Guidelines and Procedures for Project Management

Coin-OR Foundation

December 2, 2009

Contents

1   Introduction

The Coin-OR Foundation repository consists of a variety of independently managed open-source projects of potential interest to practitioners of operations research, including students, researchers, and professionals. In order to ensure that the repository consists of high-quality projects that are actively and effectively managed and to minimize the Coin-OR Foundation's legal exposure, these guidelines and procedures for project management have been established by the TLC in accordance with the Repository Management Policy established by the SLB. Most of the Foundation's legal exposure comes from the fact that our mission is to publicly disseminate intellectual property (IP). A guiding principle for project managers is to ensure that the origin and ownership of all significant IP associated with a project should be established before being offered for distribution (or redistribution). This document, along with the accompanying Foundation policy document, provides a minimal set of guidelines to ensure that the Foundation is in compliance with IP law and that quality standards are upheld. Project managers are encouraged to establish their own policies and procedures to maintain high quality standards for their projects.

Structure of this document. Section 2 lists individuals responsible for implementing these procedures. Section 3 lists useful contact information for easy reference. Section 4 introduces the project classification scheme and other attributes associated with projects. Section 5 describes the procedures for contributing new projects to Coin-OR. Section 6 describes the procedures for contribution to an existing project. Procedures for ongoing maintenance of projects are described in Section 7. Required forms appear in Appendices A and B. A checklist for new project submissions appears in Appendix C.

Changes to this document. Maintenance of this document is the responsibility of the TLC. Coin-OR Foundation Members may suggest revisions or additions at any time. Suggestions should be sent to the chair of the TLC. The chair will assign a member of the TLC to evaluate the suggestion. The assigned TLC member will produce a recommendation for action or no action on the suggestion, along with a justification. Both the recommendation and the justification must be approved by a simple majority of the membership of the TLC. The decision of the TLC is final. Suggestions may be resubmitted, but resubmissions must substantially refer to the TLC's response to all related submissions.

2   Responsibilities

According the Repository Management Policy of the Coin-OR Foundation, responsibility for maintaining the quality and integrity of the Coin-OR Foundation repository is the overall responsibility of the TLC. These procedures contain the implementation of that policy, as interpreted by the TLC. Responsibility for implementing these procedures and ensuring that they are followed falls on two individuals appointed by the TLC for indefinite

terms. The *submission manager* is responsible for determining the acceptability of new projects submitted as candidates for acceptance into the repository. The *repository manager* is responsible for ensuring that existing projects adhere to the standards described herein and are classified appropriately. He is also responsible for controlling access to the repository.

Individual project managers are responsible for maintaining their own projects and for being responsive to any requests from the repository manager or the TLC. A guiding principle of these procedures, however, is to avoid placing unnecessary requirements on project managers while encouraging best practices by rewarding exemplary project management with increased status conferred through the project classification scheme (see Section 4).

3   Contacts and Information

This section lists the email addresses and web page URLs that are mentioned in the remainder of this document or that might be helpful for getting additional information.

Email contacts:

Repository manager	`repository@coin-or.org`
Submission manager	`submission@coin-or.org`
Submission coordinator	varies with submission
Secretary	`secretary@coin-or.org`
TLC	`coin-tlc@list.coin-or.org`

Web pages:

Coin-OR	`http://www.coin-or.org/`
CoinHelp	`https://projects.coin-or.org/CoinHelp`
Project list	`http://www.coin-or.org/projects.html`
Submission status	`https://projects.coin-or.org/CoinTLC/wiki/NewSubmissions`

4   Definitions

A *project* consists generally of a collection of related digital files with a common purpose. Examples of projects include collections of source code for building executable software, collections of data files comprising instances of operations research problems, tutorials, and documentation, among others.

Coin-OR projects are divided into two classes according to the level of development activity and available support. An *active* project is a project actively maintained with publicly accessible channels for reporting bugs or suggesting improvements. An *archived* project is a project without such channels and made available "as is." An archived project might consist of legacy software that does not build but is deemed interesting for historical reasons. Alternatively, it might be a previously active project that is no longer maintained. It could also be software submitted to the repository by someone who is not interested in actively

maintaining it.

