LEIBNIZ INFORMATION CENTRE
FOR SCIENCE AND TECHNOLOGY
UNIVERSITY LIBRARY

TIB

FAIR licences for data & software

Martin Hammitzsch, Angelina Kraft (TIB, 13. July 2018)

Recordings: doi.org/10.5446/37831, /37830 & /37829

FAIR Data & Software (Carpentries-based workshop) #TIBFDS

Leibniz
Association

to be Reusable ♻

R1. (meta)data have a **plurality of accurate and relevant attributes**

- R1.1 (meta)data are released with a clear and accessible data **usage licence**
- R1.2 (meta)data are associated with their **provenance**
- R1.3 (meta)data meet domain-relevant **community standards**

Licensing agenda

1. **Data licensing**

2. Software licensing

Disclaimer: We are not lawyers and nothing in this presentation is intended as legal advice. When in doubt, ask your institution's / employer's / funder's legal counsel.

3. Lunch

4. Wrap-up & discussion

Data

- E.g. **governmental sources**: data are often facts → belong in *Public Domain*:

GOVDATA Das Datenportal für Deutschland · BERLIN OPEN DATA · OffeneDaten.de · data.gov.uk | Find open data · Open Data Hamburg · EUROPEAN DATA PORTAL · UNdata A world of information · *and many more ...*

- **Research Data**: "*As open as possible, as closed as necessary*" (new EU H2020 credo)

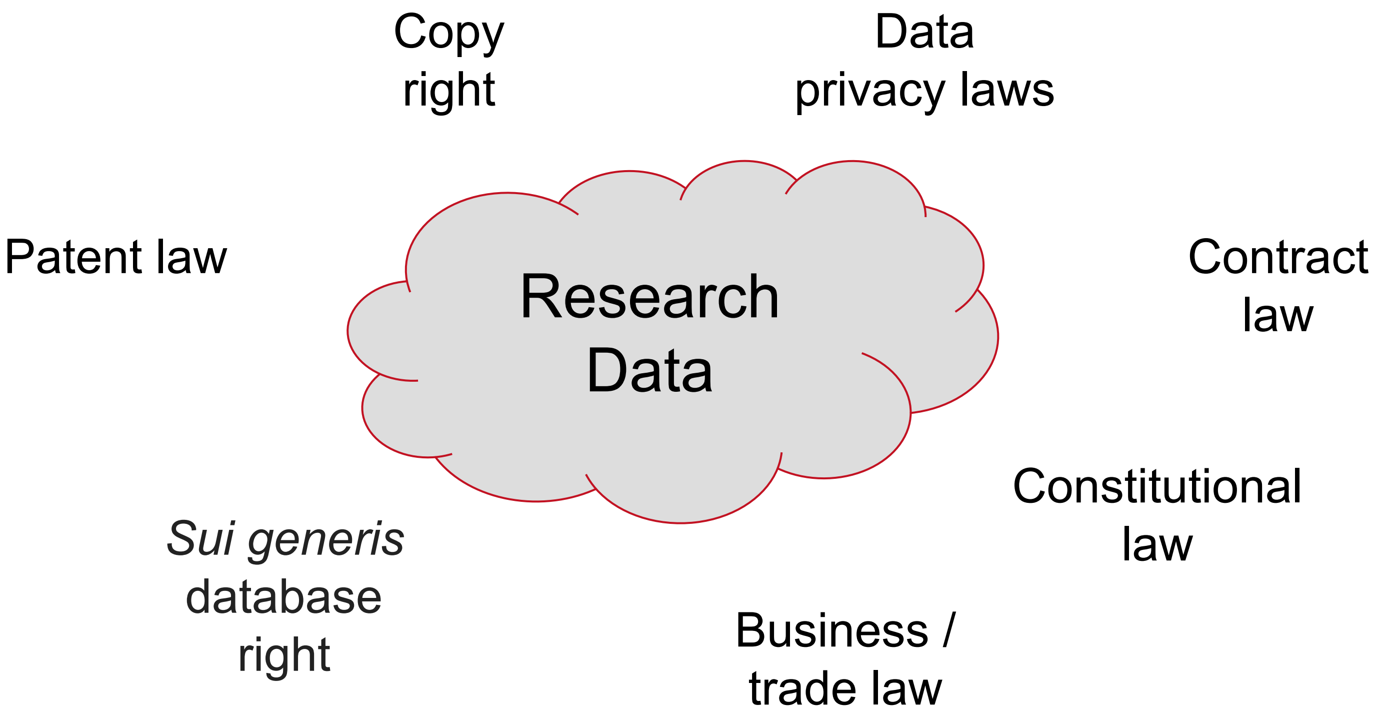
  → there is a shift from 'open data' towards 'FAIR data'

  → **FAIR does not necessarily mean open**

- **Special protection & ethical questions regarding 'sensitive' data & 'mission oriented research'**

- **For data accompanying scientific publications: Creative Commons licences recommended**

(Some) Legal considerations when dealing with research data ...



Copy right

Data privacy laws

Patent law

Research Data

Contract law

Constitutional law

Sui generis database right

Business / trade law

… and more / less depending on the country / federal state you are based in or people you cooperate with …

Data Rights & Policies

→ Depending on country / state / employer and the repository you are using
→ When in doubt, check with your institution's/employer’s/funder’s legal counsel

Software: creative work (mostly)!

⇕

Data: facts, sometimes protected by ‘creativity’ or other regulations (e.g. personal data), sometimes not!

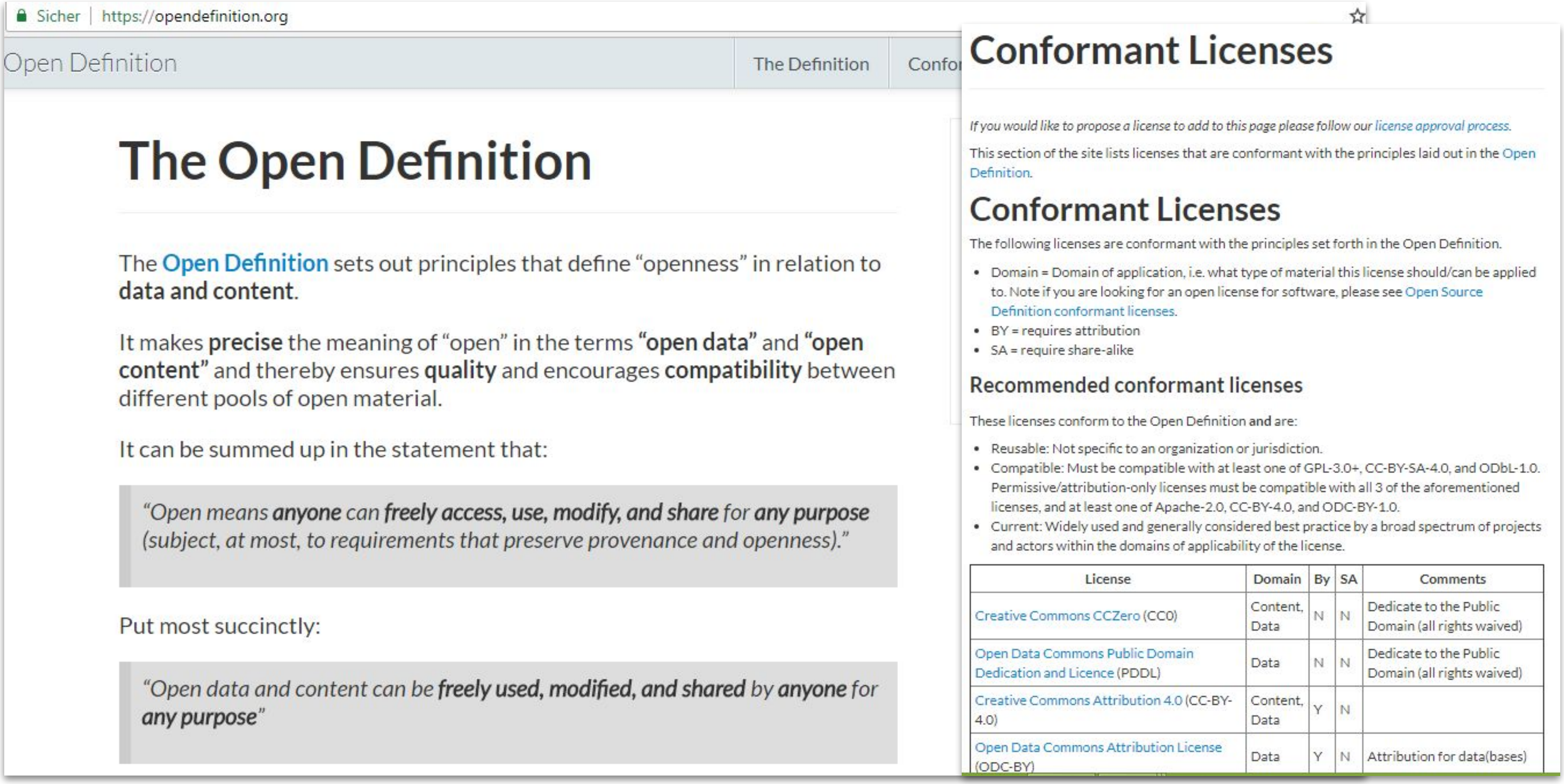
Example Copyright (Germany: UrhG)
→ Quantitative data (with reproduction in mind): usually not protected
→ Qualitative data: capable of being protected (more complex, more probability)

Fact: Most data repositories offer Copyright & Creative Commons licences

Check https://re3data.org - Filter ‘Data licences’
→ What do you find?

Licensing at repository level - Open licenses

Use: → A license grants an exploitation right to a (digital) resource



The Open Definition

The Open Definition sets out principles that define “openness” in relation to **data and content**.

It makes **precise** the meaning of “open” in the terms **“open data”** and **“open content”** and thereby ensures **quality** and encourages **compatibility** between different pools of open material.

It can be summed up in the statement that:

> *“Open means **anyone** can **freely access, use, modify, and share** for **any purpose** (subject, at most, to requirements that preserve provenance and openness).”*

Put most succinctly:

> *“Open data and content can be **freely used, modified, and shared** by **anyone** for **any purpose**”*

Conformant Licenses

If you would like to propose a license to add to this page please follow our license approval process.

This section of the site lists licenses that are conformant with the principles laid out in the Open Definition.

Conformant Licenses

The following licenses are conformant with the principles set forth in the Open Definition.

- Domain = Domain of application, i.e. what type of material this license should/can be applied to. Note if you are looking for an open license for software, please see Open Source Definition conformant licenses.
- BY = requires attribution
- SA = require share-alike

Recommended conformant licenses

These licenses conform to the Open Definition and are:

- Reusable: Not specific to an organization or jurisdiction.
- Compatible: Must be compatible with at least one of GPL-3.0+, CC-BY-SA-4.0, and ODbL-1.0. Permissive/attribution-only licenses must be compatible with all 3 of the aforementioned licenses, and at least one of Apache-2.0, CC-BY-4.0, and ODC-BY-1.0.
- Current: Widely used and generally considered best practice by a broad spectrum of projects and actors within the domains of applicability of the license.

License	Domain	By	SA	Comments
Creative Commons CCZero (CC0)	Content, Data	N	N	Dedicate to the Public Domain (all rights waived)
Open Data Commons Public Domain Dedication and Licence (PDDL)	Data	N	N	Dedicate to the Public Domain (all rights waived)
Creative Commons Attribution 4.0 (CC-BY-4.0)	Content, Data	Y	N	
Open Data Commons Attribution License (ODC-BY)	Data	Y	N	Attribution for data(bases)

List of frequently offered & used (data) licences

licence	Meaning	Abbreviation
Copyright	*German Copyright Code and others (§87a UrhG); **country specific**; re-use for private cases only, no modification, permission needed*	©
Public Domain	*Data is 'free from copyright'*	CC-licence CC0 v1.0 Universal,
Creative Commons	***allows re-use under certain restrictions, e.g. citation and same attributes and/or non-commercial***	**CC BY, CC BY-ND, CC BY-SA, CC BY-NC, CC BY-NC-SA, CC BY-NC-ND**
Open Data Commons	*3 variations:* *Open Data Commons Attribution licence* *Open Data Commons Public Domain Dedication & licence* *Open Data Commons Open Database licence*	ODC ODC-By v1.0 PDDL ODbL

Note: The are many more licences, sometimes country-specific! (e.g. OGLC - Open Government licence Canada)

Other licences in re3data.org refer e.g. to MIT, (new) BSD, or Apache licences … these are primarily software licences and will be covered in the following slides.

Note: Some 'data' repositories also offer 'software' licences, as they treat data as software!

LEIBNIZ INFORMATION CENTRE
FOR SCIENCE AND TECHNOLOGY
UNIVERSITY LIBRARY

TIB

Which questions do you have for us?

Contact information:
Angelina.Kraft@TIB.eu

Leibniz
Association

Licensing agenda

1. Data licensing
2. **Software licensing**

Disclaimer: We are not lawyers and nothing in this presentation is intended as legal advice. When in doubt, ask your institution's / employer's / funder's legal counsel.

3. Lunch
4. Wrap-up & discussion

Why Open Source licenses?

Hey there!

- Which software licenses do you know?
- What are software licenses good for?
- When do licenses matter (use cases)?
- Who benefits from software licenses and how (roles)?
- Who has been licensing software before?

Copyright and licensing

- Copyright
  - Copyright protects the expression of an idea (in source code and object code)
  - Copyright gives the owner certain exclusive rights
    - The right to copy the software
    - The right to create derivative works
    - The right to distribute copies of the software
  - Copyright gives the end user almost no rights
    - Not allowed to copy
    - Not allowed to modify
    - Not allowed to distribute

      … unless the copyright owner says it's OK

- A license is a way for a copyright holder to grant rights to other people
  - Every piece of software you use is covered by a license (hopefully)
  - Your end users are covered by whatever license you place on software you write

Licensing issues

- Development of complex open source solutions is generally done by adapting and integrating multiple existing components
  - The resulting application (or “solution”) may look as a single program from the user point of view, but is in fact a combined work
  - The different components may be covered by different licences
  - Are they compatible and legally interoperable?

- Determining whether the various licences involved are compatible is important when the aim is to redistribute the application to third parties
  - Under what open source licence should it be distributed?
  - Is this always possible?

