

REUSE

Making copyrights and licenses *computer* readable.



About Us



FSFE is a charity that empowers users to control technology. Software is deeply involved in all aspects of our lives; and it is important that this technology empowers rather than restricts us. By greasing the legal wheels which underpin free and open source software, the FSFE contributes to a healthy and lively ecosystem with minimum fuss.





What is REUSE?



An important initiative

- REUSE is seeking to make working with copyrights and licenses a little bit easier.
- We do this by making license information and copyrights in software computer readable (and understandable)



Git repos and templates

- It is also Git repositories which follow the best practices, to be forked as the starting point for projects.
- These Git repositories work as templates to show the best practices in action.



Best practices

- REUSE is a set of three basic best practices focusing on making licensing understandable.



A REUSE badge

- For projects compliant with the REUSE best practices, there's also a badge for README.md and similar!





Best practice overview



1

License texts

Provide the exact text of each license used, in verbatim form, without removing any existing license texts.

2

License header

Include a copyright notice and license in each file, with a consistent style, with a reference to the license text and an appropriate SPDX License Identifier.

3

Inventory

Provide an inventory for included software, but only if you can generate it automatically!



“The Software Package Data Exchange® (SPDX®) specification is a standard format for communicating the components, licenses and copyrights associated with software packages.”



1. License texts

Provide the exact text of each license used

Three overlapping stars in yellow, cyan, and blue in the top-left corner.

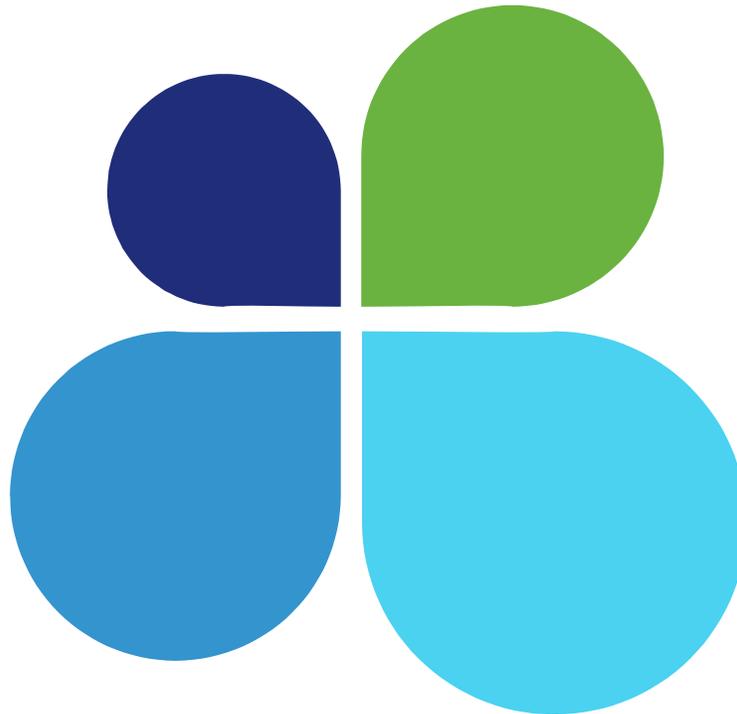
1. License texts

Standard texts

License texts are standardised. Include the verbatim form of the license you use, and include the verbatim license text of any 3rd party code included in your project.

LICENSES/

If there are more licenses, as there often are, include each one in a subdirectory called LICENSES/



LICENSE{.md}

Include the license in your repository, and give it the filename LICENSE, potentially with a suffix like LICENSE.md.

Don't change!

Don't change any existing license texts. Even if two license texts look almost identical, if both are relevant for your project, include both!