Coin-OR projects are also classified as either *regular* or *umbrella projects*. An umbrella project consists of a collection of related subprojects. Examples are the Cut Generator Library (*CGL*) project or the Open Solver Interface (*OSI*) project. Active umbrella projects have one or more overall project manager(s), as well as subproject managers for each subproject. Any project that is not an umbrella project is a *regular* project.

Projects containing source code are divided into five levels reflecting the status of the development of the project:

- Level 1: Projects that are not yet functional, that cannot be built without special tools, or that otherwise do not meet the standard for a Level 2 project.

- Level 2: Projects that can be built following installation instructions using commonly available tools and that do something useful on at least one platform.

- Level 3: Level 2 projects that also have substantial documentation, tutorials, example codes, or other aids to enhance usability.

- Level 4: Level 3 projects that have versioning, a stable release, and a substantial unit test.

- Level 5: Level 4 projects that have a binary distribution.

The procedure for classifying projects is described in Section 5.3.

5   Establishing a New Project

The first step in establishing a new project is to contact the submission manager. The submission manager then delegates the task of reviewing the submission according to the procedures outlined below and summarized in Appendix C to a *submission coordinator*. The coordinator creates the submission status page for the submission and is then responsible for reviewing the submission and updating this information as the submission process proceeds. The submission coordinator is also the primary contact for all questions regarding the submission during the process. At the completion of the procedure, the submission coordinator makes a recommendation to the TLC regarding the acceptability of the project and its initial classification. If the coordinator rejects the submission, the rejection is automatically appealed to the full TLC and can only be upheld by unanimous vote of members present at a regular meeting of the TLC. The decision of the TLC is final. The project manager may also appeal the initial classification, as detailed below. The following sections detail the main procedures and criteria to be considered by the coordinator to determine whether a project is acceptable for inclusion in the Coin-OR Foundation repository.

5.1   Scope of Contributions

To be acceptable, the project must be within the scope of the repository. To determine whether the project is in scope, there are several litmus tests:

1. Is the project potentially of interest to OR professionals?

2. Is the project's goal to build, use, or otherwise support the development of open-source software?

Any project for which the answer to these questions is "yes" is a candidate for inclusion in the Coin-OR software repository. The term "project" is interpreted broadly here. Usually, projects of interest consist mainly of source code for software implementing operations research techniques. However, Coin-OR also welcomes contributions of data sets, tutorial material, and documentation, as well as projects supporting the development of open interface and data transmission standards.

5.2   Legal Documentation

Coin-OR has different requirements for contributions that form new projects and contributions to existing projects. In both cases, the purpose of the requirements is to balance a desire to ensure that ownership of any IP associated with the project is respected, with the bureaucratic burden resting primarily on contributors and project managers. The requirements have two goals:

- Determine the copyright holders and any other IP owners.

- Establish that the IP owners have agreed to release the contribution under an acceptable license.

The responsibility for gathering the required information falls on the *contributor*, the person who submits the contribution to Coin-OR. The submission manager will help clarify the requirements for contributors.

For contributions that form new projects, the contributor must provide the following:

- *Contributor's Statement of Respect for Ownership* (Appendix A), if one has not already been submitted. This statement indicates the responsibilities that the contributor must accept in order to make contributions to Coin-OR.

- *Contributor's Statement of Ownership and Licensing* (Appendix B), which lists the owners of any IP associated with the project and the specific license for the contribution. In the case of a Level 1 project that does not yet consist of any significant IP, this document should indicate potential owners of the IP to be generated and the license intended to be used.

- *Owner's Confirmation of Licensing* (Appendix B) or equivalent documentation from each legal owner.

These documents can either be submitted by email to `secretary@coin-or.org` and electronically "signed" or printed out and submitted in hard copy by fax or US mail to the secretary of the Foundation.

Determining Ownership. The creator of a work may not be the sole owner of the intellectual property associated with the work. In general, *any* individual or organization which contributed resources to the development of a work may be a co-owner. The legal ownership depends on the particulars of the situation and the contracts involved. Some employment contracts assert that the employer has ownership rights to any work created by the employee, even if that work is created outside of regular working hours and without the use of the organization's resources. Contributors should consult with their management, legal counsel, and/or technology transfer officers when determining the legal ownership of a contribution.

