Treasury Board of Canada
Secretariat

Secrétariat du Conseil du Trésor
du Canada

Canada

Guide for Publishing Open Source Code

Published: 2020-04-28

Published by Treasury Board of Canada, Secretariat
90 Elgin, Ottawa, Ontario, K1A 0R5, Canada

Catalogue Number: BT22-253/2020E-PDF
ISBN or ISSN: 978-0-660-34786-8

This document is available on the Government of Canada website, Canada.ca

This document is available in alternative formats upon request.

Aussi offert en français sous le titre : Guide pour la publication du code source libre

Guide for Publishing Open Source Code

The Directive on Management of Information Technology, Appendix C provides Mandatory Procedures for Enterprise Architecture Assessment that will be used by departmental Architecture Review Boards (ARB) and the Government of Canada Enterprise ARB as an assessment framework to review digital initiatives to ensure the GC acts as a single enterprise and to ensure departmental alignment with the GC digital direction.

These align with the Digital Standards and the requirement in C.2.3.8 and C.2.3.9.5 of the Mandatory Procedures for Architecture Assessment which provide that all source code must be released under an appropriate open source software license, and when not, it must be shared within the Government of Canada.

As such it is recommended that where they have the right to do so, departments publish all source code as open source software, whether the software solution was (i) acquired as OSS; (ii) developed in-house by GC employees or (iii) acquired through the terms of procurement contracts where appropriate license terms were negotiated.

When releasing the code at large is not appropriate, or possible, it is recommended to start with sharing the source code internally to all departments, to the extent that the terms of the applicable license permit such sharing. Care should be taken to ensure that the license to Canada does not limit the distribution of such code to specified user departments. Doing so will help make the source code ready to be released publicly as open source where Canada has received the right to do so.

The steps to publish GC source code are:

1. Seek Approvals
2. Obtain Rights to Custom Code in Contracts
3. Consider Security Implications
4. Select Open Source Software Licence
5. Select Source Code Repository
6. Add Recommended Files
7. Publishing a Legacy Application
8. Work in the Open

Seek Approvals

▼ In this section

- Team
- Department

Team

The Directive on Management of IT support the Digital Standard #3: Working in the open by default, via Mandatory Procedures for Enterprise Architecture Assessment (C2.3.8).

> Share evidence, research and decision making openly. Make all non-sensitive data, information, and **new code developed in delivery of services open to the outside world for sharing and reuse under an open licence**.

Aligned with the Open Government vision, teams should by default consider adapting their process to develop as open source from the inception of projects to reduce the overhead required to release their source code later down the road.

It has also been found that doing so improves the quality of the code developed and encourages collaboration.

Department

Similar to open data or information covered by the Directive on Open Government, the publication of source code under an open source software licence, requires appropriate department or agency approvals.

That person may vary according to your department and delegation should be encouraged to the lowest level possible to encourage the release of open source code.

Obtain Rights to Custom Code in Contracts

The ISED Policy on Title to Intellectual Property Arising Under Crown Procurement Contracts provides that the contractor is to own the rights to foreground intellectual property (IP) created as a result of a Crown procurement contract. But when the Crown's intended use of the IP can be met through licence arrangements, it has the opportunity to seek the needed licence(s) whether broad or narrow.

As part of the discussion with the institution's legal services unit and the consideration of applicable policy, it should be noted that PSPC Standard Acquisition Clauses and Conditions Manual provides clauses to request a License to Material Subject to Copyright, which can use the clauses in contracts if the department or agency wants the copyright in the work to belong to the contractor but wishes to obtain a license to exercise all rights comprised in the copyright.

Departments or agencies are able to release code developed as a result of a Crown procurement contract under an open source software licence where such rights have been granted to Canada. The procurement contract could also require that the contracting body be responsible for publishing the source code under an acceptable open source software licence or contribute directly to existing open source software using that project’s licence, and such clauses would be effective where agreed to by the supplier.

Consider Security Implications

▼ In this section

Developing Software in the Open

- Keep sensitive data such as credentials secure and separate from source code.
- Avoid storing keys and other sensitive material in systems not approved for that purpose.
- Code reviews increase the likelihood of catching bugs, security vulnerabilities, and reduces the risk of committing sensitive data.
- Implement controls sufficient to prevent unauthorized or inadvertent changes such as code signing and repository user rights.

- Develop mitigation strategies and processes to address security related incidents.
- Embed security practices in your daily processes and methodologies.
- Leverage tools and services to automate finding known security vulnerabilities, possible secret keys and personally identifiable information.

