

A Practical Guide to Using Free Software in the Public Sector

(with references to the French copyright law, when applicable)

Version 1.31

June 2010

This document is distributed under a Creative Commons ["Attribution + ShareAlike"](https://creativecommons.org/licenses/by-sa/4.0/) licence.



Author: Thierry Aimé (DGI – Ministry for the Budget, Public Accounts and the Civil Service)

With the participation of: Philippe Aigrain (Sopinspace), Jean-François Boutier (Ministry for Ecology, Sustainable Development and Spatial Planning), Frédéric Couchet (April), Elise Debies (DGME – Ministry for the Budget, Public Accounts and the Civil Service), François Elie (ADULLACT), Jean-Paul Degorce-Duma (DGSIC – Ministry of Defence), Esther Lanaspá (DGME – Ministry for the Budget, Public Accounts and the Civil Service), Sylvie Poussines (DAJ – Ministry for the Economy, Finance and Employment), Patrice-Emmanuel Schmitz (OSOR.eu).

Contents

1 -What is software?.....	3
2 -Legal regimes governing the use of software.....	3
3 -Who holds the copyright to software?.....	4
4 -What is a software licence?.....	4
5 -What is a free software licence?.....	4
6 - What is free software?.....	5
7 - What is proprietary software?.....	5
8 - Is software downloaded from the Internet free?.....	5
9 - How and where to find free software.....	5
10 -Checking that a software licence is free.....	6
11 -How to determine whether an "exotic" licence is free.....	7
12 -How to use and redistribute free software.....	8
13 -The most popular free software licences.....	9
14 -How to ensure that a government procurement contract will enable the sharing of free licence software ?.....	10
15 -How can free components be required in the STS?.....	11
16 -Can a specific free software be required?.....	13
17 -Can open standards be required in the STS?.....	13
18 -How to Select the most appropriate free software licence for distributing software.....	15
19 -The compatibility of free software licences.....	15
20 -How to select a licence for distributing documentation.....	17
21 -The liability of government entities that share software.....	18
22 -Who can decide to make a software development free?.....	19
23 -How to use a software component under a multiple licence.....	19

1 - What is software?

Software, also known as an **application** or **program**, performs a sequence of tasks on a computer. Software consists of or uses more elementary **software components** that are generally shared by program "libraries".

Software exist in two forms:

- **Source code**, which is a tree structure of text files that describes, in a programming language, the basic actions to be performed in sequence to accomplish a more complex task. This is the form in which the program is designed by the programmer.
- **Executable code**, that the computer is able to use directly.

In the case of interpreted languages (such as PHP or Perl), the source code directly controls the execution of the program.

In the case of compiled languages (e.g. Java or C++) the executable code is obtained by automatically translating the source code and may be stored for future execution. This process is known as compiling.

As defined in Art L. 112.2-13 of the French Intellectual Property Code, software also encompasses the preparatory design documentation, which includes:

- functional analysis and technical design reports, mockups and prototypes
- any online documentation that is included in the software.

2 - Legal regimes governing the use of software

As of "work of the mind", software is automatically covered by the applicable copyright law, i.e. no formality is required. In France, article L. 111-1 of the French Intellectual Property Code (CPI hereinafter) stipulates that: *"The author of a work of the mind shall enjoy with respect to this work, and by the sole virtue of its creation, an exclusive intellectual property right that is enforceable against all parties"*.

French copyright law consists of **proprietary rights** (which are equivalent to those granted under copyright law in the "Anglo-Saxon" countries) and **moral rights**.

Anyone who uses, copies, modifies or distributes software without the explicit authorization of the holder of the proprietary rights is guilty of infringement and may be sanctioned with up to three years of prison and a €300,000 fine (Art. L. 335-2 of the CPI).

Software use right includes the following rights, which however are subject to substantial restrictions pursuant to Art L122-6-1 of the CPI:

- Correct errors (unless the author reserves this right in a licence)
- Make a backup copy if this is necessary to preserve the use of the software
- Study the software's external operation
- Copy and translate the code to enable interoperability with other applications.

The protection of proprietary rights is limited in time, and in the case of France extends 70 years after the death of the creator, or author of the work (if a natural person) or the initial distribution or publication, if the author is a legal person. After this period, the software, or a given version thereof, enters the **public domain** and may be used by anyone without restriction.

The author's **moral rights** are inalienable. In the case of software, moral rights basically involve indicating the names of the people who worked on the software.

3 - Who holds the copyright to software?

The context in which the software is created determines who holds the copyright.

If the software is created by one or more people in a relationship of subordination to their employer, the authors retain the moral rights but the proprietary rights are automatically ceded to the employer. This also applies to people or firms working for the public sector (Art. L. 131-3-1).

If the software is created during the author's leisure time, under this person's initiative, using his or her resources and is not related to the author's work, the author is entitled to all associated moral and proprietary rights.

If the software is made to order, the author or authors retain the moral rights and the ownership of the proprietary rights must be initially specified in the contract between the person ordering the software and the person providing it. If nothing is specified contractually the person providing the software retains the proprietary rights. The ceding of proprietary rights is therefore an essential condition of any contract to purchase software. For contracts that are subject to the French Government Procurement Code (*code des marchés publics*), this issue is resolved in the "General Government Procurement Terms" (GGPT), but may be modified in the "Special Government Procurement Terms" (SGPT).