Acceptable Licenses. The second legal requirement is that the owners be willing to license the IP associated with the project under an appropriate license. Source code must be licensed under an OSI-certified license, *i.e.*, an open-source license approved by the Open Source Initiative (`http://www.opensource.org`). The Foundation encourages the use of the Common Public License (CPL), but any existing open-source license is acceptable. For contributions that are not software source code, if the contributor feels that no OSI-license is appropriate then alternatives will be considered on a case-by-case basis. The chosen license must *at minimum* allow the project to be freely redistributable with attribution and should be generally compatible with the open-source philosophy. A member of the TLC or SLB should be consulted to discuss the specific license requested and the contributor's reasons for choosing the license. Regardless of the license, every project must be distributed entirely under a single license. This rule applies to both regular and umbrella projects. Exceptions may be granted on a case-by-case basis.

5.3   Project Classification

For software projects, the main job of the project coordinator is to determine the project's initial classification according to the scheme described in Section 4. This scheme is clearly very subjective, so the initial classification decision may be appealed to the full TLC, who have final authority. Umbrella projects receive a single classification covering all subprojects. The submission requirements differ, depending on the current development level of the project when it is submitted and the initial classification the contributor hopes to attain.

Level 1. The main requirement for acceptance of projects initially classified at Level 1 is submission of a brief description of the project, including its goals and a strategy for eventually satisfying the requirements for a Level 2 project (provided the project is to be

active and has not been submitted to be archived). This description should be submitted along with any additional existing project files. All projects, regardless of level, must be submitted with at least the following files used to populate the project's repository initially.

README Contains the name of the project; a brief description of the project and its intended use; instructions for contacting the project maintainer; the address of the project web page; instructions for obtaining support, requesting features, and submitting bug reports. Dependencies on third-party software must be listed here. The file may contain other information, such as a brief revision history, comments from the author(s), *etc.*

LICENSE Contains a copy of the license agreement for the code. The license must be an OSI-license as described in §5.2.

In the absence of any other files, these will be used along with the description to help the Foundation determine whether the project is in scope and should be accepted.

Level 2. To be classified as a Level 2 project, the main additional requirement is that the code must work on at least on platform of interest. Code is deemed to work if it can be built without error using easily available tools from its distribution by following the instructions for installation. The code must also perform some useful function once it is built.

For Level 2 projects and above, the electronic files comprising the project should be contained in a well-organized directory tree with a single root. The root must contain four files with standard file names, including the README and LICENSE files described above, as well as the following additional files.

AUTHORS Contains a list of the author(s) of the code. The list should include the original author(s) and anyone who has made a significant contribution to the code.

INSTALL Contains the instructions for building, installing, and running the code on at least one platform. There are no specific requirements regarding the build procedure, though it is strongly recommended that the project manager use the procedures recommended by the TLC. Examples of common build and installation procedures are those utilizing the Unix `make` command (perhaps augmented by `autoconf`), or those depending on a commonly used integrated development environment, such as Microsoft Visual C++. The installation procedure must include information on installation of third-party software the code may depend on.

The complete contribution should be collected in a `tar` or `zip` archive for submission.

Level 3. Coin-OR strongly recommends that all projects be documented in some fashion, but projects classified as Level 3 or above are required, in addition to the above requirements, to have significant documentation, consisting of such things as user guides, developer manuals, and high-level summary information on the project's web page. Many existing projects automatically generate source code documentation using the `doxygen` system, which is well-suited for C and C++ code.

Level 4. Each project classified at Level 4 or above must satisfy all of the above requirements and include a substantial unit test. For Level 4 projects, Coin-OR requires

- documentation that describes what the tests do; and

- a simple methodology (or framework) for adding tests in the future.

The initial suite of tests can be relatively simple, with the main requirement being that (1) it can be run immediately after the build and installation procedure and (2) it clearly indicates success or failure of the installation. The unit test could consist of a separate executable that runs a series of tests, a shell script, or even just a sample input file with instructions on how to run it and verify the result. The testing facility can be minimal at first, but should be easy to build upon as the software is developed. Use of more advanced testing environments (*e.g.*, `CppUnit`, `DejaGNU`, or a thorough custom test) is encouraged.

Level 4 projects must also have a versioning system in place and a system of producing stable releases. It is highly recommended that all project managers adopt the procedures recommended by the TLC in this regard.