Select Open Source Software Licence

▼ **In this section**

When the project is part of a larger open source software community, like plugins, modules, extensions, or derivative works of existing open source software, it is highly recommended to use the license which is usually used by the community.

When the project is started by the GC itself and isn’t related to an external community, the choice of the open source licence has to be based off of the licences of the components (3rd party libraries and framework) you will use and the desired outcome.

Recommended permissive licences

MIT

When to use Small, simple projects and scripts.

Licence text - MIT

Apache 2.0

When to use Larger software project use Apache 2.0 because it provides a grant of patents.

Licence text - Apache2.0

Recommended reciprocal licences

‘GPL 3.0 or later’, ‘LGPL 3.0 or later’ and ‘AGPL 3.0 or later’ refer to version 3 of the licenses with the “or later” notice. It is not a commitment to future version of the licenses.

‘GPL 3.0 or later’

When to use Software

Licence text - ‘GPL 3.0 or later’

‘LGPL 3.0 or later’

When to use Libraries

Licence text - ‘LGPL 3.0 or later’

‘AGPL 3.0 or later’

When to use Web applications and services

Licence text - ‘AGPL 3.0 or later’

Applying a License

To apply to source code, add the text of the chosen licence to a LICENSE.txt file in the root folder of the project. See Add Recommended Files. You could also just add the licence text directly in one of your source code file but by making it clearly available at the root of your project, you make it easier for people to find it and some collaboration platforms can automatically display your licence in the web interface.

If multiple licences can be applied, choose a licence which matches the goal of the project and its interactions with other projects. This tends to revolve around the decision of whether to apply a permissive or reciprocal licence.

Permissive

Beneficiaries of the OSS release Everyone: maximizes the scope of downstream users and has a broad appeal to the entire private sector… greater flexibility for end users, developers and companies to reuse the software as they see fit, including as part of commercial software.

Beneficiaries of downstream code changes The whole community, but only where the business, organization, or individual developer chooses to contribute modifications back under the permissive licence.

Licence complexity Often short, simple and understandable (for example, MIT).

Interoperability Permissively-licenced code can be included in projects under reciprocal licences, other permissive licences, or closed-source licences.

Reciprocal

Beneficiaries of the OSS release Everyone: but only where they are willing to release their software under the same licensing terms that were granted to them. Appropriate in cases where it is important to receive back downstream changes, or where it is important to ensure that work built on an initial investment remains free and open source software... can also put a focus on benefiting other private-sector businesses that provide services and support.

Beneficiaries of downstream code changes The whole community in every case where a business, organization, or individual developer distributes the modifications or contribute modifications back under the reciprocal licence.

Licence complexity Relatively long and complex legal jargon (for example AGPL 3.0).

Interoperability Reciprocal-licenced code cannot generally be included in a project under any other single licence.

Key Differences

The differences amongst the GPL suite of reciprocal licences illustrate how the type of distribution and extent of integration interact to determine when a reciprocal obligation engages, as shown in the following table.

Derivative work of the original

Derivative work of the original - Distribution of source code

GPLv3: **Yes** - LGPLv3: **Yes** - AGPLv3: **Yes**

Derivative work of the original - Provision of access over a computer network

GPLv3: **No** - LGPLv3: **No** - AGPLv3: **Yes**

Derivative work, but only links to original

Derivative work, but only links to original - Distribution of source code

GPLv3: **Yes** - LGPLv3: **No** - AGPLv3: **Yes**

Derivative work, but only links to original - Provision of access over a computer network

GPLv3: **No** - LGPLv3: **No** - AGPLv3: **Yes**

Collection, including the unchanged original

Collection, including the unchanged original - Distribution of source code

GPLv3: **No** - LGPLv3: **No** - AGPLv3: **No**

Collection, including the unchanged original - Provision of access over a computer network

GPLv3: **No** - LGPLv3: **No** - AGPLv3: **No**

Outbound Rights

You should always make sure that the outbound rights associated with the licence selected do not exceed inbound rights of any software components used in the source code; e.g.: it would not be legally possible to release a project under an MIT licence (permissive) if software components used within it were originally released under GPL3 (reciprocal).

Copyright

The Canadian Copyright Act (section 12) provides that where any work is, or has been, prepared or published by or under the direction or control of Her Majesty or any government department, the copyright in the work shall, subject to any agreement with the author, belong to Her Majesty. This applies to source code developed by Government of Canada employees.