Two worlds

- Licences for distributing free or open source software (FOSS) are divided in two families

- Copyleft licenses
  - “The world is evil”

- Permissive licenses
  - “The world is good”

Copyleft or “viral” licenses

- Impose the use of the same licence as soon the distributed work is a derivative of the covered work
  - Allow open distribution, modification, and re-use of the code (with attribution)
  - Insist that any derivative works be distributed under the same terms
    - if the original work, even modified or improved, is re-distributed, the same licence must be applied
    - both the original and the new work are Open Source; the copyleft license simply ensures that property is perpetuated to all downstream derivatives
  - Disables proprietary derivatives by third parties (unless the copyright holder gives permission)
    - a majority of open source projects have adopted copyleft licensing terms to avoid software appropriation by third parties
    - the commercial use and derivation by anyone is permitted, as long as the terms of the license are honored
- For example the two GNU GPLs and the EUPL are copyleft

Permissive or "anything goes" licenses

- Is simply a non-copyleft open source license
- Generally compatible and interoperable with most other licences, tolerating to merge, combine or improve the covered code and to re-distribute it under many licences
  - Place very few restrictions on what can be done with the code
  - Permit using the code in non-free or proprietary derivative works
  - Require only attribution in a specified manner
- For example BSD-style, MIT/X11-style, ASLv2 are permissive

Purpose of a licence

- Share openly
    - Practice Open Science and FAIR from start
- Protect and restrict the use
    - Disallow commercialization or any other further use
    - Enable commercialization
- Get credit and acknowledgement
    - Amount of use and citations
- Implement successive sharing
- Refuse warranties
- Refuse liability
- Clarify which license is best for you and other stakeholders
- Deliver a contract with your work

License requirements by FSF

1. The freedom to run the program as you wish, for any purpose (freedom 0).
2. The freedom to study how the program works, and change it so it does your computing as you wish (freedom 1). Access to the source code is a precondition for this.
3. The freedom to redistribute copies so you can help others (freedom 2).
4. The freedom to distribute copies of your modified versions to others (freedom 3). By doing this you can give the whole community a chance to benefit from your changes. Access to the source code is a precondition for this.

→ https://www.gnu.org/philosophy/free-sw.html

License requirements (OSD) by OSI

1. Free redistribution
2. Source code provision
3. Derived works / modifications
4. Integrity of author's source code
5. No discrimination against persons or groups
6. No Discrimination against fields of endeavor
7. Distribution of license
8. License must not be specific to a product
9. License must not restrict other software
10. License must be technology-neutral

→ https://opensource.org/docs/osd

License requirements of the EC

1. Grant all Free (or Open Source) software freedoms;
2. Ensure protection from exclusive software appropriation (therefore be a “share alike” or “copyleft” licence);
3. Have working value in all official EU languages (so there is no need for sworn translators in Court and related institutions for translations);
4. Conform with European copyright law and terminology;
5. Include the “communication to the public” right, including Web distribution / Software as a Service - SaaS;
6. Clarify the applicable law and competent court, as requested by EU institutions;
7. Approach warranties and liability in conformity with “Case law” (a general exclusion of liability is not valid before most European courts);
8. Not be too long, too complex, but be comprehensive and pragmatic.

→ http://dx.doi.org/10.5033/ifosslr.v5i2.91

Content or coverage of licenses - an attempt

- Licenses may cover
  - Definitions
  - Scope
  - Distribution, provision, communication
  - Copyright
  - Obligations (attribution, copyleft, compatibility, provision, legal protection)
  - Authorship
  - Disclaimer (warranty, liability)
  - Additional agreements
  - Acceptance
  - Information
  - Termination
  - Legal issues (jurisdiction, applicable law, miscellaneous)
- Sounds like a contract :-(

Which Open Source licenses?

GNU is not a licence - be specific

- GNU is not UNIX → continue reading at https://en.wikipedia.org/wiki/GNU
  - GNU is an operating system and an extensive collection of computer software
  - GNU is composed wholly of free software most of which is licensed under the GNU Project's own GPL
- GPL, LGPL and AGPL are part of a license family

- Name the licence and its version
- Integrate license text
- Refer to license source

Open Source Initiative (OSI) and others help

- Stewards of the Open Source Definition (OSD)
  - Community-building, education, public advocacy to promote awareness and the importance of non-proprietary software
- Standards body, maintaining the OSD
  - Community-recognized body for reviewing and approving licenses as OSD-conformant (approved licenses)
  - Licenses must go through the Open Source Initiative's license review process to be approved
- OSI Approved License trademark and program creates a nexus of trust
- Approved Licenses (>80)
  - By category → https://opensource.org/licenses/category
  - By name → https://opensource.org/licenses/alphabetical
  - Including other resources, e.g. TLDRLegal, OSSWatch License Diff, Choosealicense, Choosing a license by Civic Commons

Popular (and noteworthy unknown) copyleft licenses

- GNU General Public License (GPL)
  - The copyleft applies to all software based on GPLed code
- GNU Library or "Lesser" General Public License (LGPL)
  - The copyleft applies to any library based on LGPLed code
- Eclipse Public License (EPL)
  - The copyleft applies to any module containing EPLed code
- Mozilla Public License 2.0 (MPL)
  - The copyleft applies to any files containing MPLed code
- Common Development and Distribution License (CDDL)
  - The copyleft applies to source code files containing CDDLed code

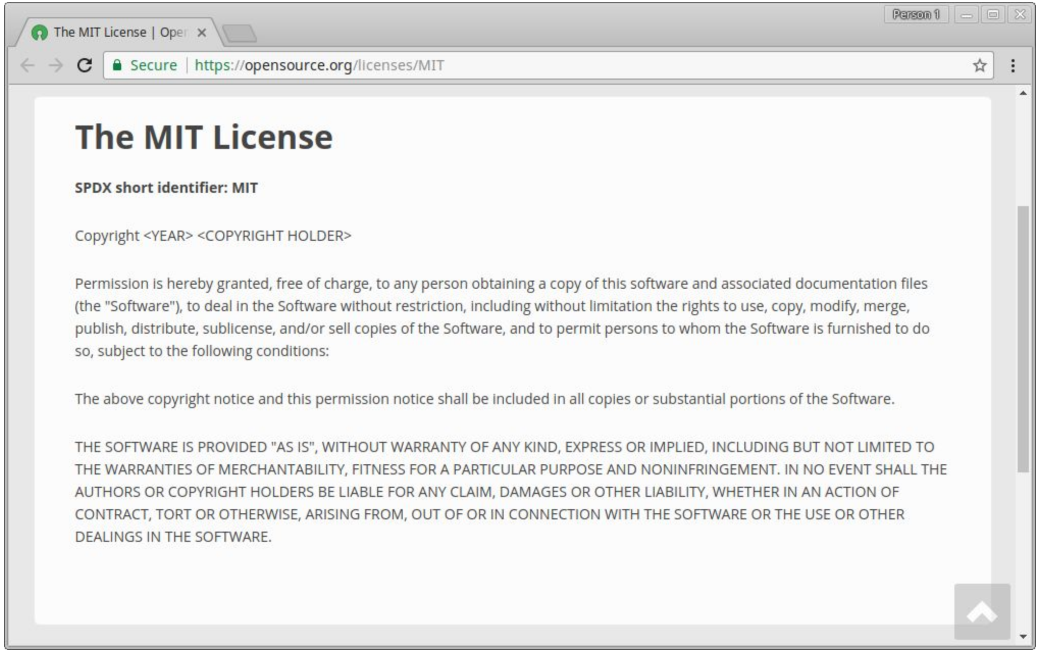
- GNU Affero General Public License (AGPL)
- European Union Public Licence (EUPL)

Popular permissive licenses

- Apache (Software) License 2.0
- BSD 3-Clause "New" or "Revised" license
- BSD 2-Clause "Simplified" or "FreeBSD" license
- MIT license

MIT

- Copyright
- Permission
- Attribution and notice on permissions
- Warranty and Liability disclaimer



The MIT License

SPDX short identifier: MIT

Copyright <YEAR> <COPYRIGHT HOLDER>

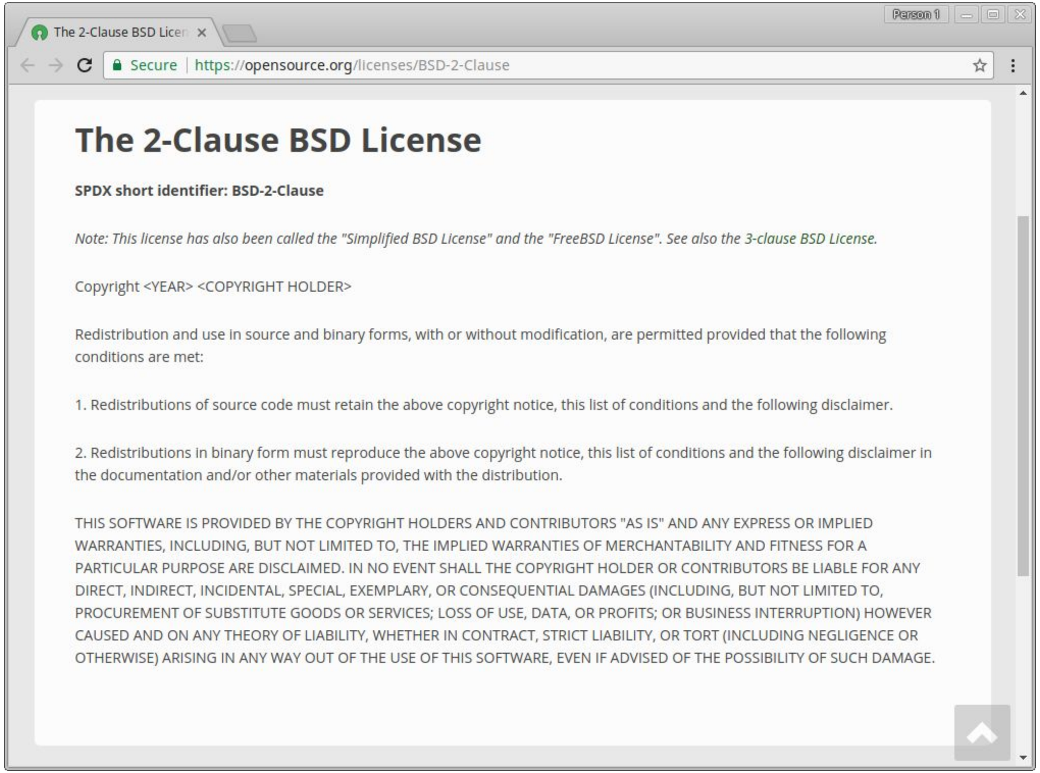
Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

BSD 2-clause

- Copyright
- Permission
- Attribution and notice on conditions and disclaimer
- Warranty and Liability disclaimer



The 2-Clause BSD License

SPDX short identifier: BSD-2-Clause

Note: This license has also been called the "Simplified BSD License" and the "FreeBSD License". See also the 3-clause BSD License.

Copyright <YEAR> <COPYRIGHT HOLDER>

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

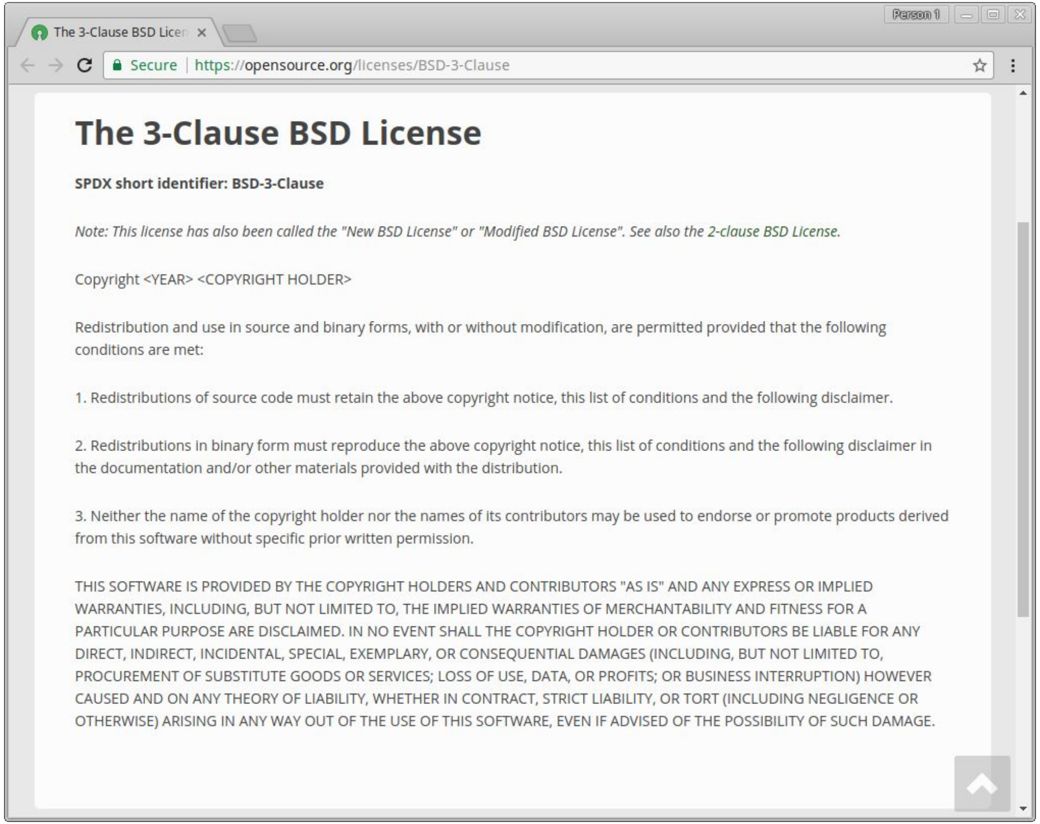
1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

BSD 3-clause

- Copyright
- Permission
- Attribution and notice on conditions and disclaimer
- Promotion
- Warranty and Liability disclaimer



The 3-Clause BSD License

SPDX short identifier: BSD-3-Clause

Note: This license has also been called the "New BSD License" or "Modified BSD License". See also the 2-clause BSD License.