2. License headers

Include a copyright notice and license in each file



2. License headers

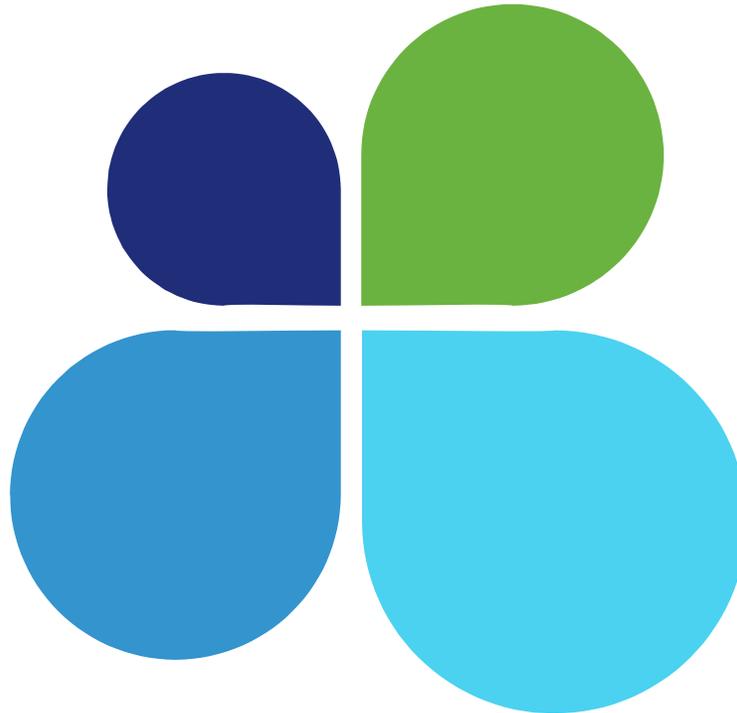


Be consistent

Use the same header style throughout your project, so computers can learn to read it more easily.

Link back to license

Each header should include a link back to the license file, together with an SPDX license identifier.



Headers everywhere

Include a license header even for files which don't support headers. For `foo.jpg`, place the license header in `foo.jpg.license`. Or use `DEP-5/copyright` format.

Be vary of VCS

You can use a version control system to track copyright and license information, but you must take special care if you do so.



3. Inventory

Provide an inventory for included software



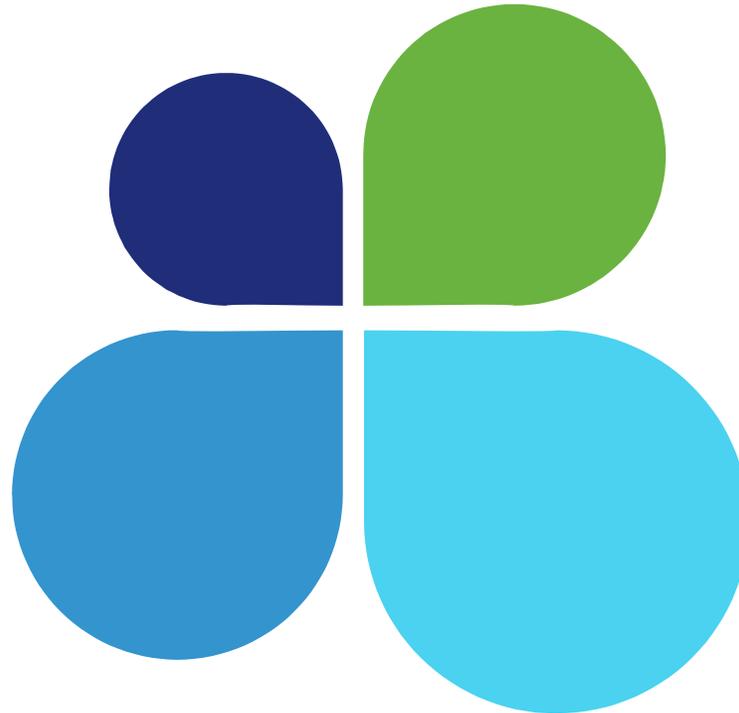
3. Inventory

Don't do this first

Make sure you follow best practices #1 and #2 first, before you consider an inventory.

Don't do it manually

It's very difficult to keep an inventory updated if you do it manually. Always use tools to generate your inventory.



Use SPDX

If you do include an inventory, make it compliant to the SPDX specification, which is widely used.

Verify & validate

Use tools like FOSSology and ScanCode to validate your code and verify the inventory.



Catch the details on
<http://reuse.software/>



REUSE
SOFTWARE



Free Software Foundation Europe e.V.
Schönhauser Allee 6/7
Stairway 2, 5. floor
10119 Berlin
Germany

Phone: +49-30-27595290



Hi everyone, I'm [name] from the Free Software Foundation Europe and I want to talk to you today about an initiative we're running to increase the level of automation we can have in processes which relate to managing free and open source software copyrights and licenses.

[Use the following if the audience is unfamiliar]

Before we start, I would like to emphasize that the FSFE, which I'm representing, and the FSF, are two distinct organisations. We may share a name, but we are otherwise completely separate, with different leadership, activities, and each with its own way of working.



About Us



FSFE is a charity that empowers users to control technology. Software is deeply involved in all aspects of our lives; and it is important that this technology empowers rather than restricts us. By greasing the legal wheels which underpin free and open source software, the FSFE contributes to a healthy and lively ecosystem with minimum fuss.



2

The FSFE is a non-profit organisation headquartered in Berlin, Germany, working on advancing issues important for the free and open source software ecosystem.

We work on legal issues by hosting the world's largest network of legal experts on free software.

We work on policy, largely on the European arena as that is closest to us, and we work on public awareness.