4 - What is a software licence?

A software licence (according to the UK spelling; US uses "license") is a contract that is generally considered to be a "contract for hire" (*contrat de louage*), between the software copyright-holder and the user of the software (the licensee), that specifies how the software may be used. This contract may require the total or partial ceding of proprietary rights.

5 - What is a free software licence?

A software licence is considered to be "free" if it ensures the user (or the "licensee" the four following "freedoms":

- Running of the program for any purpose whatsoever.
- Studying of the program's operation and adaptation to the user's requirements. Access to the source code must be provided to enable this.
- Distributing copies of the software, either gratis or for remuneration.
- Improving the software and publishing these improvements for the benefit of all software

users. Access to the source code must be provided to enable this.

The above definition is that proposed by the Free Software Foundation (<http://fsf.org>), which pioneered the free software movement. Another organization, the Open Source Initiative, proposes a 10-point definition of [an "open source" licence](#). Although these two definitions differ conceptually, from a practical standpoint they are equivalent and are accepted throughout the free software community.

Approximately 100 different free-software licences have been identified. Although what they may require from the user may differ, they all ensure the four essential freedoms listed above. For a list of the most common free-software licences see the [list of the most popular licences](#).

6 - What is free software?

Free software is software distributed under a free software licence. For us, free software is a synonym of open source software. The acronyms FOSS (free & open source software) or FLOSS (free, libre & open source software) are often used too.

7 - What is proprietary software?

Proprietary software is non-free software.

Note that proprietary software is not synonymous commercial software.

8 - Is software downloaded from the Internet free?

Freeware and shareware is usually downloaded from the Internet. Such software is distributed in binary or executable code and generally without the source code. The user's licence is often provided online with the application or is displayed when the application is installed. It rarely grants the user more than the right to run the software. Some software is provided on a trial basis only or for non-commercial use. In conclusion, the fact that software may be downloaded from the Internet is not a sufficient or even a necessary reason to conclude that it is free.

9 - How and where to find free software

Some collaborative software development sites are known for providing software distributed exclusively under a free software licence. This is even a requirement for distribution on these sites. Such sites include for example:

- [SourceForge](#) – This is the most popular of the collaborative development sites, with 158,000 projects in progress and 1,600,000 users registered. Any software distributed on Sourceforge is very likely to be free, since this is initially required for hosting on the site. The name of the software's licence is indicated directly on the project's homepage;
- [OSOR.eu](#) : This recent site (October 2008) was set up by the European Commission and is dedicated to free software aimed to the public sector. It hosts directly (at the end of 2009) 140

projects and federates the other European Forges that are operating for public sector (OSOPR search engine provide access to more than 2000 projects);

- [Adullact](#) – The forge of the “ Association Des Développeurs et Utilisateurs de Logiciels libres pour les Administrations et les Collectivités Territoriales” (France) hosts more than 450 software applications that are directly relevant for administrations, which may be local, regional or national. Adullact has taken over the project of the previous AdmiSource);
- [Plume](#) – Supported by the French CNRS network (UREC), this repository provides access to software aimed or produced by the high schools and research communities. The relevant free software licence is clearly mentioned;
- [Framasoft](#) – This website offers a database of over 1,200 Windows-compatible applications under free licence. Inclusion in this site is a good guarantee that the software is free.
- [Apache](#) – The Apache Foundation has a very strict project governance policy. Its bylaws require that only projects under a version 1.0, 1.1 or 2.0 Apache licence may be hosted. The free nature of all software components is guaranteed.
- [Debian](#) – The inclusion of software in the "Main" and "Contrib" sections of the Debian community's distribution repositories is a very good guarantee that it is free. Moreover, [Open Source Initiative's](#) 10 criteria for defining an open-source licence are based on Debian's free software principles. To verify that software is listed in Debian's "Main" or "Contrib" section see this page: <http://www.debian.org/distrib/packages>.
- [FSF/UNESCO Free Software Directory](#) – The Free Software Foundation and UNESCO have identified over 5,000 programs that are considered to be free.

Please note! Software available on [Freshmeat](#) is not necessarily free!

10 - Checking that a software licence is free

Some free software have their own development and distribution web site. A simple search on the Web is usually sufficient to locate these sites. Once the site's URL has been located, searching for "license" (or "licence" if the UK spelling is used) within the site should locate a page that indicates the licence under which the software is distributed, as seen in the following example for the Evolvica project:

- The term "Evolvica" is used to find the project site's URL: <http://www.evolvica.org>
- A search with "license site:<http://www.evolvica.org>" locates the <http://www.evolvica.org/license.html> page that provides information concerning the licence.

In some rare cases reliable information concerning a software's licence cannot be found directly. As a last resort it may be necessary to check the source code, although this does require some technical expertise. The source repository is generally flagged in some way or another from the project's host site.

The source code may be provided in various forms, such as:

- A downloadable archive (e.g. zip, tar, tgz or rar). A standard archive manager will be sufficient to navigate through the archive's contents.
- A browsable versioned repository (cvs, svn, etc.). A standard web browser will enable you to navigate through the source code.
- A non-browsable versioned repository, (cvs, svn, etc.). You must have the appropriate client

software to enable the connection string to "export" from the repository and then deposit a copy of the source code tree in your own local file system.

