

Version 2.0

This is version 2.0 of the REUSE practices, released 2017-12-14.

1. Provide the exact text of each license used

Free and open source software licenses are standardised and have standard texts. Regardless of which license you use, you should include the license text in your project.

You must include all licenses which are used in your project, and you must never change any license texts even if they are very similar to existing ones.

If you are adding a new license file, the easiest way to make sure that you provide an unchanged license is by copying it verbatim from [this repository of licenses](#) maintained by the SPDX Workgroup. The SPDX Workgroup maintains a [list of commonly-used licenses](#). Each license has a shorthand name (called an SPDX identifier) that can be used to uniquely reference each license. For instance, the GNU GPL version 3 has `GPL-3.0` as shorthand identifier.

If your project only includes code licensed under a single license, you may provide the text of this license in a file in the top level directory of your repository with the name `LICENSE`, `LICENCE`, `COPYING` or `COPYRIGHT` (you may attach some suffix to the filename as well, such as `LICENSE.txt`).

Since many projects include code under different licenses, it's often not feasible to include all licenses in the top level, in which case you should create a directory at the top level called `LICENSES` within which you include the license text of each license used.

Some licenses, like the BSD 2 Clause license, exist in many variations where the only difference is the name of the copyright holder in the license text. Your project may end up including multiple versions of the same BSD 2 Clause license because some parts may be written by Alice and others by Bob, resulting in two different license files.

When you have added all of your license files, they should all match the glob `{LICENSE*, LICENCE*, COPYING*, COPYRIGHT*, LICENSES/**}`. These are the common places to find licenses. If you place your license elsewhere, it is possible that humans and computers alike may overlook it.

In order to be able to reference the license file from your source code, you should embed the SPDX identifier (e.g., `GPL-3.0`, `MIT`) of each license into the corresponding license file. The easiest way to do this is to simply include the name of the license into the filename: The license file for `GPL-3.0` becomes `LICENSES/GPL-3.0.txt`.

If it is not possible to rename your file (for instance, if you want to have the license text under `COPYING`), then you should add a header to the file in tag-value format. You can add this header either to the license file itself, or in a file with the same filename suffixed with `.license`. The header must include a `Valid-License-Identifier` tag, the value of which should be the SPDX identifier for your license. The header should terminate with the `License-Text` tag to signal the start of your license. The header may include other tags.

This is an example of an MIT license file, which conveys that all references to the SPDX identifier MIT should point to this license file.

```
Valid-License-Identifier: MIT
License-Text:
```

```
MIT License
```

```
Copyright (c) <year> <copyright holders>
```

```
Permission is hereby granted, free of charge, to any person obtaining a
copy of this software and associated documentation files (the "Software"),
etc...
```

Note: Even though the copyright statement in the above example is not in accordance to the REUSE spec, we keep the same form as it is in the canonical license template, so tools identify it as the same MIT license.

While the identifier should be a license in the SPDX license list, it should be noted that the license text itself may not necessarily be released under that license, nor does the tag claim that.

The license identifier may be repeated if the same license file is used with multiple license identifiers. The example below is for `GPL-2.0` and `GPL-2.0+`, but please note that it is not mandatory to list `GPL-2.0+` separately.

```
Valid-License-Identifier: GPL-2.0
Valid-License-Identifier: GPL-2.0+
```

If the license is not included in the SPDX license list, or the license is a modified version of a license included in that list, you should create a custom identifier for it in the form `LicenseRef-<unique_code>`.

```
Valid-License-Identifier: LicenseRef-MyLicense
```

In the unlikely event your source code includes exceptions, such as the [Classpath exception 2.0](#), you may include them in the same place as your licenses, using the same header style or the tags `SPDX-Exception-Identifier` and `Exception-Text` instead. This is an example of a valid exception:

```
SPDX-Exception-Identifier: Classpath-exception-2.0
Exception-Text:
```

```
Linking this library statically or dynamically with other modules is
making a combined work based on this library. Thus, the terms and
conditions of the GNU General Public License cover the whole combination.
```

Keep in mind

- Don't change any license texts, use the verbatim form of the license text.
- Don't remove any license texts, include the license texts of all software.
- Include the SPDX identifier in each license file, either by including it in the filename or by inserting a header, such that you can refer to the license from the source code files.

2. Include a copyright notice and license in each file

Source code files are often reused across multiple projects, taken from their origin and repurposed, or otherwise end up in repositories where they are separate from its origin. You should therefore ensure that all files in your project have a comment header that convey that file's copyright and license information: Who are the copyright holders and under which license(s) do they release the file?

It is highly recommended that you keep the format of these headers consistent across your files. It is important, however, that you do not remove any information from headers in files of which you are not the sole author.

You must convey the license information of your source code file in a standardised way, so that computers can interpret it. You can do this with an `SPDX-License-Identifier` tag followed by an SPDX expression defined by the SPDX specifications.