Copyright <YEAR> <COPYRIGHT HOLDER>

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

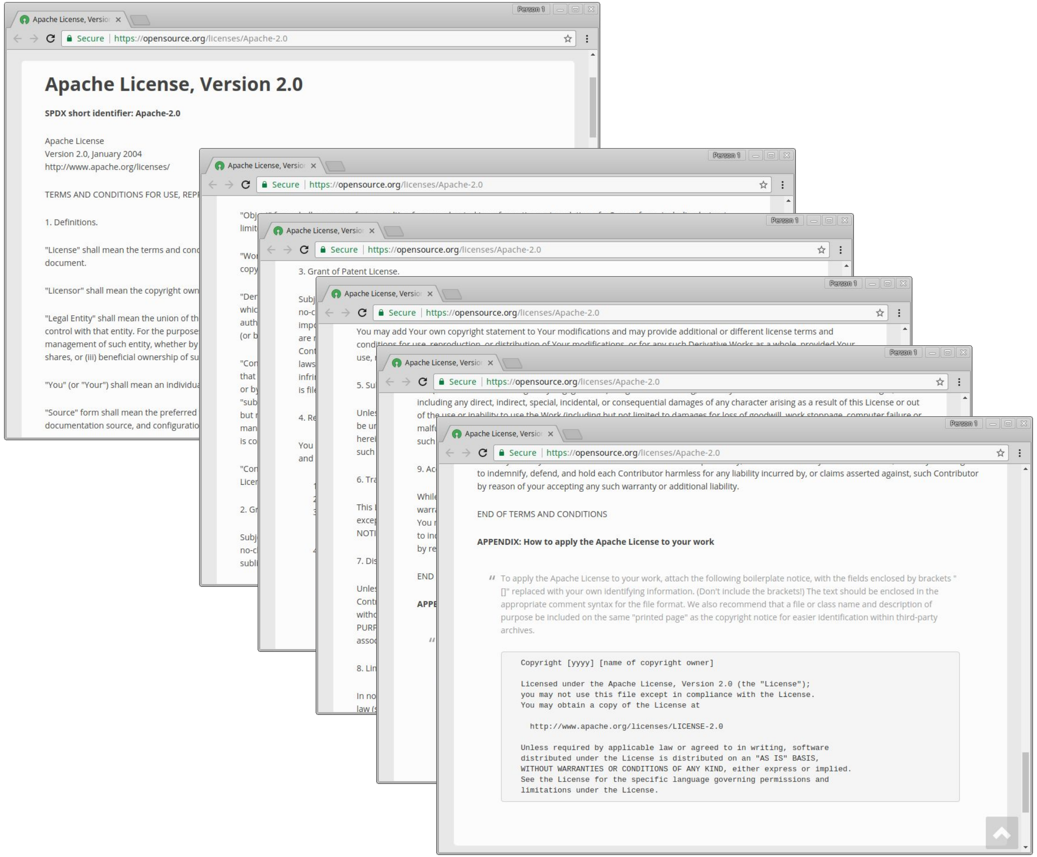
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. Neither the name of the copyright holder nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Apache License version 2.0

- Definitions
- Grant of Copyright License
- Grant of Patent License
- Redistribution
- Submission of Contributions
- Trademarks
- Disclaimer of Warranty
- Limitation of Liability
- Accepting Warranty or Additional Liability
- APPENDIX: How to apply the Apache License to your work



Apache License, Version 2.0

SPDX short identifier: Apache-2.0

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REP

1. Definitions.

"License" shall mean the terms and cond
document.

"Licensor" shall mean the copyright own

"Legal Entity" shall mean the union of th
control with that entity. For the purpose
management of such entity, whether by
shares, or (iii) beneficial ownership of su

"You" (or "Your") shall mean an individua

"Source" form shall mean the preferred
documentation source, and configuratio

3. Grant of Patent License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and

including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out

to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

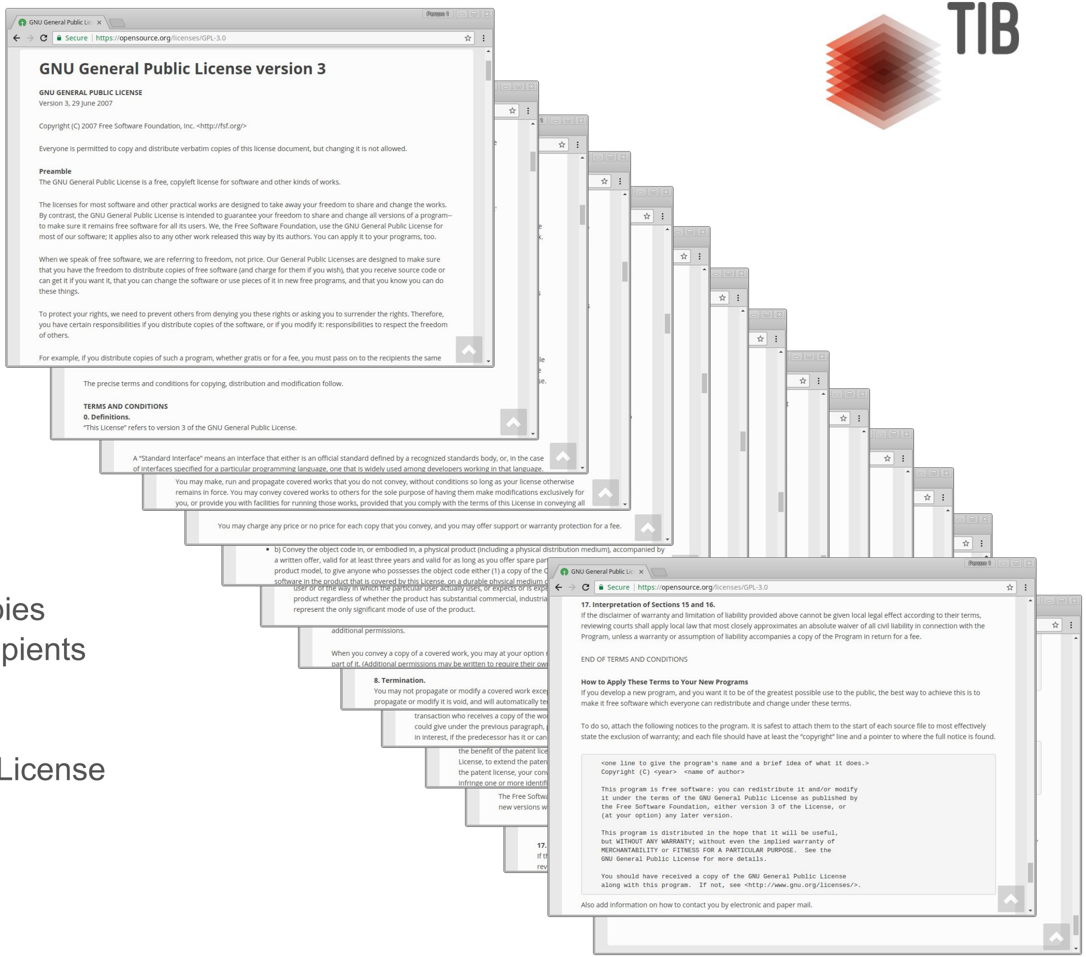
APPENDIX: How to apply the Apache License to your work

> To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

```
Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

  http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.
```

GPLv3

- Preamble
- Definitions
- Source Code
- Basic Permissions
- Protecting Users' Legal Rights From Anti-Circumvention Law
- Conveying Verbatim Copies
- Conveying Modified Source Versions
- Conveying Non-Source Forms
- Additional Terms
- Termination
- Acceptance Not Required for Having Copies
- Automatic Licensing of Downstream Recipients
- Patents
- No Surrender of Others' Freedom
- Use with the GNU Affero General Public License
- Revised Versions of this License
- Disclaimer of Warranty
- Limitation of Liability
- Interpretation of Sections 15 and 16
- How to Apply These Terms to Your New Programs



GNU General Public License version 3

GNU GENERAL PUBLIC LICENSE
Version 3, 29 June 2007

Copyright (C) 2007 Free Software Foundation, Inc. <http://fsf.org/>

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble
The GNU General Public License is a free, copyleft license for software and other kinds of works.

The licenses for most software and other practical works are designed to take away your freedom to share and change the works. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change all versions of a program--to make sure it remains free software for all its users. We, the Free Software Foundation, use the GNU General Public License for most of our software; it applies also to any other work released this way by its authors. You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for them if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs, and that you know you can do these things.

To protect your rights, we need to prevent others from denying you these rights or asking you to surrender the rights. Therefore, you have certain responsibilities if you distribute copies of the software, or if you modify it: responsibilities to respect the freedom of others.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must pass on to the recipients the same

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS
0. Definitions.
"This License" refers to version 3 of the GNU General Public License.

A "Standard Interface" means an interface that either is an official standard defined by a recognized standards body, or, in the case of interfaces specified for a particular programming language, one that is widely used among developers working in that language.

You may make, run and propagate covered works that you do not convey, without conditions so long as your license otherwise remains in force. You may convey covered works to others for the sole purpose of having them make modifications exclusively for you, or provide you with facilities for running those works, provided that you comply with the terms of this License in conveying all

You may charge any price or no price for each copy that you convey, and you may offer support or warranty protection for a fee.

- b) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by a written offer, valid for at least three years and valid for as long as you offer spare par product model, to give anyone who possesses the object code either (1) a copy of the C software in the product that is covered by this License, on a durable physical medium c

user or of the way in which the particular user actually uses, or expects or is expe product regardless of whether the product has substantial commercial, industria represent the only significant mode of use of the product.

additional permissions.

When you convey a copy of a covered work, you may at your option r part of it. (Additional permissions may be written to require their ow

8. Termination.
You may not propagate or modify a covered work exce propagate or modify it is void, and will automatically te

transaction who receives a copy of the wo could give under the previous paragraph, in interest, if the predecessor has it or can

the benefit of the patent lice License, to extend the paten the patent license, your con infringe one or more identifi

The Free Softwa new versions w

17. If th rev

17. Interpretation of Sections 15 and 16.
If the disclaimer of warranty and limitation of liability provided above cannot be given local legal effect according to their terms, reviewing courts shall apply local law that most closely approximates an absolute waiver of all civil liability in connection with the Program, unless a warranty or assumption of liability accompanies a copy of the Program in return for a fee.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs
If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

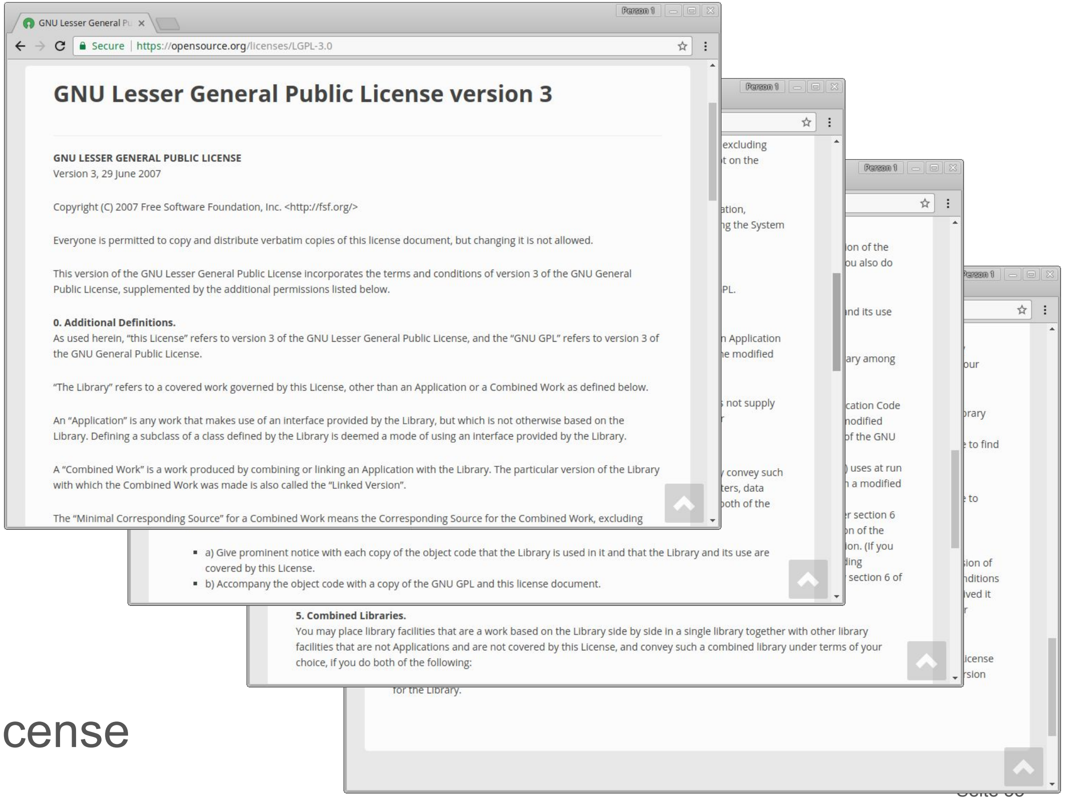
To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively state the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

```
    <one line to give the program's name and a brief idea of what it does.>
    Copyright (C) <year>  <name of author>

    This program is free software: you can redistribute it and/or modify
    it under the terms of the GNU General Public License as published by
    the Free Software Foundation, either version 3 of the License, or
    (at your option) any later version.

    This program is distributed in the hope that it will be useful,
    but WITHOUT ANY WARRANTY; without even the implied warranty of
    MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
    GNU General Public License for more details.

    You should have received a copy of the GNU General Public License
    along with this program.  If not, see <http://www.gnu.org/licenses/>.
```

Also add information on how to contact you by electronic and paper mail.

LGPLv3

- Incorporates the terms and conditions of GPLv3
- Additional Definitions
- Exception to Section 3 of the GNU GPL
- Conveying Modified Versions
- Object Code Incorporating Material from Library Header Files
- Combined Works
- Combined Libraries
- Revised Versions of the GNU Lesser General Public License



GNU Lesser General Public License version 3

GNU LESSER GENERAL PUBLIC LICENSE
Version 3, 29 June 2007

Copyright (C) 2007 Free Software Foundation, Inc. <http://fsf.org/>

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

This version of the GNU Lesser General Public License incorporates the terms and conditions of version 3 of the GNU General Public License, supplemented by the additional permissions listed below.

0. Additional Definitions.
As used herein, "this License" refers to version 3 of the GNU Lesser General Public License, and the "GNU GPL" refers to version 3 of the GNU General Public License.

"The Library" refers to a covered work governed by this License, other than an Application or a Combined Work as defined below.

An "Application" is any work that makes use of an interface provided by the Library, but which is not otherwise based on the Library. Defining a subclass of a class defined by the Library is deemed a mode of using an interface provided by the Library.

A "Combined Work" is a work produced by combining or linking an Application with the Library. The particular version of the Library with which the Combined Work was made is also called the "Linked Version".