What is REUSE?



An important initiative

- REUSE is seeking to make working with copyrights and licenses a little bit easier.
- We do this by making license information and copyrights in software computer readable (and understandable)



Git repos and templates

- It is also Git repositories which follow the best practices, to be forked as the starting point for projects.
- These Git repositories work as templates to show the best practices in action.



Best practices

- REUSE is a set of three basic best practices focusing on making licensing understandable.



A REUSE badge

- For projects compliant with the REUSE best practices, there's also a badge for README.md and similar!



The REUSE initiative is important to us because while there are plenty of practices which show how to make the license choices of a project understandable to humans, there aren't as much information about how to make it understandable for computers.

We believe making this information computer understandable is key to increasing the automation we can see when it comes to managing copyrights and licenses.

REUSE is a set of three basic best practices meeting the need for computer understandable licenses and copyrights.

It's also a set of Git repositories which show how this work in practice, which people can fork or be inspired by as templates.

And finally, there's a REUSE compliant badge which projects meeting these practices can use.



Best practice overview



To look at the best practices, let me first give you an overview, before getting to the details.

Our first best practice is focusing on license texts.

Provide the **exact** text of each license used, in verbatim form, without removing any text. This helps a computer understand which licenses are identical.

Two, include a license header in each file, and make that header consistent in style. So if a program learns to read one header, it can read all headers in your software. Each header should also include a reference to the license text, and an SPDX License Identifier. I'll talk about SPDX in a second.

Three, provide an inventory, a bill of material, for your software, but **only** if you can generate it automatically. In practice, manual inventories don't get updated as they should, so don't use them.



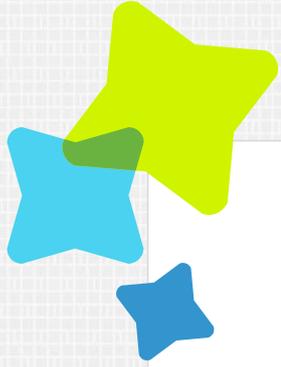
"The Software Package Data Exchange® (SPDX®) specification is a standard format for communicating the components, licenses and copyrights associated with software packages."



SPDX, or Software Package Data Exchange, is a standard for conveying information about licenses and copyrights in software packages. It's not the only one, there are a few others, but it's a reasonably popular one which meets the requirements we have for our best practices.

We only use parts of SPDX though. The complete specification is quite verbose and allow for a great many scenarios. We use the SPDX License Identifier, which is a unique identifier for each free and open source license.

We also use the SPDX format for inventories, but as you remember, we don't ever recommend anyone write or update SPDX information manually. So the only part of the SPDX standard you would usually be concerned with, is the SPDX License Identifier.



1. License texts

Provide the exact text of each license used

Moving on with our best practices.
One, provide the exact text of each license used.



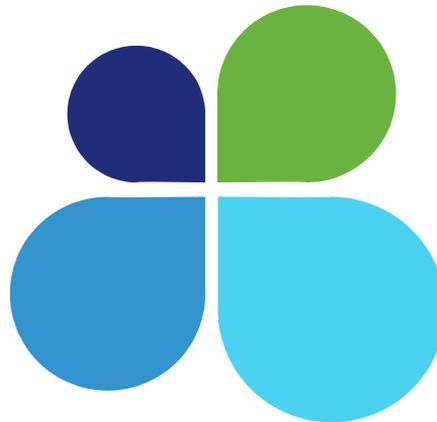
1. License texts

Standard texts

License texts are standardised. Include the verbatim form of the license you use, and include the verbatim license text of any 3rd party code included in your project.

LICENSES/

If there are more licenses, as there often are, include each one in a subdirectory called LICENSES/



LICENSE{.md}

Include the license in your repository, and give it the filename LICENSE, potentially with a suffix like LICENSE.md.

Don't change!

Don't change any existing license texts. Even if two license texts look almost identical, if both are relevant for your project, include both!

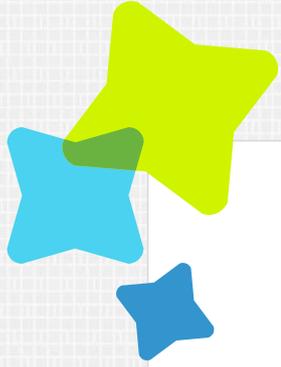
Licenses are standardised and have standard texts.

Use them, in verbatim format. So that a computer can compare two license texts and easily see they are identical. Also include any license text verbatim for any third party code included in your project.

If you have a single license, put it in a LICENSE file in the top level directory of your repository.

If you have multiple licenses, create a LICENSES/ folder in which you put all the different license texts which apply.