In any case, at the root of the source code tree there must be a file named "LICENSE.TXT", or something similar, that contains the complete text of the licence that covers the project.

Once the text of the licence has been recovered it must be ensured that it is recognized as being free or open-source by checking on the FSF or Open Source Initiative websites.

- On the FSF's site (<http://www.fsf.org/licensing/licenses/index.html>) the list of licences is divided into three sections. The first section lists all free software licences that are compatible with the GPL licence. The second section lists free licences that are not compatible with the GPL licence. The last section lists all licences that cannot be considered to be free for various reasons.
- On the OSI site (<http://www.opensource.org/licenses/index.php>) a complete list of all licences that OSI recognizes as "open source" is provided, along with the full text of each licence.

Specific case of the European Union Public Licence: The EUPL v1.1 is a free software licence (according to the FSF) and is the sole licence that has been certified by the OSI with identical value in 22 languages of the European Union. The EUPL can be used and produced (i.e. in the framework of some litigation) in courts and administrations of numerous States, without the obligation or the risk to call upon a sworn translator. The EUPL is also compliant with the European Member States laws. It is compatible with the GPL v2 et CeCILL-v2 licences (among others).

Specific case of CeCILL licences. The FSF considers CeCILL v2 and CeCILL-A licences to be free. No opinion is given for CeCILL B and CeCILL-C licences, perhaps because they are too recent. Yet the latter are genuine free software licences and are listed along with the various licences recommended (see [CeCILL licences and the public sector](#)).

11 - How to determine whether an "exotic" licence is free

If the licence is not listed as either free or open source by the aforementioned bodies, it must be read very carefully before coming to an opinion. The FSF's four criteria are certainly the easiest to use for this purpose. If even one of these criteria is not met the licence must not be considered free.

The following characteristics also conflict with the definition of free software:

- Preventing or limiting pecuniary compensation for redistributing the software
- Requiring the redistribution of any changes or corrections made for internal purposes.
- Revoking the licence of all users in the event of a simple accusation of patent infringement by any third party
- Revoking the licence on any patented content if the source code is modified
- Prohibiting the redistribution of the source code or of a patch without the original author's consent.
- Prohibiting redistribution to certain countries (e.g. export restrictions imposed by the United States)
- Requiring that all modified versions be sent to the original author
- Prohibiting certain uses of the software for (such as for defense purposes or spying for

example)

- Prohibiting the software from being used by certain categories of people, for example on the basis of their skin color, gender, language, religion, political opinions, nationality or social origin.
- Prohibiting the redistribution of the source code without the binary code
- Limiting the licence's period of validity.

In the case of the Evolvica software presented above <http://www.evolvica.org/license.html> the licence is based on the Artistic License 2.0, but the authors do not guarantee that it is compatible with the latter.

Exercise: is the "Evolvica Artistic License" a free software licence?

12 - How to use and redistribute free software

When a user acquires software, whether free of charge or otherwise, he or she must observe the terms of the software licence, while bearing in mind that everything that is not explicitly authorized is prohibited¹. A distinction is generally made between two main types of actions:

- Software use and modification
- Software redistribution, with or without modification.

There are no limits to how software distributed under a free licence may be used or modified, as long as the software remains within the user's organization. However, if the software is redistributed outside of this organization, the extent to which the initial licence is to be preserved, gives rise to three distinct types of free licence:

Strong copyleft-type licence – The software may be redistributed with or without modification, but always under the initial licence. In addition, any components that may be combined with the software in any way to form a new and larger software development will also be covered by the initial licence. For example, since the Linux kernel is under a GPL licence, the new Ext4 file system currently being developed will also be covered by the GPL licence.

Weak copyleft-type licence – Although the software must still be redistributed, with or without modification, under the original licence, code under other and even proprietary licences may be added to provide new functions. For example, for OpenOffice.org, which is under an LGPL licence, Sun distributes StarOffice, which although still under an LGPL licence has been enhanced with proprietary add-ons.

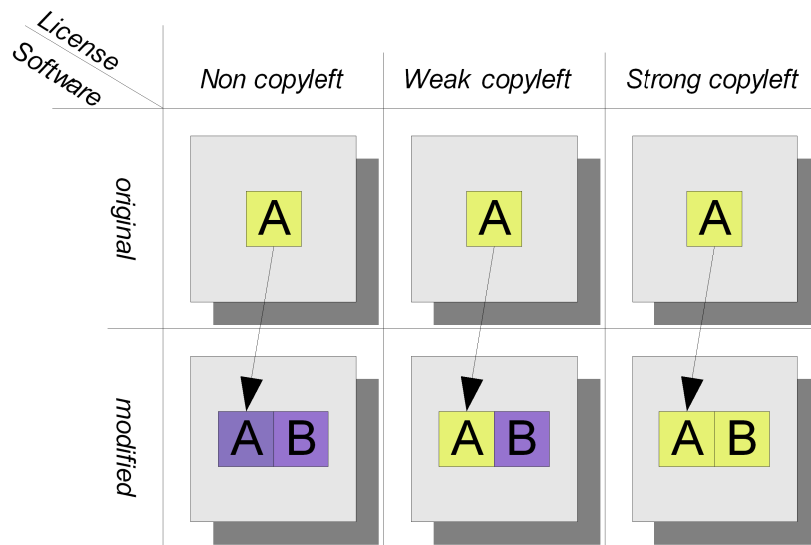
Non-copyleft licence – The software may be redistributed, with or without modification, under another licence. For example, components of the FreeBSD operating system under a BSD free licence are used in the Mac OS X operating system, which is in turn redistributed under a proprietary licence.