The "Minimal Corresponding Source" for a Combined Work means the Corresponding Source for the Combined Work, excluding

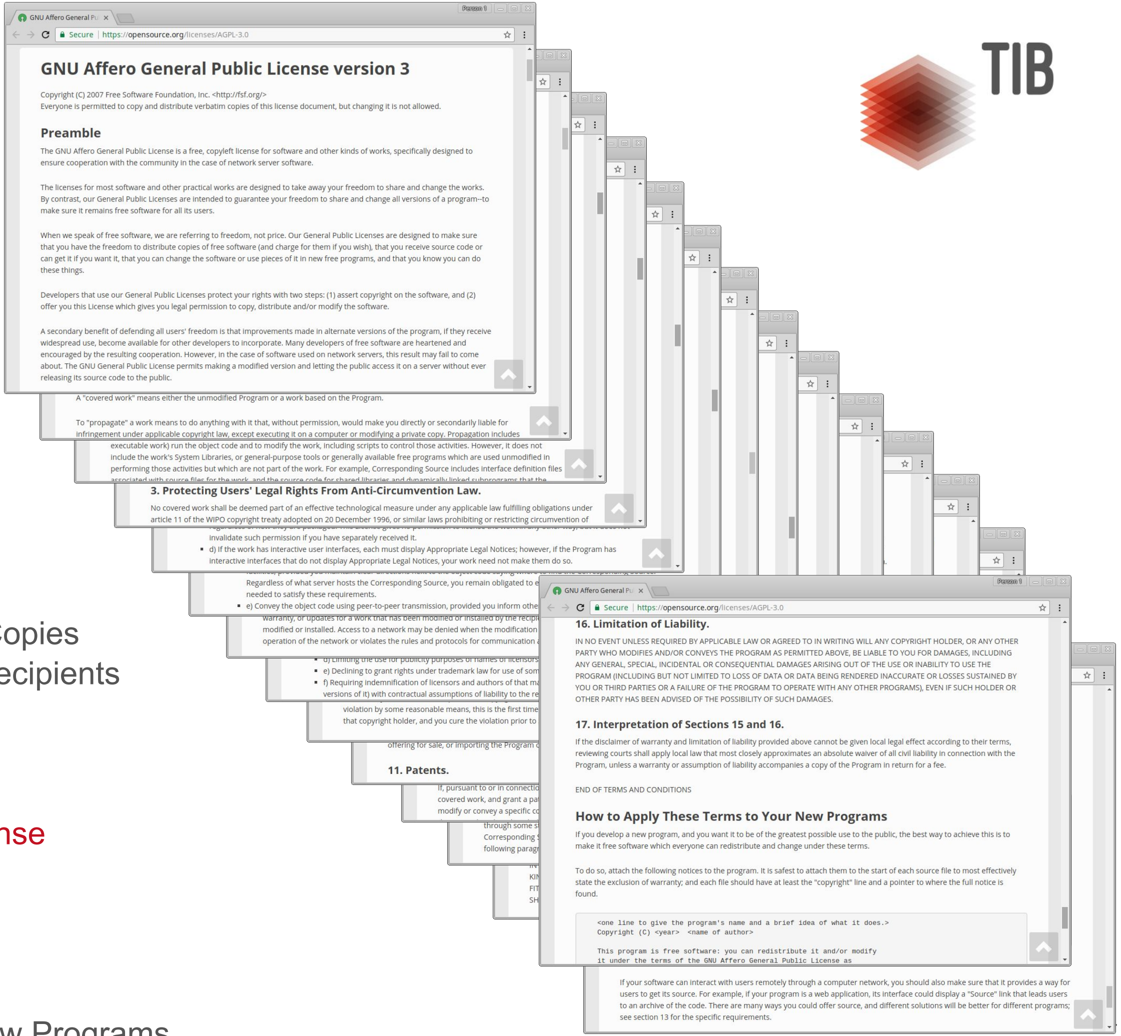
- a) Give prominent notice with each copy of the object code that the Library is used in it and that the Library and its use are covered by this License.
- b) Accompany the object code with a copy of the GNU GPL and this license document.

5. Combined Libraries.
You may place library facilities that are a work based on the Library side by side in a single library together with other library facilities that are not Applications and are not covered by this License, and convey such a combined library under terms of your choice, if you do both of the following:

for the Library.

AGPLv3

- Preamble
- Definitions
- Source Code
- Basic Permissions
- Protecting Users' Legal Rights From Anti-Circumvention Law
- Conveying Verbatim Copies
- Conveying Modified Source Versions
- Conveying Non-Source Forms
- Additional Terms
- Termination
- Acceptance Not Required for Having Copies
- Automatic Licensing of Downstream Recipients
- Patents
- No Surrender of Others' Freedom
- Remote Network Interaction; Use with the GNU General Public License
- Revised Versions of this License
- Disclaimer of Warranty
- Limitation of Liability
- Interpretation of Sections 15 and 16
- How to Apply These Terms to Your New Programs



GNU Affero General Public License version 3

Copyright (C) 2007 Free Software Foundation, Inc. <http://fsf.org/>
Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The GNU Affero General Public License is a free, copyleft license for software and other kinds of works, specifically designed to ensure cooperation with the community in the case of network server software.

The licenses for most software and other practical works are designed to take away your freedom to share and change the works. By contrast, our General Public Licenses are intended to guarantee your freedom to share and change all versions of a program--to make sure it remains free software for all its users.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for them if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs, and that you know you can do these things.

Developers that use our General Public Licenses protect your rights with two steps: (1) assert copyright on the software, and (2) offer you this License which gives you legal permission to copy, distribute and/or modify the software.

A secondary benefit of defending all users' freedom is that improvements made in alternate versions of the program, if they receive widespread use, become available for other developers to incorporate. Many developers of free software are heartened and encouraged by the resulting cooperation. However, in the case of software used on network servers, this result may fail to come about. The GNU General Public License permits making a modified version and letting the public access it on a server without ever releasing its source code to the public.

A "covered work" means either the unmodified Program or a work based on the Program.

To "propagate" a work means to do anything with it that, without permission, would make you directly or secondarily liable for infringement under applicable copyright law, except executing it on a computer or modifying a private copy. Propagation includes

executable work) run the object code and to modify the work, including scripts to control those activities. However, it does not include the work's System Libraries, or general-purpose tools or generally available free programs which are used unmodified in performing those activities but which are not part of the work. For example, Corresponding Source includes interface definition files

3. Protecting Users' Legal Rights From Anti-Circumvention Law.

No covered work shall be deemed part of an effective technological measure under any applicable law fulfilling obligations under article 11 of the WIPO copyright treaty adopted on 20 December 1996, or similar laws prohibiting or restricting circumvention of

invalidate such permission if you have separately received it.

- d) If the work has interactive user interfaces, each must display Appropriate Legal Notices; however, if the Program has interactive interfaces that do not display Appropriate Legal Notices, your work need not make them do so.

Regardless of what server hosts the Corresponding Source, you remain obligated to e needed to satisfy these requirements.

- e) Convey the object code using peer-to-peer transmission, provided you inform othe

modified or installed. Access to a network may be denied when the modification operation of the network or violates the rules and protocols for communication

- e) Declining to grant rights under trademark law for use of som
- f) Requiring indemnification of licensors and authors of that ma versions of it) with contractual assumptions of liability to the re

violation by some reasonable means, this is the first time that copyright holder, and you cure the violation prior to

offering for sale, or importing the Program

11. Patents.

If, pursuant to or in connectio covered work, and grant a pa modify or convey a specific co

through some s Corresponding following parag

KI
FIT
SH

16. Limitation of Liability.

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

17. Interpretation of Sections 15 and 16.

If the disclaimer of warranty and limitation of liability provided above cannot be given local legal effect according to their terms, reviewing courts shall apply local law that most closely approximates an absolute waiver of all civil liability in connection with the Program, unless a warranty or assumption of liability accompanies a copy of the Program in return for a fee.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

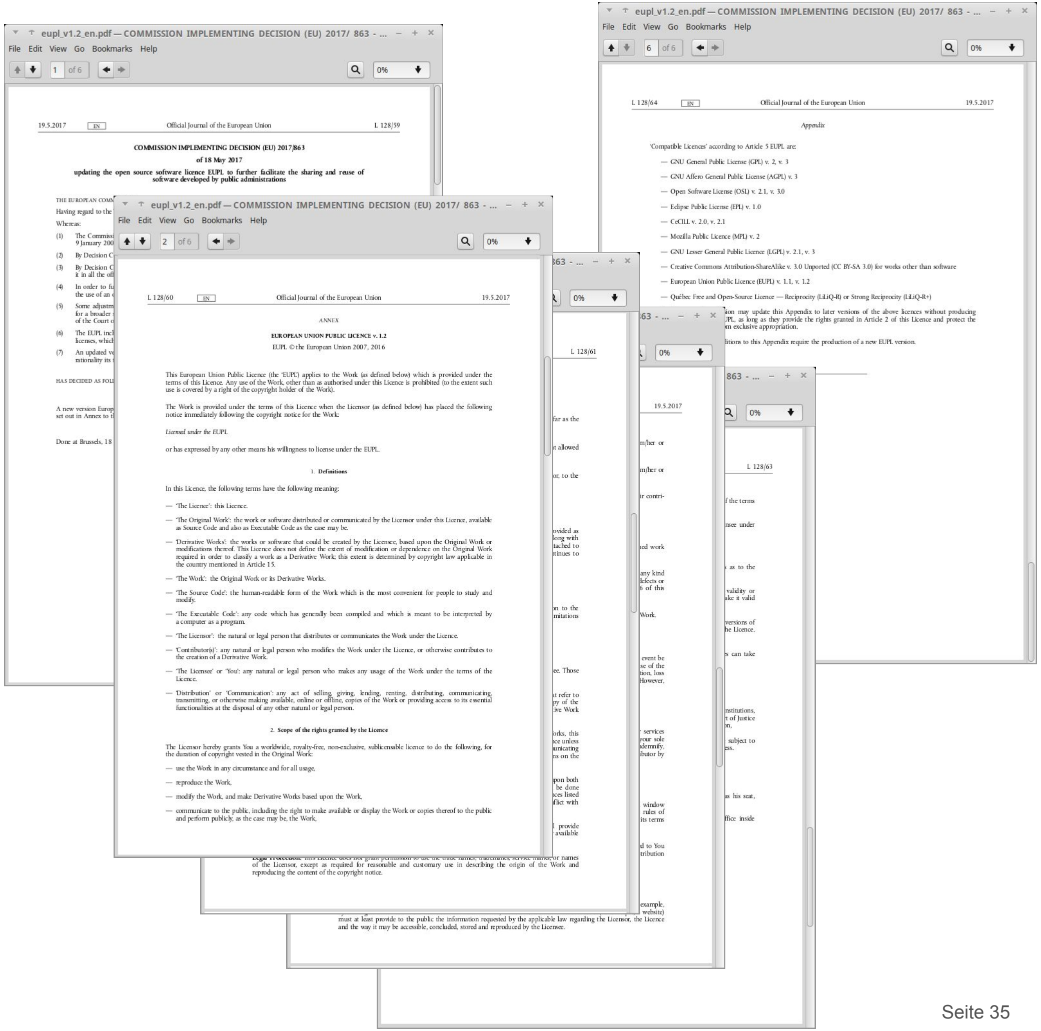
To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively state the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

```
    <one line to give the program's name and a brief idea of what it does.>
    Copyright (C) <year>  <name of author>

    This program is free software: you can redistribute it and/or modify
    it under the terms of the GNU Affero General Public License as
```

If your software can interact with users remotely through a computer network, you should also make sure that it provides a way for users to get its source. For example, if your program is a web application, its interface could display a "Source" link that leads users to an archive of the code. There are many ways you could offer source, and different solutions will be better for different programs; see section 13 for the specific requirements.

EUPLv1.2

- Definitions
- Scope of the rights granted by the Licence
- Communication of the Source Code
- Limitations on copyright
- Obligations of the Licensee
- Chain of Authorship
- Disclaimer of Warranty
- Disclaimer of Liability
- Additional agreements
- Acceptance of the Licence
- Information to the public
- Termination of the Licence
- Miscellaneous
- Jurisdiction
- Applicable Law
- Appendix ‘Compatible Licences’


eupl_v1.2_en.pdf — COMMISSION IMPLEMENTING DECISION (EU) 2017/ 863 - ...
File Edit View Go Bookmarks Help
1 of 6
2 of 6
6 of 6
0%

19.5.2017 EN Official Journal of the European Union L 128/59

COMMISSION IMPLEMENTING DECISION (EU) 2017/863

of 18 May 2017

updating the open source software licence EUPL to further facilitate the sharing and reuse of software developed by public administrations

L 128/60 EN Official Journal of the European Union 19.5.2017

ANNEX

EUROPEAN UNION PUBLIC LICENCE v. 1.2

EUPL © the European Union 2007, 2016

This European Union Public Licence (the ‘EUPL’) applies to the Work (as defined below) which is provided under the terms of this Licence. Any use of the Work, other than as authorised under this Licence is prohibited (to the extent such use is covered by a right of the copyright holder of the Work).

The Work is provided under the terms of this Licence when the Licensor (as defined below) has placed the following notice immediately following the copyright notice for the Work:

Licensed under the EUPL

or has expressed by any other means his willingness to license under the EUPL.

1. **Definitions**

In this Licence, the following terms have the following meaning:

— ‘The Licence’: this Licence.

— ‘The Original Work’: the work or software distributed or communicated by the Licensor under this Licence, available as Source Code and also as Executable Code as the case may be.

— ‘Derivative Works’: the works or software that could be created by the Licensee, based upon the Original Work or modifications thereof. This Licence does not define the extent of modification or dependence on the Original Work required in order to classify a work as a Derivative Work; this extent is determined by copyright law applicable in the country mentioned in Article 15.

— ‘The Work’: the Original Work or its Derivative Works.

— ‘The Source Code’: the human-readable form of the Work which is the most convenient for people to study and modify.

— ‘The Executable Code’: any code which has generally been compiled and which is meant to be interpreted by a computer as a program.

— ‘The Licensor’: the natural or legal person that distributes or communicates the Work under the Licence.

— ‘Contributor(s)’: any natural or legal person who modifies the Work under the Licence, or otherwise contributes to the creation of a Derivative Work.

— ‘The Licensee’ or ‘You’: any natural or legal person who makes any usage of the Work under the terms of the Licence.

— ‘Distribution’ or ‘Communication’: any act of selling, giving, lending, renting, distributing, communicating, transmitting, or otherwise making available, online or offline, copies of the Work or providing access to its essential functionalities at the disposal of any other natural or legal person.

2. **Scope of the rights granted by the Licence**

The Licensor hereby grants You a worldwide, royalty-free, non-exclusive, sublicensable licence to do the following, for the duration of copyright vested in the Original Work:

— use the Work in any circumstance and for all usage,

— reproduce the Work,

— modify the Work, and make Derivative Works based upon the Work,

— communicate to the public, including the right to make available or display the Work or copies thereof to the public and perform publicly, as the case may be, the Work,

of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the copyright notice.