Don't ever change an existing license text. You may end up with several almost identical license texts in your LICENSES/ folder, but if they deviate by so much of a single character, it's best to include both versions relevant. Again, this makes it easier for a computer to relate licenses with one another and between projects.



2. License headers

Include a copyright notice and license in each file

Two, include a copyright notice and license in each file.



2. License headers

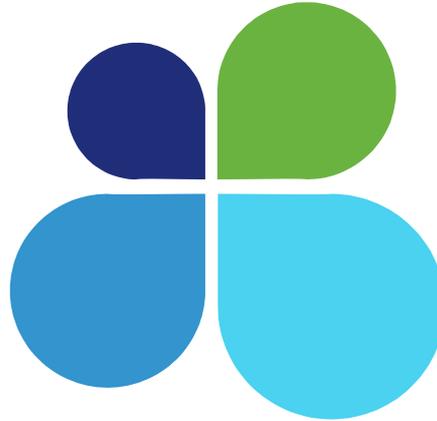


Be consistent

Use the same header style throughout your project, so computers can learn to read it more easily.

Link back to license

Each header should include a link back to the license file, together with an SPDX license identifier.



Headers everywhere

Include a license header even for files which don't support headers. For `foo.jpg`, place the license header in `foo.jpg.license`. Or use DEP-5/copyright format.

Be vary of VCS

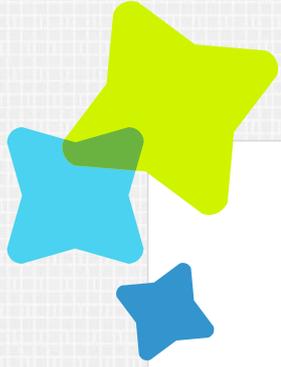
You can use a version control system to track copyright and license information, but you must take special care if you do so.

Don't touch existing headers if they are there, but add to them. And be consistent in what your own headers look like, so that if a computer learns to read one header, it can read the other headers too.

Each header should include an SPDX License Identifier, and a reference back to the top level LICENSES/ directory and the filename of the license text relevant for that file.

If a file can't contain a full license header, like images or other files, then create a separate file with the same filename but ending with "dot license", and include the license header in it. You may also look at the Debian DEP-5/Copyright format and use it to specify licenses for files which can not include license headers.

You can use version control systems to manage copyrights, and it's sometimes attractive to do so. But if you do, we have some demands, such as that the version control system should be publicly accessible, and that you take steps to keep it so.



3. Inventory

Provide an inventory for included software

Three, provide an inventory for included software.



3. Inventory

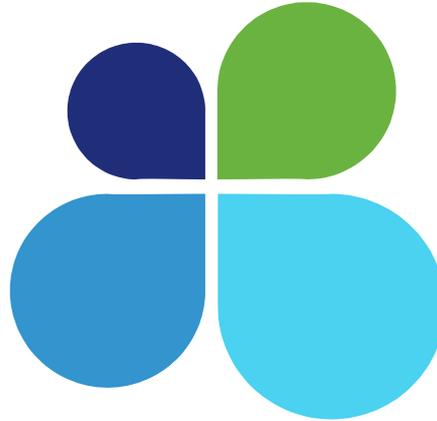


Don't do this first

Make sure you follow best practices #1 and #2 first, before you consider an inventory.

Don't do it manually

It's very difficult to keep an inventory updated if you do it manually. Always use tools to generate your inventory.



Use SPDX

If you do include an inventory, make it compliant to the SPDX specification, which is widely used.

Verify & validate

Use tools like FOSSology and ScanCode to validate your code and verify the inventory.

Make sure you comply with best practice #1 and #2 first, before you think about an inventory.

Practice shows it's very difficult to maintain an inventory manually. Every change requires an update not only of the file you're working with, but also a copyright inventory which is somewhere else in the repository. This leads to inventories becoming obsolete over time, and when it comes to either having an obsolete and inaccurate inventory, or not having one at all, it's better to not have one at all.

If you want to provide an inventory, and we do recommend providing one, then it should be automatically generated from the best possible information you have, which is usually the file-level license headers.

You may use the SPDX format to convey that inventory, and once you have it, please do use tools like FOSSology or ScanCode to make sure they can read and understand the licenses and copyrights of your software.



Catch the details on
<http://reuse.software/>



If you do all these three things, you'll be REUSE compliant and any tool which tries to read and understand copyrights and licenses in software will thank you for it :)

I'm sure you would like to get your hands on the exact best practices, and you can find them on reuse dot software. At that address, you can also find this presentation as well as links to our Git repositories and other material about REUSE.



Thank you for coming, it's been a pleasure talking to you and I hope to have inspired some thinking in terms of where to go now.
Thank you.