Contrary to what many people think, **no** free software licence requires anyone to cede their rights to the software they develop. A copyleft-type licence simply requires that the people who will receive your software version will be able to enjoy the same user rights you yourself enjoy. This is simply a **requirement of reciprocity**.

The diagram below shows the various ways in which a licence for component A may or may not be

¹ Pursuant to the Intellectual Property Code, which stipulates that all rights granted by the author must be specified.

modified when a new component B is added to make a larger application:



Under a non-copyleft type licence, the component B licence could be applied to component A. Under a weak copyleft type licence component A must retain its licence, while component B may retain its licence. Lastly, under a strong copyleft licence, the component A licence must also apply to component B.

13 - The most popular free software licences

The table below lists the most popular free software licences, the EUPL and the CeCILL licences. The full text of these licences as available on the Open Source Initiative² website may be obtained by clicking on the link. We also indicate the type of each licence:

Licence	Type
GNU General Public License (GPL)	Strong copyleft
GNU Library or "Lesser" General Public License (LGPL)	Weak copyleft
Apache License, 2.0	Non-copyleft
New BSD license	Non-copyleft
MIT license	Non-copyleft
Mozilla Public License 1.1 (MPL)	Weak copyleft
Common Development and Distribution License	Non-copyleft
Common Public License 1.0	Weak copyleft
Eclipse Public License	Weak copyleft
European Union Public Licence (EUPL)	Strong copyleft
CeCILL V2	Strong copyleft

² <http://www.opensource.org>

Licence	Type
CeCILL-B	Non-copyleft
CeCILL-C	Weak copyleft

14 - How to ensure that a government procurement contract will enable the sharing of free licence software ?

Before a government entity enters into a public procurement contract with an external supplier to develop software, it must first determine its requirements, draft specifications and make these available to competing firms.

If a need to share this software with some other government entity is identified that will require a free licence, this requirement must be clearly expressed in the bidding documents and it must be ensured that the solution selected will enable the software to be shared effectively and without restriction.

The contract will generally refer to³ the General Government Procurement Terms (GGPT), which set forth the terms and conditions that will apply to the contract, subject to other particular provisions.

The Intellectual Services GGPT (IS-GGPT)⁴ currently offers the most appropriate terms for procuring IT services. This document deals with the question of ceding the intellectual property rights to the service provided. None of the options proposed by this GGPT currently⁵ deals adequately with the problem of how rights should be ceded to enable the sharing of software under a free software licence. The following clause must therefore be systematically included in the Special Government Procurement Terms (SGPT):

Article XX – Proprietary rights to software developments

The contractor shall cede to the contracting entity, on a non-exclusive basis, all rights to use or copy the software, including the software's distribution on all media, presentation, adaptation and translation for the time these rights remain protected by copyright in all jurisdictions.

The contractor shall provide the contracting entity with the source codes of the software developed.

After accepting the services, the contracting entity shall authorize the contractor to use the software developed under the contract for commercial purposes.

³ Even though this is not mandatory.

⁴ New versions of the current GGPTs (which cover standard supplies and services, intellectual services, industrial contracts or construction work) are to be issued in 2008, along with a new GGPT for telecommunications and IT services that should be particularly well suited for government contracts for IT services. The latter document will contain terms and conditions regarding intellectual property that are similar to those of the current Intellectual Services GGPT.

⁵ This guide was drafted in mid-2007. The Finance Ministry's legal department (the DAJ) oversees the drafting of these GGPT and may decide to adapt the Intellectual Services GGPT as required.

Observation:

The ceding of the rights is not exclusive since the contractor still retains the right to use the developed software, even for remuneration, as stated explicitly at the end of the above article.

The contracting entity safeguards the software developed by obtaining all of the rights that will enable it to seek competitive bids for subsequent maintenance under an applications maintenance contract⁶.

The contracting entity ensures that it will be able to pool its investment with other government entities, possibly by means of a free licence, which may be of any type since the above clause simply provides for the ceding of the necessary intellectual property rights.

Specifying a particular free software licence in the SGPT is not appropriate since this will depend largely on the technical components selected to develop the application.

However, if specifying a particular licence is deemed useful it must be indicated in the Special Technical Specifications (STS).

15 - How can free components be required in the STS?

When setting up or modernizing their information systems, government agencies must consider various requirements, such as the need to pool their investments and share resources, to ensure the interoperability, independence, open-endedness and control of the software and to reduce costs.

These objectives can often be met effectively by selecting free software or solutions consisting of free components. Care should be taken to observe the rules that govern public-sector procurement and the Government Procurement and Competition codes.

The principles set forth in the first article of the Government Procurement Code – and in particular ensuring equal access to government contracts and equal treatment of bidders – require that a competitive bidding procedure be used and also that the most cost-effective bid be selected. With respect to competition law, articles 87 and subsequent of the European Union treaty and their associated exemptions must also be observed. These articles of the above code and treaty are intended to ensure that all bidders will have equal access to government procurement contracts and that none will be automatically disqualified by some arbitrary obstacle.