L 128/64 EN Official Journal of the European Union 19.5.2017

Appendix

‘Compatible Licences’ according to Article 5 EUPL are:

— GNU General Public License (GPL) v. 2, v. 3

— GNU Affero General Public License (AGPL) v. 3

— Open Software License (OSL) v. 2.1, v. 3.0

— Eclipse Public License (EPL) v. 1.0

— CeCILL v. 2.0, v. 2.1

— Mozilla Public Licence (MPL) v. 2

— GNU Lesser General Public Licence (LGPL) v. 2.1, v. 3

— Creative Commons Attribution-ShareAlike v. 3.0 Unported (CC BY-SA 3.0) for works other than software

— European Union Public Licence (EUPL) v. 1.1, v. 1.2

— Québec Free and Open-Source Licence — Reciprocity (LiLiQ-R) or Strong Reciprocity (LiLiQ-R+)

ion may update this Appendix to later versions of the above licences without producing UPL, as long as they provide the rights granted in Article 2 of this Licence and protect the m exclusive appropriation.

litions to this Appendix require the production of a new EUPL version.

EUPLv1.2 - uniqueness

- First open source licence to be released by an international governing body
- Legally consistent with the copyright law of all EU countries
  - Considers the specificity and diversity of Member States Law and the Community Law (copyright terminology, information, warranty, liability, applicable law and jurisdiction)
  - Legal instrument, as it has been elaborated in respect of European law requirements and has legal value in 23 European languages, no similar example exists in the world
- Covers data, documents, technical specifications and standards, as well as software source codes (broad coverage)
- Compatible with a wide range of other open source licences, integrates interoperable solutions to possible licence conflicts
  - Ensures downstream compatibility issues with the most relevant other licences, e.g. to merge the work in a larger work with other software components covered by compatible licences (wider compatibility)
  - Creates a new category of F/OSS licence: "Copyleft compatible" (other are: "Strong copyleft", "Weak copyleft" and "Without copyleft" / “Permissive”)
- Allows addition of provisions, as long as these do not contradict the licence itself, e.g. users can add a reference to a specific law that is to be applied, choose a court for arbitration, etc. (more flexibility)

→ https://joinup.ec.europa.eu/collection/eupl/how-use-eupl
→ http://dx.doi.org/10.5033/ifosslr.v5i2.91

What matters?

Licence compatibility

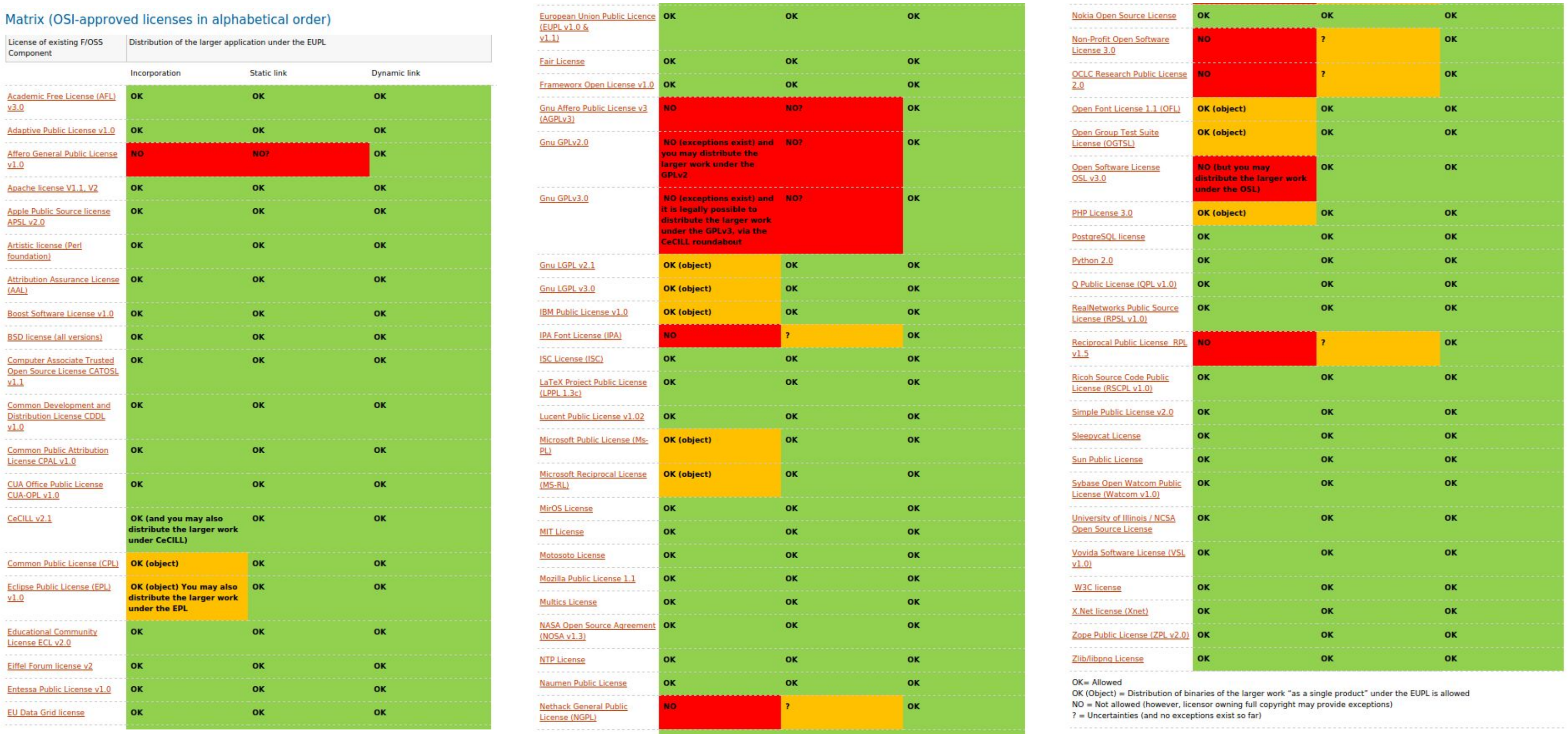
- Licence incompatibility exists when a program is derivative of components licensed under two different copyleft licenses
  - for example, the GPLv2, which is still the most used copyleft licence, is not compatible with GPLv3, and vice-versa
- Compatibility matrices and information exist that help practically, e.g. for
  - GPL: https://www.gnu.org/licenses/license-list.en.html
  - EUPL: https://joinup.ec.europa.eu/collection/eupl/eupl-compatible-open-source-licences
  - BSD: https://en.wikipedia.org/wiki/BSD_licenses

Two ways of licence compatibility

- Upstream compatibility
  - Allows you to merge a work covered by another F/OSS license into a larger work that you may distribute under a license you want to select, e.g. the EUPL
  - This is the main scope of a compatibility matrix, and
  - This is the scope of a licence clause for variable compatibility, e.g in the EUPL

- Downstream compatibility
  - Allows you to merge the work received under a specific license, e.g. the EUPL, into a larger work that you may distribute under a compatible license
  - This is the scope of a licence appendix with a compatibility / exception list

Upstream compatibility for EUPL

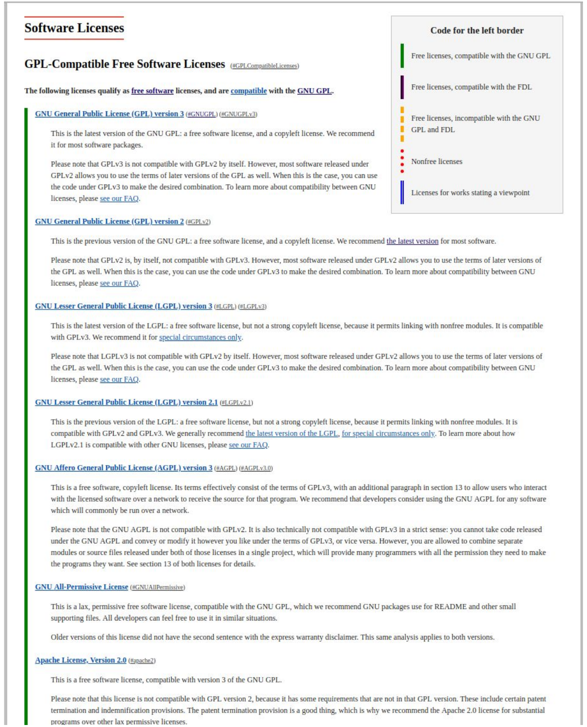


Matrix (OSI-approved licenses in alphabetical order)

License of existing F/OSS Component	Distribution of the larger application under the EUPL		
	Incorporation	Static link	Dynamic link
Academic Free License (AFL) v3.0	OK	OK	OK
Adaptive Public License v1.0	OK	OK	OK
Affero General Public License v1.0	NO	NO?	OK
Apache license V1.1, V2	OK	OK	OK
Apple Public Source license APSL v2.0	OK	OK	OK
Artistic license (Perl foundation)	OK	OK	OK
Attribution Assurance License (AAL)	OK	OK	OK
Boost Software License v1.0	OK	OK	OK
BSD license (all versions)	OK	OK	OK
Computer Associate Trusted Open Source License CATOSL v1.1	OK	OK	OK
Common Development and Distribution License CDDL v1.0	OK	OK	OK
Common Public Attribution License CPAL v1.0	OK	OK	OK
CUA Office Public License CUA-OPL v1.0	OK	OK	OK
CeCILL v2.1	OK (and you may also distribute the larger work under CeCILL)	OK	OK
Common Public License (CPL)	OK (object)	OK	OK
Eclipse Public License (EPL) v1.0	OK (object) You may also distribute the larger work under the EPL	OK	OK
Educational Community License ECL v2.0	OK	OK	OK
Eiffel Forum license v2	OK	OK	OK
Entessa Public License v1.0	OK	OK	OK
EU Data Grid license	OK	OK	OK
European Union Public Licence (EUPL v1.0 & v1.1)	OK	OK	OK
Fair License	OK	OK	OK
Frameworx Open License v1.0	OK	OK	OK
Gnu Affero Public License v3 (AGPLv3)	NO	NO?	OK
Gnu GPLv2.0	NO (exceptions exist) and you may distribute the larger work under the GPLv2	NO?	OK
Gnu GPLv3.0	NO (exceptions exist) and it is legally possible to distribute the larger work under the GPLv3, via the CeCILL roundabout	NO?	OK
Gnu LGPL v2.1	OK (object)	OK	OK
Gnu LGPL v3.0	OK (object)	OK	OK
IBM Public License v1.0	OK (object)	OK	OK
IPA Font License (IPA)	NO	?	OK
ISC License (ISC)	OK	OK	OK
LaTeX Project Public License (LPPL 1.3c)	OK	OK	OK
Lucent Public License v1.02	OK	OK	OK
Microsoft Public License (Ms-PL)	OK (object)	OK	OK
Microsoft Reciprocal License (MS-RL)	OK (object)	OK	OK
MirOS License	OK	OK	OK
MIT License	OK	OK	OK
Motosoto License	OK	OK	OK
Mozilla Public License 1.1	OK	OK	OK
Multics License	OK	OK	OK
NASA Open Source Agreement (NOSA v1.3)	OK	OK	OK
NTP License	OK	OK	OK
Naumen Public License	OK	OK	OK
Nethack General Public License (NGPL)	NO	?	OK
Nokia Open Source License	OK	OK	OK
Non-Profit Open Software License 3.0	NO	?	OK
OCLC Research Public License 2.0	NO	?	OK
Open Font License 1.1 (OFL)	OK (object)	OK	OK
Open Group Test Suite License (OGTSL)	OK (object)	OK	OK
Open Software License OSL v3.0	NO (but you may distribute the larger work under the OSL)	OK	OK
PHP License 3.0	OK (object)	OK	OK
PostgreSQL license	OK	OK	OK
Python 2.0	OK	OK	OK
Q Public License (QPL v1.0)	OK	OK	OK
RealNetworks Public Source License (RPSL v1.0)	OK	OK	OK
Reciprocal Public License RPL v1.5	NO	?	OK
Ricoh Source Code Public License (RSCPL v1.0)	OK	OK	OK
Simple Public License v2.0	OK	OK	OK
Sleepycat License	OK	OK	OK
Sun Public License	OK	OK	OK
Sybase Open Watcom Public License (Watcom v1.0)	OK	OK	OK
University of Illinois / NCSA Open Source License	OK	OK	OK
Vovida Software License (VSL v1.0)	OK	OK	OK
W3C license	OK	OK	OK
X.Net license (Xnet)	OK	OK	OK
Zope Public License (ZPL v2.0)	OK	OK	OK
Zlib/libpng License	OK	OK	OK

OK= Allowed
OK (Object) = Distribution of binaries of the larger work "as a single product" under the EUPL is allowed
NO = Not allowed (however, licensor owning full copyright may provide exceptions)
? = Uncertainties (and no exceptions exist so far)

Upstream compatibility for GPL



Software Licenses

Code for the left border

🟩 Free licenses, compatible with the GNU GPL

🟪 Free licenses, compatible with the FDL

🟧 Free licenses, incompatible with the GNU GPL and FDL

🔴 Nonfree licenses

🟦 Licenses for works stating a viewpoint

GPL-Compatible Free Software Licenses (#GPLCompatibleLicenses)

The following licenses qualify as free software licenses, and are compatible with the GNU GPL.

GNU General Public License (GPL) version 3 (#GNUGPL) (#GNUGPLv3)

This is the latest version of the GNU GPL: a free software license, and a copyleft license. We recommend it for most software packages.

Please note that GPLv3 is not compatible with GPLv2 by itself. However, most software released under GPLv2 allows you to use the terms of later versions of the GPL as well. When this is the case, you can use the code under GPLv3 to make the desired combination. To learn more about compatibility between GNU licenses, please see our FAQ.

GNU General Public License (GPL) version 2 (#GPLv2)

This is the previous version of the GNU GPL: a free software license, and a copyleft license. We recommend the latest version for most software.

Please note that GPLv2 is, by itself, not compatible with GPLv3. However, most software released under GPLv2 allows you to use the terms of later versions of the GPL as well. When this is the case, you can use the code under GPLv3 to make the desired combination. To learn more about compatibility between GNU licenses, please see our FAQ.

GNU Lesser General Public License (LGPL) version 3 (#LGPL) (#LGPLv3)

This is the latest version of the LGPL: a free software license, but not a strong copyleft license, because it permits linking with nonfree modules. It is compatible with GPLv3. We recommend it for special circumstances only.

Please note that LGPLv3 is not compatible with GPLv2 by itself. However, most software released under GPLv2 allows you to use the terms of later versions of the GPL as well. When this is the case, you can use the code under GPLv3 to make the desired combination. To learn more about compatibility between GNU licenses, please see our FAQ.

GNU Lesser General Public License (LGPL) version 2.1 (#LGPLv2.1)

This is the previous version of the LGPL: a free software license, but not a strong copyleft license, because it permits linking with nonfree modules. It is compatible with GPLv2 and GPLv3. We generally recommend the latest version of the LGPL, for special circumstances only. To learn more about how LGPLv2.1 is compatible with other GNU licenses, please see our FAQ.