Level 5. To achieve Level 5 status, a project must satisfy all of the above requirements and distribute its software in the form of binaries built from the latest stable release on at least one platform.

5.4 Dependence on Third-Party Software

All software components of a project must be distributed under the same license. However, a project may depend on third-party software (open-source or not). Such dependencies must be listed in the `README` and in the `INSTALL` files. If the third-party software is available under an OSI-certified license, the contributor may make it available in the Coin-OR repository separate from the project itself.

6 Contribution to an Existing Project

Contributions to existing projects can only be made if the project is active and should be submitted directly to the project manager (or the subproject manager within an umbrella project). In the case of umbrella projects, if the contribution is related to one specific subproject, the decision to accept or reject the contribution rests with the relevant subproject manager who takes the role of project manager in the following three subsections. If the contribution is related to more than one subproject, the project manager should handle the contribution in consultation with the relevant subproject managers.

A contribution to an existing project can be either *trivial*, *significant*, or *substantial*. The distinction between the three is not easy to define precisely, but a basic guideline is that if

the contribution can be seen as adding non-trivial functionality, but does not raise IP issues, then it is significant. If an IP claim could reasonably be made related to the contribution, then it is substantial. All other contributions, such as very small bug fixes, typographical corrections or other obvious errors are trivial. In case there is doubt about the status of a contribution, the project manager should select substantial over significant and significant over trivial. The contribution of a new subproject to an umbrella project should always be considered substantial (see 6.4). The submission manager can help a project manager to categorize a contribution upon request. In all cases, a contribution to an existing project must be made under the same license as the rest of the project.

6.1 Trivial Contribution to an Existing Project

If a trivial contribution is accepted by the project manager, he or she can simply incorporate it into the repository without additional paperwork.

6.2 Significant Contribution to an Existing Project

If a significant contribution is accepted by the project manager, he or she must check with the Secretary to determine whether the contributor has previously filled out a Contributor's Statement of Respect for Ownership (CSRO; see Appendix A). If not, the contributor needs to submit a CSRO before the contribution can be added to the Coin-OR repository. Instructions for submitting a CSRO are listed in Section 5.2. The project manager must receive confirmation from the Secretary that the contributor has filed a CSRO before adding the contribution to the Coin-OR repository. The `AUTHORS` file of the project must be modified to record the contribution.

6.3 Substantial Contribution to an Existing Project

If a substantial contribution is accepted by the project manager, the contributor must provide the three documents listed in Section 5.2 for the contribution. The project manager must receive confirmation from the Secretary that the contributor has filed all documents related to the contribution before adding the contribution to the Coin-OR repository. The `AUTHORS` file of the project must be modified to record the contribution.

6.4 Contribution of a New Subproject to an Umbrella Project

The acceptance procedure for subprojects is the same as the acceptance procedure for new projects with the following alterations:

- The process will be supervised by and the final decision rendered by the project manager of the umbrella project.

- Subproject contributions must conform to the documented specifications/standards of the umbrella project.

- Subprojects should have the same license as the umbrella project. Exceptions may be considered at the discretion of the TLC.

- In handling a new subproject contribution, the umbrella project manager follows the steps in Section 5, except that the `README`, `AUTHORS`, `COPYING/LICENSE`, and `INSTALL` files may be omitted at the discretion of the project manager. If any of these files are not required, the corresponding information must be added to the appropriate files of the umbrella project.

Except for the above exemptions, subprojects should comply with all other requirements for a regular project.

7   Project Maintenance

There are no maintenance requirements for archived projects. This section applies only to active projects.

7.1   The Project Manager

7.1.1   Regular Project

The Coin-OR Foundation requires that every active project have a project manager. This individual is the main contact for the project and the liaison with the TLC. Once a contribution is accepted for inclusion in the Coin-OR repository, the project manager is granted write access to the project's repository and is expected to observe Coin-OR policy for maintaining the project, especially with respect to respect for intellectual property. The project manager must also subscribe to the Coin-OR project managers' mailing list, and must provide channels for users to report bugs, make suggestion, and request support.

The project manager is usually, but does not have to be, one of the primary authors of the code. The project manager may delegate ongoing maintenance, as long as the designated individual(s) has the knowledge to manage contributions to the project and is educated about intellectual property issues. Requests to grant write access to the repository for additional developers should be submitted to the repository manager.