It is when requirements are initially expressed that all of the essential characteristics of the desired service or product must be examined and precisely specified. It is during this phase that any need to share or transfer software, for example, must be identified and expressed. The nature and content of the requirements expressed will determine what specific characteristics may be necessary and consequently whether free licence software should be preferred or even required in some cases.

The type of licence is therefore not selected in advance but instead must address needs that are first clearly specified, as explained in the above paragraph.

The box below shows the various non-functional requirements that may be specified in the STS.

⁶ Obviously, in subsequent applications maintenance contracts, the applications to be maintained and their origin must be specified in the terms and conditions or specifications, including brand names, versions, etc., regardless of whether the software is developed under a free, partially open or proprietary framework.

Use requirements

If the software components to be supplied under this contract are licensed, these licences must be indicated in detail and enable the following:

1. Unrestricted use of the software component; and in particular with no restriction as to the number of workstations on which the component is installed, simultaneous users, computers hosting the service, sites, documents processed, etc.
2. Study of the software component, by enabling access to the source code to ensure perfect interoperability with other systems.
3. Correction and improvement of the software by the government entity or a third party to ensure that the component provider does not have exclusive access to the subsequent applications maintenance contract. The source code must be supplied.
4. Redistribution of identical or modified copies of the software subject to the same rights, to ensure that public funds are disbursed only once.

Each software component supplied must be accompanied with all of the source files necessary for their compilation.

The bidder must show that its proposal complies with the above principles, by indicating for each software component to be included, its name, version, licence (with a link to an online version of the licence on the ISO or FSF websites), the copyright holders and the URL where the component may be downloaded.

The contractor must provide all additional information that may be required throughout the contract and shall deliver a **Compliance Report** when the contract is completed that provides a complete description of all of the components reused and those specifically developed for the needs of the contract.

If the licence of some components is more restrictive with respect to any of the four points above, the bidder must provide thorough justification for its choice and guarantee the compatibility of the overall software development.

Note: The contracting entity will end up with an application that includes components under a free software licence and other specific components to which it will hold proprietary rights. It may therefore, if it deems appropriate, distribute the software development under a free software licence.

A more radical approach

Because of licence proliferation - a great number of new licences that are frequently not compatible - it becomes more difficult to distribute a combined work (as a whole) under a single licence. Since 2009, for avoiding such issue and simplifying distribution, some administrations publish more radical provisions making a single licence compulsory. This is the case of the Spanish central ICT agency Red.es that selected the EUPL.

In Spain, article 16 of [Royal decree 4/2010](#) states⁷ that for transferring and reusing public sector application and documentation, “the **EUPL** will be procured, without prejudice of other licences that can guarantee the same rights”.

⁷ http://www.csae.mpr.es/csi/pdf/ENI_INTEROPERABILITY_ENGLISH_final.pdf

In Malta, a [1st June 2010 directive](#) states⁸ that “Government shall seek to facilitate distribution of OSS Government solutions under the **EUPL**”.

Single distribution licence requirements

In case the contractor integrates in the development that is the object of the contract with modules or elements owned by third parties, he must first obtain from the legal owners the licences and rights necessary to transfer the ownership of the development to <the public authority>, which will submit it, including the elements that are performed under the contract (such as fonts, dll, scripts, etc..) to the public licence EUPL. In any case the total and final result of the development and the overall project (meaning the combined work) will be subject to a licence EUPL.

The advantage of such provision is to request from contractors that the provided ICT solution - the combined work (as a whole) could be distributed under the single licence selected by the administration. However, this imposes additional constraints to contractors: in addition to their own contributions, they can make use of all components that are distributed without copyleft under a permissive licence (BSD, MIT, Apache etc.) and of all components that are distributed under a “weaker” copyleft licence (LGPL, MPL, CPL, Eclipse, OSL etc.). At the contrary, contractors can not legally use components distributed under “strong” copyleft terms (Gnu GPLv2 or V3), except if their copyright holder is entitled to dual licence these components under a LGPL type of licence in the framework and for the purpose of the contract.

In most cases, components licensed under a weak copyleft licence will keep/retrieve their primary licence if they are distributed alone, out of the context of the combined work where they are integrated. The primary licence covers also the possible improvements of these components.

16 - Can a specific free software be required?

With a few rare exceptions, free software is not purchased since it can be used freely. As a result, the rules that govern government contracts do not apply to the appropriation of free software.

If the contract is to provide a solution there is no reason to initially require that a specific software be used, regardless of whether this software is free or proprietary.

On the other hand, if the contract is to provide a service or services, the software for which the service is to be provided must be specified. For example, in the case of a government contract it is not Thunderbird software that is purchased but the installation and maintenance of Thunderbird.

17 - Can open standards be required in the STS?

Act 2004-575 of June 21, 2004 defines the term "open standard" as "any communication, interconnection or data-exchange protocol or any interoperable data format, the technical specifications of which are made public and which offers unrestricted access and use". The definition provided by the European IDABC project (version 1 of November 2004)⁹ is more complete:

⁸ http://ictpolicies.gov.mt/docs/GMICT_D_0097_Open_Source_Software_v1.0.pdf

- The standard is maintained democratically by a non-profit organization that is open to all interested parties.
- The standard's specifications are published and made available free of charge (or for a nominal cost) and may be used without restriction.
- Any patents that may apply to various features of the standard must be irrevocably made available free of charge (for those jurisdictions that recognize software patent rights).
- There is no restriction as to the standard's reuse.