GNU Affero General Public License (AGPL) version 3 (#AGPL) (#AGPLv3.0)

This is a free software, copyleft license. Its terms effectively consist of the terms of GPLv3, with an additional paragraph in section 13 to allow users who interact with the licensed software over a network to receive the source for that program. We recommend that developers consider using the GNU AGPL for any software which will commonly be run over a network.

Please note that the GNU AGPL is not compatible with GPLv2. It is also technically not compatible with GPLv3 in a strict sense: you cannot take code released under the GNU AGPL and convey or modify it however you like under the terms of GPLv3, or vice versa. However, you are allowed to combine separate modules or source files released under both of those licenses in a single project, which will provide many programmers with all the permission they need to make the programs they want. See section 13 of both licenses for details.

GNU All-Permissive License (#GNUAllPermissive)

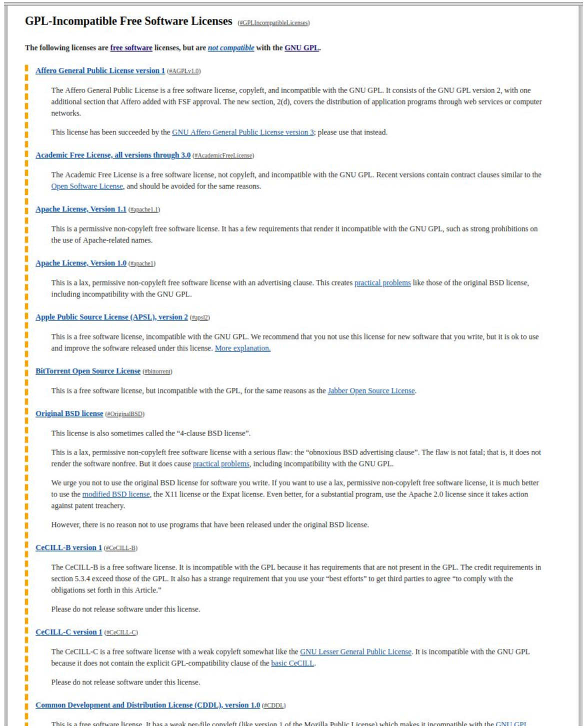
This is a lax, permissive free software license, compatible with the GNU GPL, which we recommend GNU packages use for README and other small supporting files. All developers can feel free to use it in similar situations.

Older versions of this license did not have the second sentence with the express warranty disclaimer. This same analysis applies to both versions.

Apache License, Version 2.0 (#apache2)

This is a free software license, compatible with version 3 of the GNU GPL.

Please note that this license is not compatible with GPL version 2, because it has some requirements that are not in that GPL version. These include certain patent termination and indemnification provisions. The patent termination provision is a good thing, which is why we recommend the Apache 2.0 license for substantial programs over other lax permissive licenses.



GPL-Incompatible Free Software Licenses (#GPLIncompatibleLicenses)

The following licenses are free software licenses, but are *not compatible* with the GNU GPL.

Affero General Public License version 1 (#AGPLv1.0)

The Affero General Public License is a free software license, copyleft, and incompatible with the GNU GPL. It consists of the GNU GPL version 2, with one additional section that Affero added with FSF approval. The new section, 2(d), covers the distribution of application programs through web services or computer networks.

This license has been succeeded by the GNU Affero General Public License version 3; please use that instead.

Academic Free License, all versions through 3.0 (#AcademicFreeLicense)

The Academic Free License is a free software license, not copyleft, and incompatible with the GNU GPL. Recent versions contain contract clauses similar to the Open Software License, and should be avoided for the same reasons.

Apache License, Version 1.1 (#apache1.1)

This is a permissive non-copyleft free software license. It has a few requirements that render it incompatible with the GNU GPL, such as strong prohibitions on the use of Apache-related names.

Apache License, Version 1.0 (#apache1)

This is a lax, permissive non-copyleft free software license with an advertising clause. This creates practical problems like those of the original BSD license, including incompatibility with the GNU GPL.

Apple Public Source License (APSL), version 2 (#apsl2)

This is a free software license, incompatible with the GNU GPL. We recommend that you not use this license for new software that you write, but it is ok to use and improve the software released under this license. More explanation.

BitTorrent Open Source License (#bittorrent)

This is a free software license, but incompatible with the GPL, for the same reasons as the Jabber Open Source License.

Original BSD license (#OriginalBSD)

This license is also sometimes called the “4-clause BSD license”.

This is a lax, permissive non-copyleft free software license with a serious flaw: the “obnoxious BSD advertising clause”. The flaw is not fatal; that is, it does not render the software nonfree. But it does cause practical problems, including incompatibility with the GNU GPL.

We urge you not to use the original BSD license for software you write. If you want to use a lax, permissive non-copyleft free software license, it is much better to use the modified BSD license, the X11 license or the Expat license. Even better, for a substantial program, use the Apache 2.0 license since it takes action against patent treachery.

However, there is no reason not to use programs that have been released under the original BSD license.

CeCILL-B version 1 (#CeCILL-B)

The CeCILL-B is a free software license. It is incompatible with the GPL because it has requirements that are not present in the GPL. The credit requirements in section 5.3.4 exceed those of the GPL. It also has a strange requirement that you use your “best efforts” to get third parties to agree “to comply with the obligations set forth in this Article.”

Please do not release software under this license.

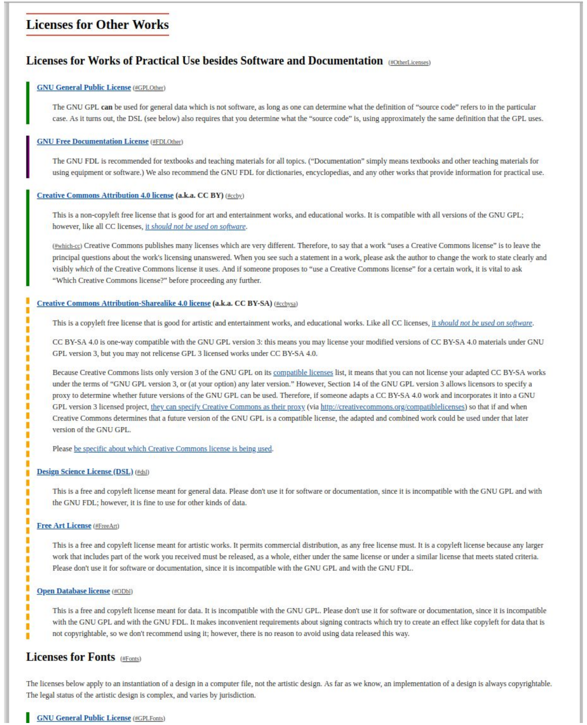
CeCILL-C version 1 (#CeCILL-C)

The CeCILL-C is a free software license with a weak copyleft somewhat like the GNU Lesser General Public License. It is incompatible with the GNU GPL because it does not contain the explicit GPL-compatibility clause of the basic CeCILL.

Please do not release software under this license.

Common Development and Distribution License (CDDL), version 1.0 (#CDDL)

This is a free software license. It has a weak per-file copyleft (like version 1 of the Mozilla Public License) which makes it incompatible with the GNU GPL



Licenses for Other Works

Licenses for Works of Practical Use besides Software and Documentation (#OtherLicenses)

GNU General Public License (#GPLOther)

The GNU GPL **can** be used for general data which is not software, as long as one can determine what the definition of “source code” refers to in the particular case. As it turns out, the DSL (see below) also requires that you determine what the “source code” is, using approximately the same definition that the GPL uses.

GNU Free Documentation License (#FDLOther)

The GNU FDL is recommended for textbooks and teaching materials for all topics. (“Documentation” simply means textbooks and other teaching materials for using equipment or software.) We also recommend the GNU FDL for dictionaries, encyclopedias, and any other works that provide information for practical use.

Creative Commons Attribution 4.0 license **(a.k.a. CC BY)** (#ccby)

This is a non-copyleft free license that is good for art and entertainment works, and educational works. It is compatible with all versions of the GNU GPL; however, like all CC licenses, *it should not be used on software*.

(#which-cc) Creative Commons publishes many licenses which are very different. Therefore, to say that a work “uses a Creative Commons license” is to leave the principal questions about the work's licensing unanswered. When you see such a statement in a work, please ask the author to change the work to state clearly and visibly *which* of the Creative Commons license it uses. And if someone proposes to “use a Creative Commons license” for a certain work, it is vital to ask “Which Creative Commons license?” before proceeding any further.

Creative Commons Attribution-Sharealike 4.0 license **(a.k.a. CC BY-SA)** (#ccbysa)

This is a copyleft free license that is good for artistic and entertainment works, and educational works. Like all CC licenses, *it should not be used on software*.

CC BY-SA 4.0 is one-way compatible with the GNU GPL version 3: this means you may license your modified versions of CC BY-SA 4.0 materials under GNU GPL version 3, but you may not relicense GPL 3 licensed works under CC BY-SA 4.0.

Because Creative Commons lists only version 3 of the GNU GPL on its compatible licenses list, it means that you can not license your adapted CC BY-SA works under the terms of “GNU GPL version 3, or (at your option) any later version.” However, Section 14 of the GNU GPL version 3 allows licensors to specify a proxy to determine whether future versions of the GNU GPL can be used. Therefore, if someone adapts a CC BY-SA 4.0 work and incorporates it into a GNU GPL version 3 licensed project, they can specify Creative Commons as their proxy (via http://creativecommons.org/compatiblelicenses) so that if and when Creative Commons determines that a future version of the GNU GPL is a compatible license, the adapted and combined work could be used under that later version of the GNU GPL.

Please be specific about which Creative Commons license is being used.

Design Science License (DSL) (#dsl)

This is a free and copyleft license meant for general data. Please don't use it for software or documentation, since it is incompatible with the GNU GPL and with the GNU FDL; however, it is fine to use for other kinds of data.

Free Art License (#FreeArt)

This is a free and copyleft license meant for artistic works. It permits commercial distribution, as any free license must. It is a copyleft license because any larger work that includes part of the work you received must be released, as a whole, either under the same license or under a similar license that meets stated criteria. Please don't use it for software or documentation, since it is incompatible with the GNU GPL and with the GNU FDL.

Open Database license (#ODbl)

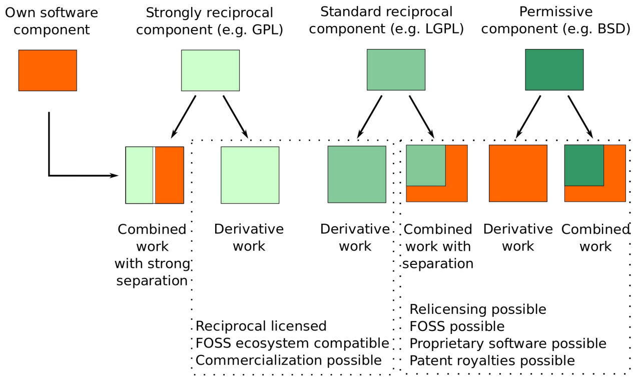
This is a free and copyleft license meant for data. It is incompatible with the GNU GPL. Please don't use it for software or documentation, since it is incompatible with the GNU GPL and with the GNU FDL. It makes inconvenient requirements about signing contracts which try to create an effect like copyleft for data that is not copyrightable, so we don't recommend using it; however, there is no reason to avoid using data released this way.

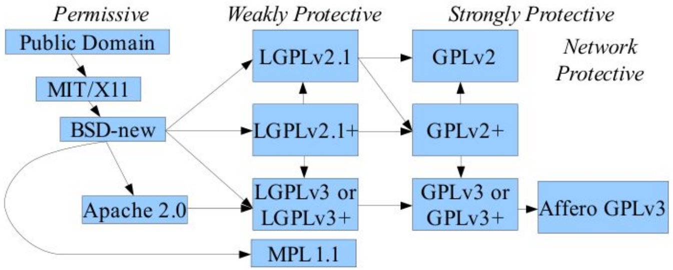
Licenses for Fonts (#Fonts)

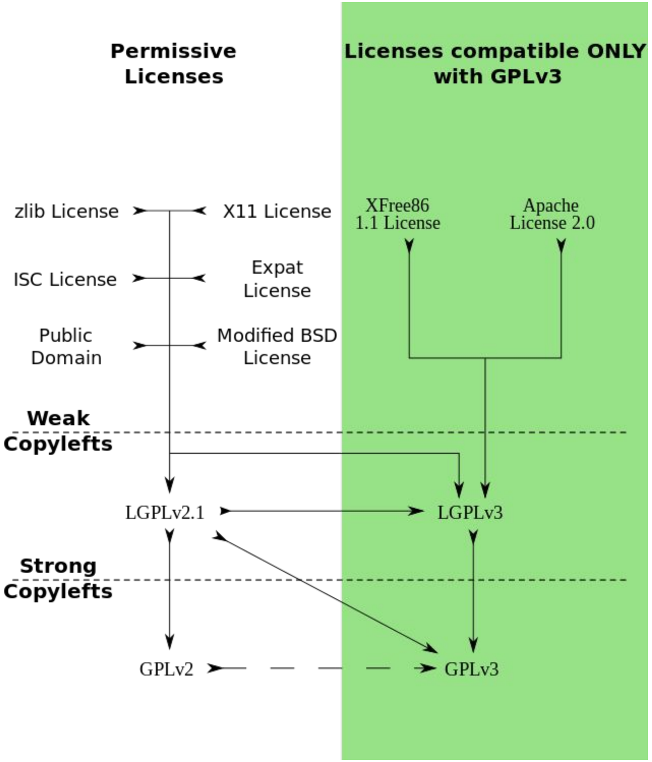
The licenses below apply to an instantiation of a design in a computer file, not the artistic design. As far as we know, an implementation of a design is always copyrightable. The legal status of the artistic design is complex, and varies by jurisdiction.