However, Government of Canada employees have Moral Rights and as the author of a work has the right to the integrity of the work and the right to be associated with the work as its author by name or under a pseudonym and the right to remain anonymous.

Appropriate Government of Canada Copyright Identification

As per the Crown Copyright Request article found on Canada.ca, the following structure should be applied for the Government of Canada Copyright notice.

> Copyright (c) Her Majesty the Queen in Right of Canada, as represented by the Minister of (legal departmental name), (year of publication).

Replace the **(legal departmental name)** and **(year of publication)** with the appropriate information.

This notice should be added to the LICENCE file in your project. See Add Recommended Files

Select Source Code Repository

▼ **In this section**

Recommended public source code repositories for Government of Canada open source code are:

- GitLab
- GitHub
- framagit
- Bitbucket

The Government of Canada also has an internal source code repository available to all departments and agencies.

- GCcode (internal to Government of Canada only)

Organizations

Departments and agencies are free to choose the platform that best suits their operational needs but their projects should, where possible, all be regrouped under a unique organization or group. This would help discoverability of your projects but also help increase collaboration opportunities.

Version Control System

The recommended version control system for GC open source code is Git. Departments are also encouraged to use Git to manage their source code internally.

Add Recommended Files

Before publishing, source code should include the following file:

- a LICENCE (see Select Open Source Software Licence) file containing a copy of the licence under which the source code is released;

By default, a project without an open source licence applied to it would only be released under the Crown Copyright.

> Note: the open source licence itself can be applied in the source code directly but it is highly recommended to put it in a separate LICENCE file at the root of your project directory.

Additionally, the following are recommended as best practice:

- a README.md file providing information about the project, how to use it and general documentation.
    - It is also recommended that this file be bilingual to increase use and contribution to the project.
- a CONTRIBUTING.md file explaining how to contribute to the project.
- a SECURITY.md file explaining security policy as well as procedures for reporting security vulnerabilities.
- a CODE_OF_CONDUCT.md file explaining the values and ethics for the public sector employees and for the project.

Examples of these files are available in the Template Repository.

Publishing a Legacy Application

Publishing a legacy application can seem like a lot of work but it is feasible and actually a good investment if the application will continue to be used in the future. Documentation could be improved during the release project to help increase community contributions.

Additionally, releasing a legacy application may lead to reuse and increase in development contributions from interested parties. It may revive the active development of the application, providing it with enhanced features and bug fixes.

Vulnerability risks already exist and releasing it as open source doesn’t change their state. One way of limiting those risks is to not provide the configurations of the production version.

Scanning tools with advanced functionalities and security tests should be considered to help the development teams speed up the review and clean up process.

Work in the Open

▼ **In this section**

Release Early, Release Often

It is recommended that the source code be released as early as possible in the project’s life cycle to avoid the overhead of publishing source code late in the process. The public source code repository is encouraged to be the single source of truth where developers are working. The latest code version may not necessarily mean it’s the version deployed in production.

You’re In Control

When working in the open teams still have control over what makes it into the source code and a chance to review contributions from internal and external developers. Access rights can be configured for repositories to ensure only authorized team members can accept changes. Others may distribute modified version of your code, but it doesn’t mean the changes have to be accepted as part of your code.

Identify as an employee of the Government of Canada

Employees should use their full name and Government of Canada email address for all code contributions to public repositories while acting within the scope of their duties or employment.

Official languages

The Policy on Official Languages does not apply to software source code (including inline comments).

However, the same rules on OL as any other applications developed by the GC or on behalf of the GC in the past should apply in terms of end user documentation, whether the project has been released or not as open source software.

Community

Building a welcoming community can influence your project’s future and reputation. By providing a positive experience for contributors and making it easy for them to interact with the project team, you encourage them to keep coming back and contributing. You should respond to questions, bugs and merge requests to encourage people to continue helping you.

It is recommended to include a README.md and CONTRIBUTING.md file with your source code. See Add Recommended Files.

Contributor License Agreement

Government of Canada projects don’t require contributor license agreements, but rely on the open source software licenses providing the necessary terms. Contributions are made under the same license under which the project is released and that authors retain their copyright for their contributions.

Open Resource Exchange

It is highly recommended that departments add a link to their projects’ source code repositories on the Open Source Code section of the Open Resource Exchange. This will help raise visibility to all the projects managed by the GC and potentially increase collaboration.

Instructions for how to update the data can be found on GitHub - Open Resource Exchange Data.

Date modified:
2020-01-10