7.1.2   Umbrella Projects

Umbrella projects should be actively managed by a single project manager (or possibly by multiple co-project managers), as with any other project. The project manager has the following duties with respect to the umbrella project:

- Setting technical specifications, such as the API for a given base class and the expected behavior of the derived class objects (that comprise the subprojects).

- Ensuring that subprojects comply with the aforementioned specifications.

The project manager will have write access to the project's entire repository directory hierarchy, including the subdirectories containing the subprojects and will have the right to modify and/or remove any subproject in the case of non-compliance with specifications.

A subproject manager is treated as a project manager for the corresponding subproject. The subproject manager thus has the prerogatives and responsibilities associated with project managers listed in the previous section.

7.2   Project Maintenance

Coin-OR will provide the infrastructure required for maintenance of a project's repository, including the files associated with the project itself and the project's static web pages, a bug reporting system, a project wiki, and a project mailing list. The repository is currently managed using `subversion`. Most projects will be maintained in the Coin-OR repository using the project management tools provided by Coin-OR. In some cases, projects developed hosted externally may be accepted for inclusion in the repository. In such cases, the procedures below may be modified in accordance to the requirements laid out in Section 7.3.

When a new project (or subproject) is accepted for inclusion as an active Coin-OR project, there is an explicit responsibility for its ongoing care and maintenance, including

- Maintenance and updating of files associated with the project or subproject, as appropriate (see Section 7.2.2).

- Maintenance of a project web page (see Section 7.2.4).

- Maintenance of channels for reporting bugs and providing other feedback (see Section 7.2.5).

7.2.1   Tracking Project Submissions and Authors

If the source material associated with a project is to be useful to others, especially in commercial applications, it is crucial to be able to determine the origin and authors of all contributions to the code. It is therefore strongly recommended that project managers carefully track the origin of all submissions by recording the information in the commit log message associated with any new contribution. Anyone who has made either significant or substantial contributions to a project should be listed in the AUTHORS file associated with the project. This file must be kept up to date at all times.

7.2.2   Code and Documentation

Documentation   All projects are expected to have an up-to-date README file that describes the project and its current state of development. For projects at Level 2 and above, the project manager must maintain minimal documentation in the form of an INSTALL file. Such projects are expected to build and run correctly on at least one platform and have an up-to-date INSTALL file. For projects at Level 3 and above, substantial documentation must be maintained.

Versioning and Releases   When and how to handle software releases for Level 4 and higher projects is left to the project manager. Coin-OR strongly recommends that each project maintain a `stable` branch in the project's subversion repository and that development take place in the `trunk` branch of the project. Dependencies on other software components should be documented. If possible, stable releases should depend on stable releases of other software.

7.2.3   Project Mailing Lists

By default, two mailing lists are associated with each project:

- `projectname@list.coin-or.org`

- `projectname-tickets@list.coin-or.org`

The first list (called the *main project list*) is for general discussion related to the project. The project managers and subproject managers are expected to monitor and moderate discussion in the main project list.

The second list (called the *project ticket list*) mirrors the tickets posted on the project management site (see Section 7.2.5).

If there is relatively high traffic on the main project list, the project manager may request creation of one or more of the following optional lists:

- `projectname-discuss@list.coin-or.org`

- `projectname-devel@list.coin-or.org`

- `projectname-announce@list.coin-or.org`

In addition, the project manager may wish to create a list to receive automated traffic regarding `subversion` commits and/or action on issue-tracking tickets. The expectation is that requests for such lists will be routinely granted, provided that the administrator believes the project manager understands the challenges and advantages of multiple lists.

At the project manager's discretion, the main project list may be deleted or forwarded to `projectname-discuss@list.coin-or.org`. Special requests for lists of other names will be handled on a case-by-case basis. Note that subprojects are not given stand-alone mailing lists—they share those of the umbrella project.

7.2.4   Project Web Page

By default, a project has a static web directory and a separate project management system site (currently Trac). Initially, the static web page contains only an automatic redirection to the project management site. The project manager may add content and modify either page and may choose which page will be the project web page. However, as the project management site gives facilities for code browsing and bug reporting, it is expected that the project management site will be made accessible to users. Links to web pages not hosted on Coin-OR servers are permitted.