A standard that is not considered to be "open" is referred to as a **proprietary standard**, particularly when the technical specifications that describe the protocol, encoding and organization of the data are not made public, or when their use is restricted by patent law. A proprietary standard is generally associated with a specific software.

An official standard (sometimes referred to as a "norm")¹⁰ is a standard that is recognized by a body that has been legally or formally mandated to develop standards. At the international level such bodies include the International Telecommunications Union (ITU) and above all the International Standards Organization (ISO), which federates such national standards bodies as AFNOR (France), ANSI (USA), BSI (UK) and DIN (Germany). The main European standards body is the European Committee for Standardization (CEN).

When it is necessary to make a technical requirement in an STS subject to an official standard, the following order of priority should be observed¹¹: national standards that transpose European standards into national law, European Technical Approvals, common (European) technical specifications, international official standards, other technical standards prepared by European standards bodies when there are no international official standards, national official standards and national "technical approvals".

The STS may therefore refer to a specific standard, as in the following example:

The technical solution proposed by the bidder must be based on the REF-NORM standard. If this standard is deviated from it must be shown that the contracting entity's requirements will be met.

An official standard (or "norm") is not necessarily an "open standard".

Open standards must be privileged each time it's possible because to guaranteeing the present and the future software's choice, they protect the overall strategy and sustainability of the information system. They are inherently pro-competition and therefore in harmony with the regulatory environment. The inclusion of open standards in the **General Interoperability Standard** will give them a stronger legal foundation. For the time being we may use the definition provided in Act 2004-575 of June 21, 2004 to require that an open standard be used, by including the following clause in the GGPT or even the STS:

In order to ensure interoperability between current and future systems, the bidder's technical solution must be based on an open standard as defined in Article 4 of Act 2004-575 of June 21, 2004.

⁹ <http://ec.europa.eu/idabc/en/document/3473/5887>

¹⁰ Article 4 of the ministerial order of August 28, 2006 concerning technical specifications for contracts and framework agreements.

¹¹ Article 3 of the ministerial order of August 28, 2006 concerning technical specifications for contracts and framework agreements

Such a requirement complies with Article 6 of the Government Procurement Code, which stipulates that: "technical specifications may not specify or refer to a particular production process or method, a specific origin or source, or a specific brand, patent or type, if such specification or reference may favor or eliminate some economic operators or products". Indeed:

- an open standard is not a production process or method
- an open standard does not indicate the person or entity that created it
- and open standard does not refer to any brand or patent.

if the need to respect official standards must admit exceptions (it must allow for the fact that a non-"standardized" product may comply with the standard in all respects and therefore be potentially "standardizable"), the clause that requires that the standard be "open" as this term is defined in the Act of 2004 is an "all-or-nothing" compliance clause. In other words, the bid either complies with the requirement or does not and cannot be accepted.

Requiring compliance with an open standard pursuant to the Act of 2004 is therefore entirely possible, provided that the specifications clearly indicate a need for interoperability, scalability sharing, etc.

18 - How to Select the most appropriate free software licence for distributing software

Considering that software that is developed using public money must not be appropriated in an exclusive manner by private interests and that an application under a non-copyleft type licence may be redistributed by anyone under a proprietary licence, even if it has not be modified, it follows that government entities should prefer to buy a copyleft-type licence whenever possible¹².

- If the government entity has the proprietary rights to all of the software's source codes, it is free to select the licence under which the software will be distributed. Version 2 of the CeCILL licence is recommended since it is in harmony with French law. For facilitating the distribution and the use in the whole European Union, the multilingual EUPL is recommended. The GPL licence is also highly recommended and is recognized worldwide. In the case of software that may be developed as a technical or functional component of a larger software package (a library or framework), the CeCILL C licence or the MPL licence are preferable. This limits any problems that may arise due to conflicts with other software components that may be used.
- Government entities often have software that consists of specific software developments to which they have proprietary rights and other components under various free software licences. Such software is referred to as "**derivative software**". Selecting a free software licence that will enable such software to be redistributed is often not easy (see the section below concerning the compatibility of licences). The **Compliance report** that is required for a specific software development will be a useful guide.

It should be noted that if an application incorporates a component under a proprietary licence, this application cannot be distributed under a free software licence. In this case the proprietary component may be removed and software users instructed to procure the component themselves. However, this certainly hinders and may even simply prevent the sharing of software.

¹² See Philippe Aigrain's article at http://www.adullact.org/article.php3?id_article=83

19 - The compatibility of free software licences

When a government entity plans to redistribute software that has been developed for it under a government procurement contract and under a free licence, or would like to redistribute existing software under a free licence, it must determine whether the various licences involved are compatible. In addition to any differences or incompatibilities that a lawyer might detect, it is important to understand and follow the spirit that underlies the concept of free software, as expressed by the four basic freedoms presented in section [5 above](#).