Copyright notices in a comment header should have a consistent format and be sorted by year. They should list the actual copyright holder, which may be an organisation rather than the author. For example:

```
/*
 * © 2017 Alice Commit <alice@example.com>
 * © 2009–2016 Bob Denver <bob@example.com>
 * © 2007 Company Example <charlie@example.com>
 *
 * SPDX-License-Identifier: GPL-2.0
 */
```

Alternatively, if you cannot use the Unicode copyright sign ©, you may use “Copyright” as a full word instead. For example:

```
/*
 * Copyright 2017 Alice Commit <alice@example.com>
 *
 * SPDX-License-Identifier: GPL-2.0
 */
```

You should include information about your project’s practices in the README or similar file. This could be as simple as linking to these practices.

All source files which are included in the final build of your project (verbatim, modified, partially, or in compiled form) must have copyright and license information contained within them. If it is not possible to add a header to the file, either because the file is a binary or because the format does not support comments, you must provide the information separately in one of three ways:

- You may include a textfile called `FILENAME.license` in which you store your regular comment header with all the necessary information to convey the license information about the file called `FILENAME`.
- If you have many such files, but each have the same copyright notice and license, you may instead use the [DEP-5/copyright](#) file format, and place a single copyright notice documenting those files.
- If you have a way of automatically generating an inventory of included files in SPDX format (see practice 3), you may use this and do not need to separately include a `.license` file or `DEP-5/copyright` file.

If you wish to be explicit about the license of an output file, which does not exist in the repository but which will be created at build time, you may include a `.license` file without the corresponding binary file.

If your project includes documentation which is not included in the final build artefact, you are encouraged but not obliged to include license information for these too.

If the source code is licensed under a license such as `BSD-2-Clause`, where the license texts may differ for each different copyright holder, you should reference the specific license each time. Thus for files written by Alice, reference

to `LicenseRef-BSD-2-Clause-Alice`, and for files written by Evil Corp, reference to `LicenseRef-BSD-2-Clause-Evil-Corp`, and include both those licenses separately in `LICENSES` (see practice 1).

You may record information about copyright by relying on the underlying version control system you're using, but special care must be taken if you do. Keep in mind version control systems typically record authorship, not copyright.

For a project using the version control system to convey information about copyright, you must:

- make sure the version control system is publicly accessible and take steps¹ to ensure it will remain so,
- provide a link back to the original file in version control from each header,
- ensure that if someone copies the source repository without version control system metadata (such as if they make a tar-file of it), then the version control system metadata relevant for determining copyright must still be included, for example in the form of a bill of materials (see later), an automatically generated log file or similar,
- make sure the commit metadata reflect the actual copyright (this is particularly important if a project accepts code contributed through mailing lists, bug trackers, or similar, where the original copyright holder is not the one pushing code to version control, or where the copyright of the contribution is held by an organisation rather than the author, as for example the author's employer.),
- take into account what kind of metadata version control systems store. For example, Git stores the committer and author in two different fields.

An appropriate header in this case would be:

```
/*
 * This file is part of project X. It's copyrighted by the contributors
 * recorded in the version control history of the file, available from
 * its original location http://git.example.com/X/filename.c
 *
 * SPDX-License-Identifier: GPL-3.0
 */
```

Keep in mind

- Use a consistent style of your headers throughout the project. Preferably use the form `© Alice Committer <alice@example.com>`.

¹Establish project governance committed to public access, vet material added to project to protect against vulnerability to takedown, only include material under a free and open source license so it can be copied and archived freely and ensure project is archived by institutions dedicated to long-term preservation of software code, eg. Software Heritage.

- Don't remove existing headers, but only add to them.
- If you depend on a version control system to keep a record of copyright holders do make sure to keep your version control system public and that you have processes in place to store copyright holders instead of just authors.
- Make references to the SPDX identifier from each source code file.
- Record copyright information in each source code file.
- Include license and copyright information also for files which can not include a proper header, either in a separate .license file, or using the SPDX or DEP-5/copyright specification.

3. Provide an inventory for included software

Aside from the license files included in the project, and the file level copyright information, you may include a bill of materials for your project, but you should only do so if this is generated automatically.

A bill of materials can be very complicated and lengthy, making it difficult to maintain. If you do not generate it automatically, it's very likely someone will forget to update it when making changes. In these cases, it's best not to have a bill of materials, but to rely only on the information coupled with the source code files.

If you do have a way of automatically creating a bill of materials, and if you choose to do so, you should generate it automatically from the most reliable information you have about each file in your project. This includes copyright information kept in version control systems and licenses on files which include a standard header, or which includes the header in a separate license file.

You may also choose to include in the bill of materials your output files generated when compiling the project, such that you can signal through the bill of materials, which license is relevant for the output files depending on the included source code.

The bill of materials should be conformant to the SPDX specification and included in a file in the top level directory of your repository called `LICENSE.spdx`.

Keep in mind

- Don't create a bill of materials if you can't generate it automatically.
- If you generate one automatically, it's helpful to include one.
- Make your bill of materials conformant to the SPDX specification.

- It doesn't hurt to run your project through ScanCode or FOSSology, to make sure these tools can parse and understand your project's licensing.