For Level 2 projects and higher, the project web page or Trac wiki must provide pointers on how to download and install the project, a FAQ, and, for Level 4 projects and higher, the version number of the latest release of the project. Instructions for reporting bugs must also be provided. Coin-OR strongly recommends that both the main project mailing list and the bug tracking system of the project management site (see Section 7.2.5) be available for bug reporting.

New contributors are encouraged to adopt the style of one of the existing Coin-OR projects so that the look and feel of project web pages is relatively uniform across all projects in the repository. In the future, project managers will be provided with a template to make it as easy as possible for new projects to put up a web page on the Coin-OR web site. Note that subprojects are not given stand-alone web pages; they share the resources of the umbrella project. The project manager sets the requirement regarding web pages for the subprojects.

7.2.5   Bug Reporting Channels

Every project must maintain and advertise at least one channel for reporting bugs. If a project manager chooses to use the issue tracking system provided by the Coin-OR Foundation's project management system, the user can report bugs or suggest improvements to the project using *tickets*. The project (or subproject) managers are expected to monitor the posted tickets and deal with them or give some feedback in a timely manner. If a project manager chooses not to use the Trac ticket system as a bug reporting channel, this feature must be disabled on the Trac page and the desired bug reporting channel should be described on the project web page. An alternative bug reporting channel is to use the project's main mailing list. If a project manager chooses not to use the main project list as a bug reporting channel, then this must be specified explicitly on the project web page. The main project list (or equivalent) will still be available for discussions by users of the project.

Subprojects use the mailing lists and bug reporting channels of the umbrella project. Subproject managers are expected to monitor the mailing lists of the umbrella project, if this is

the policy of the umbrella project.

7.2.6   Periodic Review

The repository manager is responsible for ensuring that all projects in the repository are given a periodic status review, at which time the classification of the project may be changed. Project managers may also request a review for the purpose of changing a project's classification. If a project's classification is downgraded as the result of a periodic review, the project manager may appeal the decision to the full TLC. The decision of the TLC on the appeal is then final.

Only the TLC can decide to change an existing active project to an archived project. In such a case, the incumbent project manager has one month to implement any changes required to regain active status. After this time, anyone with interest in actively developing the project can petition the TLC to take over as project manager. Archived projects may be deleted from the repository by decision of the TLC. Before a project is deleted, the decision will be announced on the project mailing list and on the Coin-OR mailing list used for general announcements. The public announcement of the deletion of a project will be made at least one month before the project is actually deleted.

7.3   Remotely Hosted Projects

Coin-OR projects are generally expected to be hosted on servers managed by Coin-OR Foundation and to use standard Coin-OR project management tools, i.e., subversion repository, Trac project management wiki, mailing lists, and bug tracking. In some cases, a project may be hosted *remotely*, but still be managed as a Coin-OR project. In particular, remotely hosting a Coin-OR project may be justified when a software project has an established external development community, or when project hosting is restricted by external factors (e.g. business requirements for the company that is funding the project's development).

Remotely hosted projects are still expected to be managed according to the best practices listed in the preceding sections. This includes a requirement to maintain a project Trac wiki hosted on the COIN-OR servers containing information about the project. In addition, remotely hosted projects must mirror the content of the remote repository to a subversion repository maintained on the COIN-OR servers. The mirroring is not required to include the entire remote repository, but may simply be a mirroring of project releases. If the remote respository is itself a subversion repository, then the subversion mirroring facilities may be used. The TLC will consider whether the project can reasonably be maintained according to the standards detailed above when considering a proposal to integrate a remotely hosted project with Coin-OR.

A   Contributor's Statement of Respect for Ownership

Contributor's Statement of Respect for Ownership

I _____, certify that

(a) I have read and understood the statement below on Ownership of Intellectual Property;

(b) for any contribution I make to the Coin-OR Foundation repository, I will make all reasonable efforts to determine the legal owners of the contribution, and I will obtain the permission of the owners of the contribution to make the contribution available under an open source license certified by the Open Source Initiative;

(c) I will not knowingly submit any contribution of which I am not the owner or for which I do not have the owner's permission;

(d) for any contribution I make to an existing project, I will use the same license the project was released under.