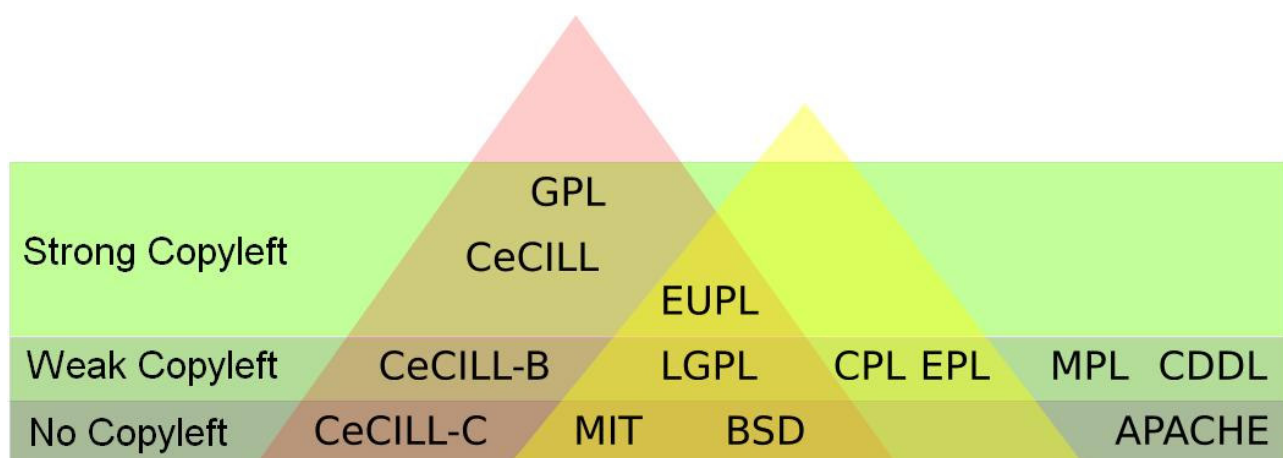
If software was developed using components under different licences, under what free licence should it be distributed? Is this always possible?

According to Benjamin Jean's method¹³ for assessing the compatibility of free licences, a licence is basically a set of obligations and rights. The basic principle is that the software licence must not grant more rights or require fewer obligations than the licences of its components. This is referred to as "logical compatibility".

We may illustrate this principle with the example of an application we would like to distribute under a GPL V2 licence after integrating a component under an Apache licence. All of the rights that the Apache licence allows this component are ensured under the GPL V2 licence. However, some of the Apache licence obligations are not required under the GPL licence V2, particularly with respect to patent law. A component under an Apache licence therefore cannot be used in an application distributed under GPL V2. Under the new GPL V3 licence this incompatibility no longer exists.

However a logical incompatibility issue may be resolved through a special agreement with the holder of the proprietary rights to the component to be incorporated in the software. This will require contacting the community responsible for the component. There is a good chance that a special exemption will be granted. In some cases the incompatibility is even resolved by an exemption clause within the component's licence.

The following table provides an overview of the compatibility of the most common free licences.



Compatibility is only really an issue when software is to be distributed under a strong copyleft type licence, either because this is considered the best option or because a software component is already

¹³ Benjamin Jean, « *Option Libre* » : *Compatibilité entre contrats. Mémoire*, 2006, Mémoire effectué dans le cadre du Master II Recherche - Droit des Créations Immatérielles sous la direction du Professeur Michel Vivant.

under a strong copyleft licence. The red triangle in the table above shows those cases where the GPL licence is the choice of reference. The EUPL compatibility area runs across the yellow and the red triangles. In addition, licences placed on the same horizontal line could cohabit. Outside this, there is incompatibility. For example, a component under an EPL licence is not compatible with software under a GPL or a CeCILL V2 licence.

Although software may be developed using modules under weak copyleft licences, this may prove difficult to manage since each component will retain its own licence and each of these must be observed. If possible, the software including each component may be relicensed under an overall licence that is compatible, i.e. that protects all of the rights and obligations under each licence.

In the case of the EUPL it is possible, - as for the weak copyleft licences - to integrate the software in a complex (or "combined") ICT solution, which will be distributed as a whole under another licence, provide this licence is in the compatibility list¹⁴ (this does not change the EUPL licence of the integrated component alone). Permitting this, the EUPL provides a relative « tolerance ».

20 - How to select a licence for distributing documentation

Creative Commons licences, like free software licences, were designed to enable the publication or distribution of non-software works subject to copyright. There are more than ten different types of Creative Commons licences, each of which allows the rights-holder to select a different combination of user rights and restrictions. The two most common Creative Commons licences used within the public sector are described in detail below. Both ensure "attribution", i.e. that the author of the work's name is mentioned.

[Creative Commons "Attribution - ShareAlike"](#) (CC-BY-Sa)

This licence is similar to a copyleft-type free software licence, since any derivative work (i.e. the work with its modifications) may be redistributed provided that the initial licence is maintained. Commercial use is possible without the author's consent.

Such a licence will enable the publication of documents that are clearly destined to be modified. This is the case in particular for software documentation, documents dealing with functional analysis, architecture, technical design, installation and operation, and user guides and tutorials. A contributor to a software's development is thus able to redistribute a new version with updated documentation. The guide you are now reading is also destined to evolve and to be enhanced and improved over time. This is why it was published under a CC-BY-Sa licence.