GNU General Public License (#GPLFonts)

Help for upstream compatibility


Own software component
Strongly reciprocal component (e.g. GPL)
Standard reciprocal component (e.g. LGPL)
Permissive component (e.g. BSD)
Combined work with strong separation
Derivative work
Derivative work
Combined work with separation
Derivative work
Combined work
Reciprocal licensed
FOSS ecosystem compatible
Commercialization possible
Relicensing possible
FOSS possible
Proprietary software possible
Patent royalties possible





→ https://en.wikipedia.org/wiki/License_compatibility

→ https://en.wikipedia.org/wiki/GNU_General_Public_License

Downstream - variable compatibility

- For example EUPL, 5. Obligations of the Licensee
  - Compatibility clause: If the Licensee Distributes or Communicates Derivative Works or copies thereof based upon both the Work and another work licensed under a Compatible Licence, this Distribution or Communication can be done under the terms of this Compatible Licence. For the sake of this clause, ‘Compatible Licence’ refers to the licences listed in the appendix attached to this Licence. Should the Licensee's obligations under the Compatible Licence conflict with his/her obligations under this Licence, the obligations of the Compatible Licence shall prevail.
- Appendix “Compatible Licences” according to article 5 EUPL are:
  - GNU General Public License (GPL) v. 2, v. 3; GNU Affero General Public License (AGPL) v. 3; Open Software License (OSL) v. 2.1, v. 3.0; Eclipse Public License (EPL) v. 1.0; CeCILL v. 2.0, v. 2.1; Mozilla Public Licence (MPL) v. 2; GNU Lesser General Public Licence (LGPL) v. 2.1, v. 3; Creative Commons Attribution-ShareAlike v. 3.0 Unported (CC BY-SA 3.0) for works other than software; European Union Public Licence (EUPL) v. 1.1, v. 1.2; Québec Free and Open-Source Licence; Reciprocity (LiLiQ-R) or Strong Reciprocity (LiLiQ-R+)

Exception list in general

- Alternative way to resolve incompatibility issues without the risk of forking is the constitution of an exception list
  - Advantage is to maintain the licensed component under a single licence including its specific derivatives, while
  - Allowing combined derivative works where the component is integrated or merged to be licensed under an alternate licence
- The difference with the more permissive LGPL system is that exception lists specify which licence(s) are accepted (not “any” licence)

Exception list at license level and/or by licensor

- Exception lists can be implemented at licence level
  - Specifies which licences can be used in case of “combined derivative” when the program is a derivative of both a EUPLed component and another component licensed under a compatible licence
  - The disadvantage of such practice is that it does not facilitate very frequent updates: adapting the list modifies the licence and impact its whole community of users
    - Adapting: In case a compatible licence is updated with a new version number
    - Modifying: Producing a new version that will not be automatically OSI-approved
    - Impact: Some of the community may disagree with the extension
- Exception lists can also (and in addition) be implemented by a specific licensor
  - This is especially recommended when a licensor distributes a library of components under a copyleft licence
  - Similarly, a licensor could distribute a library of components under the EUPL and implement an exception list for some licences that are not in the EUPL compatibility list (for example the GPLv3 in EUPL 1.1)

Dual licensing in the Open Source world

- Multi-licensing is the practice of distributing software under two or more different sets of terms and conditions (Wikipedia)
  - This may mean multiple different software licenses or sets of licenses
  - Prefixes may be used to indicate the number of licenses used, e.g. dual-licensed for software licensed under two different licenses
- A logical incompatibility issue may be resolved through dual licensing
  - The original licensor (owning full copyright) may provide the same program under two or more licences, even if these licenses are not compatible.
- The most frequent cases apply to licensors distributing their work
  - under the “GPL” (without mentioning the version number) or
  - with the comment “either version 2 of the License, or (at your option) any later version”
  - in such case, recipients can use the work under any of these licences (GPLv2 or GPLv3), which is in practice a dual licensing

Dual licensing in the proprietary world

- Copyleft licenses like GPL and AGPL are also sometimes used by vendors as part of a dual licensing business model
  - Whereby they release the code under a copyleft license, but can also sell per-copy exclusive licenses to organizations that want to use or redistribute the software under proprietary terms
- Note that “proprietary” is not the same as “commercial”
  - Commercial use is already permitted (to all parties) by all open source licenses, and this right does not go away in a dual-licensing arrangement
  - Proprietary is when a licensee wishes to redistribute the software (perhaps as part of some larger offering) under non-open-source terms
  - The licensors as the copyright holder are the only ones who could conceivably sue for copyright infringement, and thus they can agree for a fee not to sue.
  - That is what is being sold on the proprietary side of a dual-licensing arrangement: permission to redistribute the software under terms that would otherwise be incompatible with its open source license.

License provisions

- There are questions regarding software licenses in general when scientific results - such as figures or numbers - produced by software are used to publish findings in papers or scientific journals
  - It would be highly beneficial if a software license would cover this case so that the source code has to be provided for any modified version of a software that is not distributed but that is used to produce results which are presented in a scientific paper
  - AFAIK, no well-known software license is covering this case so far

- Is it possible to release software under a license so that the source code has to be provided when modified versions are used by others to produce results and findings which are then presented in scientific papers?

License provisions - example

```
/*
 * EasyWave - A realtime tsunami simulation program with GPU support.
 * Copyright (C) 2014  Andrey Babeyko, Johannes Spazier
 * GFZ German Research Centre for Geosciences (http://www.gfz-potsdam.de)
 *
 * Parts of this program (especially the GPU extension) were developed
 * within the context of the following publicly funded project:
 * - TRIDEC, EU 7th Framework Programme, Grant Agreement 258723
 *   (http://www.tridec-online.eu)
 *
 * Licensed under the EUPL, Version 1.1 or - as soon they will be approved by
 * the European Commission - subsequent versions of the EUPL (the "Licence"),
 * complemented with the following provision: For the scientific transparency
 * and verification of results obtained and communicated to the public after
 * using a modified version of the work, You (as the recipient of the source
 * code and author of this modified version, used to produce the published
 * results in scientific communications) commit to make this modified source
 * code available in a repository that is easily and freely accessible for a
 * duration of five years after the communication of the obtained results.
 *
 * You may not use this work except in compliance with the Licence.
 *
 * You may obtain a copy of the Licence at:
 * https://joinup.ec.europa.eu/software/page/eupl
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the Licence is distributed on an "AS IS" basis,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the Licence for the specific language governing permissions and
 * limitations under the Licence.
 */
```

Copyright

Project

Additional Provision

License

Flexibility of licenses and own additions

- Flexibility integrated in licenses is a game changer
  - Resolving incompatibility is easier
- Dual licensing
  - “... or any later version”, e.g. for GPL and EUPL
- Exception list
  - Appendix and clauses enabling compatible licenses for legal interoperability when licenses are incompatible and the other licenses follow the same spirit
- Special cases
  - Larger Work including another, existing GPL work, e.g. for MPLv2

- Own additions and extensions may result in a new license
  - Incompatibility and practical issues for users
  - Be careful and make changes only following the license

Patents

- Copyright protection is only provided to expressions and exempt to ideas, procedures or operational/computing methods

- Patents may cover ideas, procedures and operational methods

Patents

- Practice
  - Have been granted that cover functionality in many common software applications
  - Pose a specific risk to the development of software because they protect the idea or the method and not simply the form, as copyright does
  - It is possible to infringe some patent without copying anything or without even knowing it
- Impact
  - Users are rarely subject to legal action for patent infringement
  - Developers and software licensors must almost always be notified first, prior to legal action seeking damages, thus giving them an opportunity to replace or remove the functionality that is possibly patented
- Risk
  - Related to higher costs for replacing functionality, rather than the direct costs of being subjected to patent lawsuits
  - Appear to be higher for proprietary software, where commercial interests are higher, than for Open Source distribution

Software patents

- A software patent is a patent on a piece of software, such as a computer program, libraries, user interface, or algorithm (Wikipedia)
- There is no legal or conclusive definition for a software patent
- Different countries have different restrictions on patenting software innovations
- Examples are MP3 audio or the LZW data compression
- Known software patents have halted software development specific areas
- No Free / Open Source software has yet been subject to legal action for patent infringement

Software patents and software licenses

- Grant liberal copyright and patent licenses allowing for free use, modification, distribution, and exploitation of the work
- A patent retaliation clause protects an auxiliary distributor's further recipients against patent trolling
- Covered in licenses such as
  - Apache License version 2
  - Eclipse Public License
  - Mozilla Public License version 2 (MPLv2)
  - Common Development and Distribution License (CDDL)

Termination of license and law enforcement

- More or less individual case by case decisions, a few or no general court rulings?
- Financial resources, language and legal framework are issues
- Financial and ideological interests as well as principles and impact matter
- Individual actions are problematic
  - Purpose of law enforcement and impact
  - Lack of financial resources and/or power
  - Business model
- Foundations may help
  - Initiate friendly contact and continue with contact on pain of penalties as well as law enforcement (financial resources and)
- Translations and specific license articles may help, e.g. EUPL within Europe

Licenses and coding?

How does coding happen

- Normally you “just” start with 10 lines of code
    - You “just” start sharing the code or script with a colleague
    - Then the code base may grow slightly, and from time to time because you or your colleagues find it useful and have further requirements and wishes
    - Either it grows just by your work or others contribute somehow
    - You end up with much more than initially intended or foreseen
- Normally you “just” contribute with
    - Code snippets send by mail (yes, this happens - quite often)
    - Pull requests
- Seldom you start a complex software project on purpose
    - Either based on existing work to create a larger work
    - Or from scratch with greenfield approach
    - It does not “just” happen

Practices

- Practice Open Science, be open from the beginning
  - Select and attach a license and include licence headers
- Wait and think a bit, but not too long
  - Include a boilerplate for “All rights reserved” as a placeholder for licence headers
  - Then decide having in mind a specific purpose
  - Select and attach a license and include licence headers
- Postpone
  - Clarify use of libraries and their licenses
  - Check licenses under pressure and in a hurry
  - Do in a relatively short time what could have been done over the time
- ...

Dependencies

- Consider dependencies
  - Libraries used and their dependencies
  - Contract of employment or similar agreements
  - Contracts and agreements as part of the works, e.g. partnerships in research projects
    - Consortium Agreement regulating IPR issues
    - Memorandum of Understanding
  - Cooperation on working level
- Resolve dependencies
  - Check compatibility
  - Get in touch and talk!
  - Select and attach a license
  - Establish Contributor License Agreements (CLAs) and Developer Certificate of Origin (DCO) from the beginning, e.g. Eclipse Foundation
    → https://www.eclipse.org/legal/ECA.php
    → https://www.eclipse.org/legal/DCO.php
  - Re-implement problematic libraries

Did we ask: Licenses and coding?

- What about your programming languages?
- Python
  - Python Software Foundation (PSF) license (PSFL) → https://docs.python.org/3/license.html
    see also the licenses and acknowledgements for incorporated software
  - Non-reusable licenses* → https://opensource.org/licenses/category
    * Licenses in this group are specific to their authors and cannot be reused by others
    → https://opensource.org/proliferation-report
- R, CRAN
  - GPL and BSD license families
    → https://www.r-project.org/Licenses/ for R and https://cran.r-project.org/web/licenses/ for CRAN
- Java
  - Legal case within the United States “Oracle America, Inc. v. Google, Inc.”
    US$8.8 billion in damages due to the commercial success of the Android system
    → https://en.wikipedia.org/wiki/Oracle_America,_Inc._v._Google,_Inc.
  - It’s about copyrightability of APIs, fair use, and impact on open mimics of existing APIs
    - Linux is fully open sourced, it is based on POSIX, a set of APIs that mimic those of the commercial Unix operating system
    - Storage technology company SwiftStack uses the APIs from Amazon's various cloud services to ensure compatibility
    - ...

Best practices for license handling

Best practices for license headers

- Ensure corporate identity
  - Add copyright information with names of institutes and authors
  - Add name and description of the software
- Provide additions, e.g.
  - Information on funding and/or projects
  - URLs for further information
- Provide license information or “All rights reserved” information until license decision (e.g. → Adobe)

- Provide what matters but keep it short and consistent

```
/*
 * EasyWave - A realtime tsunami simulation program with GPU support.
 * Copyright (C) 2014  Andrey Babeyko, Johannes Spazier
 * GFZ German Research Centre for Geosciences (http://www.gfz-potsdam.de)
 *
 * Parts of this program (especially the GPU extension) were developed
 * within the context of the following publicly funded project:
 * - TRIDEC, EU 7th Framework Programme, Grant Agreement 258723
 *   (http://www.tridec-online.eu)
 *
 * Licensed under the EUPL, Version 1.1 or - as soon they will be approved by
 * the European Commission - subsequent versions of the EUPL (the "Licence"),
 * complemented with the following provision: For the scientific transparency
 * and verification of results obtained and communicated to the public after
 * using a modified version of the work, You (as the recipient of the source
 * code and author of this modified version, used to produce the published
 * results in scientific communications) commit to make this modified source
 * code available in a repository that is easily and freely accessible for a
 * duration of five years after the communication of the obtained results.
 *
 * You may not use this work except in compliance with the Licence.
 *
 * You may obtain a copy of the Licence at:
 * https://joinup.ec.europa.eu/software/page/eupl
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the Licence is distributed on an "AS IS" basis,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the Licence for the specific language governing permissions and
 * limitations under the Licence.
 */
```

Best practices for license handling in SW packages

- LICENCE and README files
  - Git clone and local copies vs. GitHub
- doc/licences/
  - Include licenses.{md|txt|html} listing libraries used and their licenses (eventually with URLs)
  - Include copies of license texts
- Ensure attribution
  - Name the authors and institutes of libraries etc.
- Include copyright and license header in code files

Best practices for copy-and-pasted code snippets

- Clarify use for copy and paste snippets
- Forums, blogs and websites
  - Stackoverflow forces CC-BY-SA for contributions → https://stackoverflow.com/help/licensing
  - Blogs and websites do not talk about licensing most likely: that’s an issue
    - Search for information
    - Ask for permission to use it with an specific licence
    - Create awareness
- Trace quick and dirty solutions (in terms of legal aspects) with comments in the source code
  - URL of the source, and
  - License information and attribution, if available or
  - FIXME, TODO, CHECK … whatever you want to indicate that this part needs a license check (be consistent to find it again quickly when it’s time to talk about licenses)

Best practices for code contributions

- Issues
  - How does a project get the legal right to redistribute some contributor's code changes?
  - How does a project know the contributor won't sue later for copyright infringement?
- Legally accept code and documentation contributions
  - License header is integrated and follows the licensing of the larger work → works for files
  - Obtain Contributor License Agreement (CLA) → works for snippets, patches, pull requests
- With a CLA a contributor
  - Gives the project rights formally, by signing an agreement saying that the project can redistribute the contributed code
  - Retains copyright, but promises not to exercise most of the powers that copyright ownership would ordinarily imply
  - Enables the project to feel safe in accepting the contribution
  - Has to do this once per project
- CLA covers not only the current contribution but all subsequent ones, so the form is only requested with someone's first contribution
  - Examples: http://wiki.civiccommons.org/Contributor_Agreements/

Software publications

Hey there!

- What is a software publication?
- Is there anyone who has published software before?
- Which distribution channel has been used or which do you know?
- Why do software publications matter?
- Who benefits from software publications?

Why opening software and source code matters?

