

Dr Wolfgang Straub
Augsburger Deutsch & Partner
P.O. Box 5860
CH-3001 Berne

www.it-recht.ch
wolfgang.straub@ad-p.ch

Open-source licence checklist

1. Choosing a suitable open-source model

The following **criteria** in particular must be borne in mind when choosing a suitable licence type to distribute software under an open-source model:¹

- **Compatibility** with third-party software licences
- **Acceptance** of the licence in the target community
- **Workload** for ensuring compliance with the licence chosen
- Existence of a legal enforcement authority

¹ See also the overview table of open-source licences in BEARDWOOD, JOHN P. / ALLEYNE, ANDREW C, The Price of Binary Freedom: The Challenge of Open Hybrid Software, CRi 2006, pages 97-104, pages 103f and the comparisons in THALHOFER THOMAS, Commercial Usability of Open Source Software Licenses: To what extent can software governed by GNU or alternative licenses be commercially exploited? CRi 2008, pages 129-136, pages 132ff; FRÖHLICH-BLEULER GIANNI, Urheber- und vertragsrechtliche Aspekte der Open Source Software (Copyright and Contractual Law Aspects of Open-source Software) in Jörg Florian S./Arter Oliver (publisher), IT-Verträge (IT Contracts), Volume 10, Bern: Stämpfli 2007, pages 179-232, pages 218ff.

2. Copyleft avoidance strategies

The following **strategies** are examples of how to avoid the copyleft effect:

- Instead of making **changes** yourself, get the **end users** to make them. Instead of modified code, they can be supplied with instructions on how to make changes to the program.
- **Distribute** programs **separately**: If your own software does not contain any protected code from open-source programs, it can in principle be distributed on its own.
- **Develop technical distancing strategies**: A special module might be created, for example, that sits between the open-source software and the new program. The copyleft effect only applies to the module. However, GPL v3 contains a protective clause designed to prevent technical circumvention of copyleft.²
- Obtain **parallel licences** from the rights owners. In this context, however, it must be noted that the program creator and the rights owner are not always the same (e.g. when software is developed under the terms of an employment or works contract).
- Obtain an **exception** from the Free Software Foundation (FSF). This presupposes, however, that the FSF has the power of disposal over the rights to the software in question. This is only the case if the rights owner has at least granted it such power in a fiduciary capacity.

² Section 5 para. 2 GPL v3. See also BEARDWOOD, JOHN P. / ALLEYNE, ANDREW C., Open Source Hybrids and the Final GPLv3, CRI 2008, pages 14-20, page 16.

3. Open-source compliance

When using open-source software in development projects, the following points in particular must be clarified and documented:³

- Draw up an **open-source strategy**: in which areas and under which conditions is open-source software to be used?
- **Which software** is to be used? Which building instructions, installation scripts, compilers etc. are to be used? Specifically, everything an expert needs to produce a workable program independently must be documented. All versions of the relevant open-source software must be documented as part of systematic release management.⁴
- **Which licences** apply to the software components used? Are these compatible with each other?⁵ How can any compatibility conflicts be avoided?
- In which form is open-source software to be **linked** to proprietary programs⁶ or third-party products and **conveyed**?⁷
- Which software components are subject to the **copyleft effect**?⁸

³ See also KUHN BRADLEY M. / WILLIAMSON AARON / SANDLER KAREN M., *A Practical Guide to GPL Compliance*, New York 2008, available online at www.softwarefreedom.org, page 2ff; BEARDWOOD / ALLEYNE, *Binary Freedom*, page 102.

⁴ The Fossology system, for example, can be used for this purpose. See www.fossology.org.

⁵ On compatibility issues, see STRAUB WOLFGANG, *Softwareschutz: Urheberrecht, Patentrecht, Open Source (Software Protection: Copyright, Patents, Open Source)*, Zurich 2011, margin nos. 675ff.

⁶ It must be noted in this context that, when code is used in third-party products, both the software developer and the creator of the end product must comply with the obligations relating to conveying the source code. On the content, see STRAUB, *Softwareschutz*, margin nos. 697ff. Where open-source software is integrated into end-user products, users must also be given instructions on how to install modified versions of the software on the product (Section 6, para. 3 GPL v3).

⁷ When standalone software components are assembled into packages, it must be clearly stated which licence applies to each component (e.g. by means of a folder structure with corresponding titles). On software aggregation, see also STRAUB, *Softwareschutz*, margin no. 716.

⁸ See STRAUB, *Softwareschutz*, margin nos. 689ff.

- **Publish** the source code correctly and ensure the availability of all versions (including building instructions etc.) for the minimum period stipulated by the licence in question.⁹
- **Display proactive efforts to ensure compliance** when accused of violations.¹⁰

⁹ Under the GPL, the entire source code must be available for at least three years after the last conveyance. If the software is subject to GPL v2, there may be a requirement to make physical data media available at your own expense (cost of media plus postage). See KUHN / WILLIAMSON / SANDLER, pages 4ff.

¹⁰ If the party committing the violation has received a first warning from the rights holder or no notice of termination is given within 60 days, the licence is permanently reinstated (section 8 para. 2 and section 3 GPL v3). Adopting a cooperative attitude can therefore make sense beyond the desire to avoid claims for damages.