[Creative Commons "Attribution – Noderivs"](#) (CC-BY-Nd)

This licence prohibits redistribution of any modified version of the work. Since a derivative work cannot be created, there is no need to require distribution or sharing on an identical basis. This licence is incompatible with free licence principles. Commercial use is possible and without the author's authorization.

This type of licence is suitable for the publication of:

- official documents: legal documents, public reports, engagement letters and technical frameworks
- factual or contractual documents: minutes of meetings, decision points, STS and SGPT
- communication documents: political communiqués, institutional or personal interviews, personal accounts and speeches.

¹⁴ GNU GPL v. 2 , OSL v. 2.1 , v. 3.0 , CPL v. 1.0 , EPL v. 1.0 and CeCILL v. 2.0

Commercial exploitation may be prohibited by adding a "No commercial use" clause. This is possible for each of the aforementioned licences. But what exactly does prohibiting the commercial use of a work prohibit?

The licensee is not allowed to obtain a commercial profit or pecuniary compensation of any type whatsoever from the publication, presentation or distribution of the work, regardless of the media, format or technical process used. For example, someone who compiles a CD that contains documents under a non-commercial Creative Commons licence cannot sell this CD, even at its cost price, without the authorization to do so. Similarly, a printed copy of a document can only be distributed free of charge.

Of course, the author's permission can always be requested and the author can thus ensure that an unjustified profit will not be made at his or her expense. But why not trust users to judge whether or not the price charged for a CD or book is justified, since the work can be downloaded for free? This is the route that many developers of free software have taken with some success and which has enabled them to develop an income-generating business activity in addition to their not-for-profit development.

21 - The liability of government entities that share software

What liability can a national or local government entity incur by making software available under a free software licence?

Free and proprietary software licences generally refuse liability for any direct or consequential loss that may result from the use of their software.

Is such a clause compatible with the law (assuming the French law is applicable)?

Under French law liability may be contractually limited or even excluded. However, consumer protection law prevents liability from being completely avoided in contracts with consumers, pursuant to article L.132-1 of the French consumer code.

The same is also true for liability for defective products, pursuant to article 1386-15 of the French civil code, which allows liability for product defects to be excluded only in contracts between professionals. If software is clearly intended for professionals and IT engineers (which is the case of applications developed and used by government entities), liability for direct loss can therefore be excluded.

As for the government entity's liability if copyright is infringed, this is a risk even when software is not distributed under a free software licence. However, the more widely software is distributed, the greater this risk.

Copyright infringement – When distributed software contains a component or even a piece of source code that the government entity is not entitled to distribute, it may be held liable for copyright infringement. However, if the software was developed under a government procurement contract, the entity can refer to the **compliance report** to invoke the liability of the provider of the infringing software component, whether this component was provided fraudulently or through negligence.

Since the objectives of government entities and free software developers and distributors are so closely aligned there is very little risk of litigation between them and any disputes are very likely to be settled amicably.

Trademark infringement – A trademark is a distinctive symbol (a logo), word or group of words

that are recognized by law as designating a particular product, service or company. It is the government entity's responsibility to make sure that distributing software will not infringe on a registered trademark. It must be checked in particular that the software's name does not infringe. This may be easily verified online at <http://www.icimarques.com/>. Generally speaking, software must be shared under a white label or brand, i.e. with no distinctive marking other than that of the government entity¹⁵.

Patent infringement – Software patents in France and Europe have no legal validity as such. This is clearly indicated in article 52 of the [European Patent Convention](#) (EPC).

22 - Who can decide to make a software development free?

Depending on the case, and provided that rights may be ceded under the contract between the service provider and the government entity, the entity may make the decision of distributing the software under the terms of a free software licence in accordance with its usual decision-making organization and procedure.

For local government entities the decision to "liberate" the software is customarily made by an assembly of elected officials responsible for the use of public money, such as decided in Paris and Pierrefite sur Seine in September 2002.

23 - How to use a software component under a multiple licence

Although multiple licences are used in other industries, they are also particularly common in the world of free software. This approach involves distributing an application under two or more licences and allowing users to select the one that best suits their purposes.

Many companies that observe free software principles use multiples licences to support their publishing activities. The best known example of this is the database package MySQL, which is distributed freely under a GPL licence. Users who wish to free themselves of the reciprocity obligation to incorporate MySQL in a proprietary product, for example, have the option of entering into a commercial contract (i.e. for pecuniary consideration) with the company MySQL-AB.

Another advantage of a multiple licence is that it provides an effective solution to the incompatibility that may exist between the licences of software components that are to be integrated in the same application. Firefox for example is distributed under MPL, GPL and LGPL licences. A software publisher that would like to distribute a version of Firefox that includes a plugin under an Apache licence could not do this if the Firefox version was initially distributed exclusively under a GPL V2 licence, which is considered incompatible with the Apache licence. The multiple licence approach enables the publisher to obtain Firefox under an MPL licence and thus incorporate the Apache licence plugin, since the MPL and Apache licences are compatible.

¹⁵ Professional ICT insurers accept to cover litigation risks related to counterfeiting in copyright and in trade mark matters. It is therefore possible for public administrations to ask their contractors a specific guarantee concerning these points. At the contrary, in so far software patents would be valid in Europe, re-insurers have notified that they could not cover the related litigation risks.