- Ensure the sustainability and verifiability of research software development as part of research processes;
- Improve the perception of the role of software in research as the foundation of publications and research data, as well as the link between them;
- Open up of research software as a key building block in Open Science;
- Link software publications to the scientific reputation system through consistent application of DOIs, referencing, metrics and reviews;
- Support the professionalization of software development in research through trainings, and the application of minimally necessary best practices from software engineering;
- Establish research software engineering as an essential core competence, especially among young scientists, the next generation of researchers;
- Integrate research software engineering with higher-level activities, e.g., in the context of e-Science and e-Infrastructures

→ http://www.de-rse.org/en/aims.html

Why opening software and source code matters?

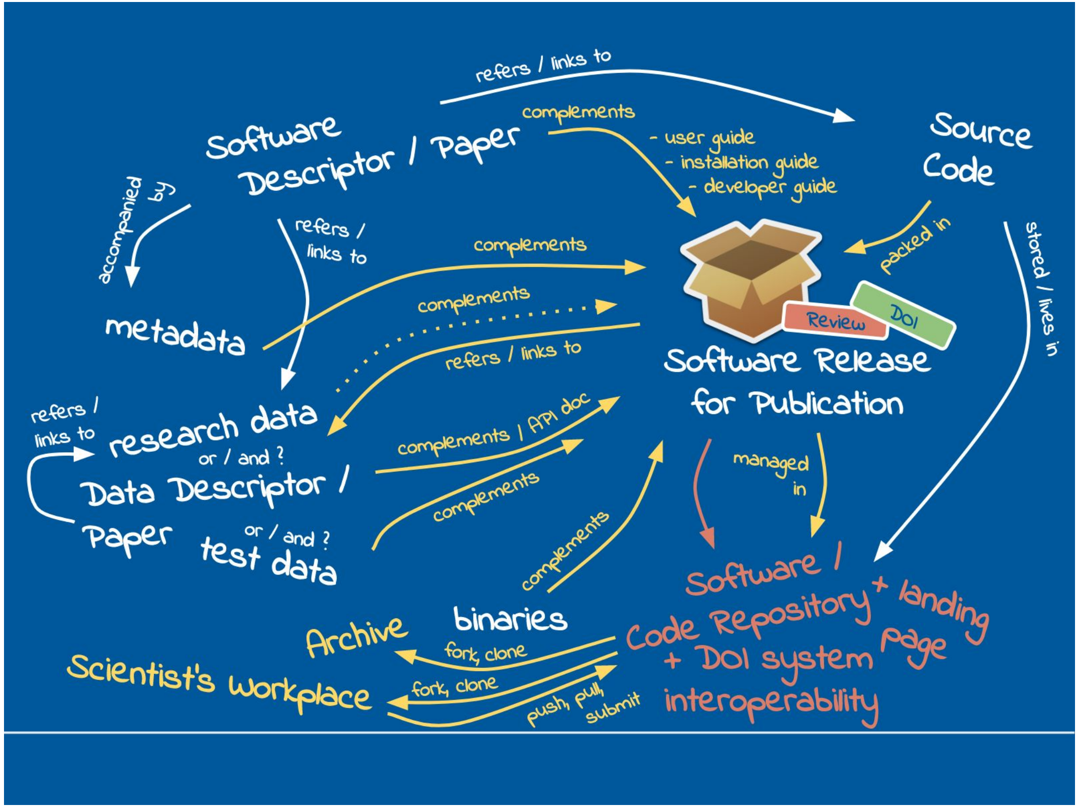
Until there is a radical change in the way that academic credit is given, the principal record of scientific research is still the peer-reviewed publication. Given that software is a fundamental part of doing science in the digital age, the question we are often asked is: where can I publish papers which are primarily focused on my scientific software?

-- Neil Chue Hong

Software journals and repositories

- Software journals
  - Use individual policies for software related papers
  - A comprehensive list can be found at https://www.software.ac.uk/resources/guides/which-journals-should-i-publish-my-software
- Digital repositories
  - Mint DOIs for source code copies and software release packages
  - Examples are Zenodo, NanoHub
- Software and source code repositories
  - Provide environment for code and software project management, collaboration tools and CI
  - Examples are Git, GitHub, GitLab, BitBucket
- Foundations
  - Provide environment for FOSS projects and apply defined processes for defined criteria/goals
  - Examples are the Apache Foundation, Eclipse Foundation
- Institutes
  - Offer software/code repositories and digital repositories for research results and development

Publications related to software - an attempt



Dive into software publications

- DataCite Search lists ~60.000 (~30.000?) software publication
  → https://search.datacite.org/works?resource-type-id=software
  - Zenodo ~50.000 software publications
    → https://search.datacite.org/works?resource-type-id=software&data-center-id=cern.zenodo

    vs. ~20.000 with Zenodo Search → https://zenodo.org/search?type=software
  - FigShare ~4700, Bioconductor ~2900 etc.

Software citation

- DataCite offers citation snippets for copy and paste after DOI minting
- Software citation starts at your screen before or without minting a DOI

- Force proper citation
  - Follow FORCE11 recommendations for software
  - Follow community practices

- Software vs. data in the context of citation
  - Software is data, but it is not just data
  - Data (in computing and information science): anything that can be processed by a computer
  - Software: special kind of data that can be a creative, executable tool that operates on data
  - Software & data are similar in with regard to credit and metrics, and both traditionally have not been cited in publications

    → https://os.helmholtz.de/fileadmin/user_upload/os.helmholtz.de/Workshops/helmholtz_oswebinar42_katz.pdf

Citation - FORCE11 recommendations for software

- FORCE11 is a community to help facilitate the change toward improved knowledge creation and sharing

- In 2016 the Software Citation Working Group produced a consolidated set of citation principles in order to encourage broad adoption of a consistent policy for software citation across disciplines and venues

- “Software citation principles” → https://doi.org/10.7717/peerj-cs.86 contains
    - Principles (general statements)
    - Use cases (where the principles should apply)
    - Discussion (suggestions on how to apply principles)

Citation - Principles 1

- **Importance**: Software should be considered a legitimate and citable product of research. Software citations should be accorded the same importance in the scholarly record as citations of other research products, such as publications and data; they should be included in the metadata of the citing work, for example in the reference list of a journal article, and should not be omitted or separated. Software should be cited on the same basis as any other research product such as a paper or a book, that is, authors should cite the appropriate set of software products just as they cite the appropriate set of papers.

Citation - Principle 2 & 3

- **Credit and attribution**: Software citations should facilitate giving scholarly credit and normative, legal attribution to all contributors to the software, recognizing that a single style or mechanism of attribution may not be applicable to all software.

- **Unique identification**: A software citation should include a method for identification that is machine actionable, globally unique, interoperable, and recognized by at least a community of the corresponding domain experts, and preferably by general public researchers.

Citation - Principle 4, 5 & 6

- **Persistence**: Unique identifiers and metadata describing the software and its disposition should persist - even beyond the lifespan of the software they describe.

- **Accessibility**: Software citations should facilitate access to the software itself and to its associated metadata, documentation, data, and other materials necessary for both humans and machines to make informed use of the referenced software.

- **Specificity**: Software citations should facilitate identification of, and access to, the specific version of software that was used. Software identification should be as specific as necessary, such as using version numbers, revision numbers, or variants such as platforms.

CITATION file and CITATION.cff ...

- Put a plaintext file named CITATION in the root directory of your code
  - Put (human-readable) information in it about how to cite your software
    → https://peerj.com/articles/cs-86/#p-136
    → https://www.software.ac.uk/blog/2013-09-02-encouraging-citation-software-introducing-citation-files
  - Examples → https://github.com/EIDA/webdc3, https://github.com/swcarpentry/2014-01-18-ucb/
  - More examples → https://github.com/search?q=filename:CITATION

- CITATION.cff (Citation File Format)
  - CFF is a human- and machine-readable format for CITATION files serialized in YAML
    → https://citation-file-format.github.io/, https://doi.org/10.5281/zenodo.1242911
  - Example → https://github.com/citation-file-format/cff-converter-python
  - More examples → https://github.com/search?q=filename:CITATION.cff

Other attempts: codemeta.json, README.md ...

- JSON-LD / CodeMeta
  - Provides a way of describing machine-readable information with semantic context
    → http://doi.org/10.5334/jors.by, https://codemeta.github.io/terms/
  - Examples
    → https://github.com/codemeta/codemeta/blob/master/codemeta.json
    → https://github.com/citation-file-format/cff-converter-python/
  - More examples → https://github.com/search?q=filename:codemeta.json
- README.md
  - Include “How to cite” or “To cite in publications, please use” etc.
    followed with your recommendation for citation
  - Add a DOI badge with link if a DOI exists
- More to come?
  - Attempts and variations “just” formed, still form and require adoption and use
  - Be up to date, check what others do and how

Citation - community practices, e.g. R and CRAN

- Communities have their own conventions, get used to them
- Citing R And R Packages In Publications: citation(package)
  → https://www.rdocumentation.org/packages/utils/versions/3.3/topics/citation

- knitr example: citation('knitr') or toBibtex(citation('knitr'))
  → https://cran.r-project.org/web/packages/knitr/citation.html
  - Citation file → https://github.com/yihui/knitr/blob/master/inst/CITATION
  - Description file → https://github.com/yihui/knitr/blob/master/DESCRIPTION
  - Google Scholar lists R packages
    → https://scholar.google.de/citations?user=lkjqsd4AAAAJ

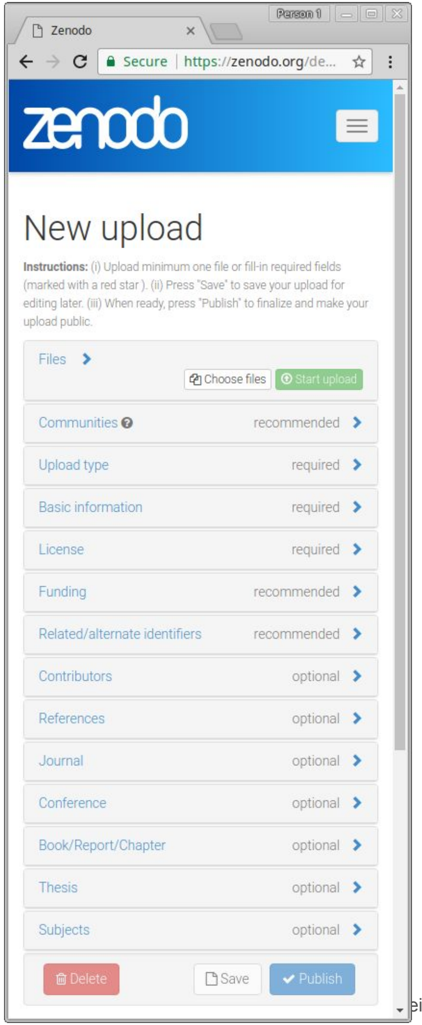
- Former days Google Scholar example
  → https://www.youtube.com/watch?v=LwBZgjlKkbM

Metadata for software publications

- DOI minting requires metadata information
  → Use https://search.datacite.org/works?resource-type-id=software
  → Select/click a software listed in results and open a “Download” option, e.g. JSON-LD

- Zenodo offers a user interface
  → It’s totally up to you and under your control
  → https://zenodo.org/deposit/new

- Further reading
  - DataCite
    → https://blog.datacite.org/metadata-schema-4-1/
    → https://schema.datacite.org/meta/kernel-4.1/
  - CodeMeta
    → https://codemeta.github.io/terms/



New upload

Instructions: (i) Upload minimum one file or fill-in required fields (marked with a red star). (ii) Press "Save" to save your upload for editing later. (iii) When ready, press "Publish" to finalize and make your upload public.

Files	
Communities	recommended
Upload type	required
Basic information	required
License	required
Funding	recommended
Related/alternate identifiers	recommended
Contributors	optional
References	optional
Journal	optional
Conference	optional
Book/Report/Chapter	optional
Thesis	optional
Subjects	optional

DOI versioning with Zenodo

- DOI versioning allows to cite all versions and a specific version of a record and allows to edit/update the record’s files after the initial publication
- How does DOI versioning work?
  - When you publish an upload on Zenodo for the first time, Zenodo registers two DOIs
    - a DOI representing the specific version of your record
    - a DOI representing all of the versions of your record
  - Afterwards, Zenodo registers a DOI for every new version of your upload
- Which DOI should I use in citations?
  - Ensure that other researchers can access the exact research artefact for reproducibility
  - Normally always use the DOI for the specific version of your record in citations
  - By default, Zenodo uses the specific version to generate citations
- Explore
  → http://help.zenodo.org/#versioning
  → https://doi.org/10.5281/zenodo.592020

Best practices for software citation

- Make your software citable
  - Publish it – if it's on GitHub, follow steps in https://guides.github.com/activities/citable-code/
  - Otherwise, submit it to Zenodo or somewhere else, with appropriate metadata, and get a DOI
  - Create a CITATION file, update your README, tell people how to cite
  - Secondary / workaround, write a software paper and ask people to cite that
- Cite someone else's software in a paper
  - Check for a CITATION file or README; if this says how to cite the software itself, …
  - … if not, do your best following the principles
    - Try to include all contributors to the software (maybe by just naming the project)
    - Try to include a method for identification that is machine actionable, globally unique, interoperable (perhaps a URL to a release, a company product number)
    - If there's a landing page that includes metadata, point to that, not directly to the software (e.g. the GitHub repo URL)
    - Include specific version/release information
  - If there's a software paper, can cite this too, but not in place of citing the software

Wrap-up

An incomplete checklist

- Select a proven and strong licence, e.g. GPL, EUPL, BSD, Apache
  - Be specific, name the license and version
- Check compatibility for distribution and resolve conflicts
  - Check up- and downstream compatibility; consider dual licensing
  - Check variable compatibility or exception list, e.g. provided in appendix
- Use additional licence provisions, e.g.
  - Add provisions for cases when distribution isn’t enforced, e.g. for publications
- Structure the software project / bundle a release
  - Document dependencies and licenses
  - Attribute other work
  - Enable citation
- Publish; a DOI is just a number
  - Ensure that all versions or releases can be referred at all and individually (versioning)
- Start somehow with the recommendations given
  - Spend time to make your own experiences and talk with others

Group work

Group work A

- Polish and finish your release
  - Anything is valid to gain experiences, don’t be shy
- License your code
  - Define the purpose
  - Consider dependencies and compatibility
- Enable citation
- Publish
  - Reserve a DOI
  - Update citation and readme information
  - Finish

LEIBNIZ INFORMATION CENTRE
FOR SCIENCE AND TECHNOLOGY
UNIVERSITY LIBRARY

TIB

Which questions do you have for us?

Contact information:
Martin.Hammitzsch@GFZ-Potsdam.de

Leibniz
Association