_____
     (signature of contributor)

Ownership of Intellectual Property

The creator of a work may not be the sole owner of the intellectual property associated with the work. In general, *any* individual or organization which contributed resources to the development of a work may be a co-owner. The legal ownership depends on the particulars of the situation and the contracts involved. Some employment contracts assert that the employer has ownership rights to any work created by the employee, even if that work is created outside of regular working hours and without the use of the organization's resources. Also, an employer may be fine with contributions to one project but not to another.

Contributors should consult with their management, legal counsel, and/or technology transfer officers when determining the legal ownership of a contribution.

B   Documentation of Ownership and Licensing for New Projects

---

Contributor's Statement of Ownership and Licensing

I, _____, represent that:

(a) the individuals and organizations listed immediately below are the only owner(s) of the contribution _____;

(b) if any part of the contribution is not owned by the individuals and organizations listed in (a), that part was obtained under an open source license certified by the Open Source Initiative; and

(c) all owners have agreed to license the contribution under the _____, an open source license certified by the Open Source Initiative.

_____
(signature of contributor)

Owner's Confirmation of Licensing

I, _____, am an owner of the contribution _____, and I agree to license any portion I own under the terms of the _____, an open source license certified by the Open Source Initiative.

_____
(signature of owner)

C "Contribution Check List" For New Projects

1. Scope

   - Is the contribution of use for Operations Research professionals or students?

   - Is the project's goal to build, use, or otherwise support the development of open-source software?

2. Project Requirements

   - Based on all material submitted, what should be the initial classification level of the software? (See Section 5.3).

   - Basic Documentation

     – Does the submission include a `README` file? Does it contain the information requested in Section 5.3?
     – Does the submission include a `LICENSE` file? Does it contain the information requested in Section 5.3?
     – For a Level 1 project, did the submission contain a description and road map to achieve Level 2? (See Section 5.3).
     – If the contribution is Level 2 or above, does it include an `INSTALL` file? Does it contain the information requested in Section 5.3?
     – If the contribution is Level 2 or above, does it include an `AUTHORS` file? Does it contain the information requested in Section 5.3?

   - If the contribution is Level 2 or above, does the build and installation procedure described in the `INSTALL` file work?

   - For a Level 3 project and higher, is the documentation substantial and accurate?

   - For a Level 4 project or higher, is the unit testing environment adequate?

     – Is there documentation that describes the test(s)?
     – Is there a simple methodology for adding tests in the future?
     – Does the test run and indicate success?

   - For a Level 5 project, are there working binaries included?

3. Legal

   - Is the contribution being made under the CPL?

     – If not the CPL, is the license certified by the Open Source Initiative?
     – If not the CPL, is the contributor aware of the issues related to using their contribution with contributions licensed under the CPL?

   - Has a "Contributor's Statement of Respect for Ownership", a "Contributor's Statement of Ownership and Licensing", and an "Owner's Confirmation of Licensing" been filed with the Secretary for the contribution? (See Section 5.2.)

4. Project Manager

- Has a PM been named?

- Has the PM agreed to maintain the code on Coin-OR?

- Is the PM familiar with the Coin-OR process for accepting contributions? Has the PM agreed to follow the process for accepting contributions? (See Section 6.)

- Has the PM been subscribed to the PM mailing list?

- Has the PM been given write access to the repository?

- Has the PM filled out the standard summary page about the project (briefly describing purpose, robustness, build process, environment, *etc.* about the project)?

- Has the PM agreed to be responsive to bug reports? (See Section 7.2.5.)

- Has the PM been offered full membership in the Coin-OR Foundation?

- Has the PM been made aware of the release guidelines? (See Section 7.2.2.)

- Has the PM announced the project on the coin-announce mail list when the contribution process is complete and the project web page is live?

5. Project Infrastructure

- For Level 2 projects and above, does the project have a web page? Does it have an FAQ section, download and installation instructions, and, if appropriate, version number of the latest release ? (See Section 7.2.4.) Does it mention the mechanism for reporting bugs? If the ticket system is disabled on the Trac page, is this mentioned on the project web page? (See Section 7.2.5.)

- Has a project mailing list been established? (See Section 7.2.3.)

- Has the project's subversion repository and Trac project management sites been established?

- If the project is remotely hosted, have procedures been setup to manage the integration of content from the remote project repository?

- Have the project information database and general Coin-OR pages been updated to include project information?

- Has the project been announced on coin-discuss?