/coding-style.html https://www.gnu.org/prep/standards/standards.html https://systemd.io/CODING_STYLE/	Must
2	Defining the contribution rule	https://www.kernel.org/doc/html/latest/process/code-of-conduct-interpretation.html https://gcc.gnu.org/contribute.html https://systemd.io/CONTRIBUTING/	Must
3	Defining the release rule.	https://www.gnu.org/software/libc/ https://github.com/openssl/openssl/releases https://github.com/wayland-project/weston/releases	Must
4	Providing a change logs.	https://sourceware.org/legacy-ml/libc-announce/2020/msg00001.html https://cdn.kernel.org/pub/linux/kernel/v5.x/ChangeLog-5.4.54	Must
5	Have a bug tracking system or other bug report and fix solution such as active mailing list, github issue, etc..	https://lkml.org/ https://github.com/systemd/systemd/issues https://bugzilla.redhat.com/	Should
6	Have and maintain a test suite.	https://github.com/linux-test-project/ltp https://github.com/systemd/systemd/tree/master/test https://wiki.musl-libc.org/libc-test.html	Should
7	Used in popular distributions such as RHEL, SUSE, Ubuntu, Debian.		Should
8	2 or more active contributors.	https://www.openhub.net/explore/projects	Should
9	Including OIN(Open Invention Network) packages list	https://www.openinventionnetwork.com/joining-oin/linux-system/linux-system-table/?cat_id=15&type=table	Recommend

3. Long Term Stable

The LTS (Long Term Stable) must be provided by one of the following means

1. LTS support by upstream community.
2. Use of existing LTS distribution down stream code base.

4. Source code assessment

Assessing the code quality of the OSS.

This criteria aim to certify to the version independent OSS quality. The candidate OSS shall evaluate for code quality using static analysis tools.

1st step is analyzing for history of code quality using static analysis tool. Has a serious bug been fixed with the minor version up? When major version up is made, how many new serious bugs increase this OSS?

This analysis cannot be based on the number of bug fix. It need to use a static analysis tool to analyze the unfixed bugs.

These OSS must not include "must fix" error from static analysis tool.

Note. The validity of the version used by that OSS, including CVE checks, will be checked in the next phase.

5. Requirement matching

All requirements assigned to the OSS must be met.

If the requirements are not met, the software must be reassigned to another OSS or AGL developed code.