# Eclipse Project Proposal Tina Tool

* [Eclipse <New Name>](#scroll-bookmark-2)
  + [Parent Projects](#scroll-bookmark-3)
  + [Background](#scroll-bookmark-4)
  + [Scope](#scroll-bookmark-5)
  + [Description](#scroll-bookmark-6)
  + [Licenses](#scroll-bookmark-7)
  + [Legal Issues](#scroll-bookmark-8)
  + [Initial Contribution](#scroll-bookmark-9)
  + [Project Scheduling](#scroll-bookmark-10)
  + [Future Work](#scroll-bookmark-11)
* [Source Code](#scroll-bookmark-12)
  + [Source Repository Type](#scroll-bookmark-13)
* [People](#scroll-bookmark-14)
  + [Project Leads:](#scroll-bookmark-15)
  + [Committers](#scroll-bookmark-16)

This page will be used to prepare internally the Eclipse proposal: <https://projects.eclipse.org/create/project-proposal>

# Eclipse <New Name>

|  |
| --- |
| Please suggest a name for the open source project here |

|  |  |
| --- | --- |
|  |  |
| SW360 Tentacle |  |
| SW360 Cirrus |  |
| SW360 Toiler |  |
| SW360 Menial |  |
| SW360 Antenna |  |
| SW360 Patronage |  |
| SW360 Custody |  |
| SW360 Adiutor |  |
| SW360 Auxiliator |  |
| SW360 Colligentes |  |
| SW360 Factotum |  |
| SW360 Omnivore |  |

## Parent Projects

|  |
| --- |
| All projects (except top-level projects) are nested under an existing parent project. You can leave this field blank initially, but it must be filled in before the project can be created. Before specifying a parent project, you should be sure to communicate your intent with that project's leadership. Note that the parent project may be the same as the top-level project. |

SW360

## Background

|  |
| --- |
| Optionally provide the background that has lead you to creating this project. |

Open Source has arrived in the day to day software development of all sorts of organizations. However, to use Open Source software in legally compliant and save way, various precautions have to be taken and obligations have to be fullfiled. E.g. many open source licenses require to ship the license text with a software or name the copyright holders. Doing that in a manual way is tedious and error prone. Hence, tools like Eclipse SW360 are created.

Todays software development is on a high pace with almost fully automated build pipelines and continuous deployment scenarios. To properly provide software development projects with compliance related artifacts it is required to provide those artifacts in an automated way.

## Scope

|  |
| --- |
| All projects must have a well-defined scope. Describe, concisely, what is in-scope and (optionally) what is out-of-scope. A project cannot have an open-ended scope. |

The project covers tooling to generate compliance related artifacts (disclosure document, source code bundle, written offer etc) directly within a build process. It relies on data that is provided by different sources such as an SW360 instance.

## Description

|  |
| --- |
| *Describe the project here. Be concise, but provide enough information that somebody who doesn't already know very much about your project idea or domain has at least a fighting chance of understanding its purpose. Note that the content you provide here will be used as the initial description of the project once it has been created (you will be able to edit the description after the project has been created).* |

Tina is a tool to automate your open source license compliance processes as much as possible. In the end that is

* collecting all compliance relevant data
* process that data and warn if there might be any license compliance related issues
* generating a set of compliance artifacts (source code bundle, disclosure document, report)

for your project.

To reflect those three different types of tasks Tina is built arround a workflow engine, which allows to orchestrate a set of analyzers to gather required information, processors to arrange, adjust and evaluate that data, and a set of generators to produce a set of compliance related artifacts. Since licensing issues can deeply affect the success of your project it is required to be notified about any issues as early as possible in the development process. It is therefore useful to generate that information directly within your build. Tina can directly be integrated into the build process. This is realized with several so called frontends to build systems, which allow to invoke the tool and provide it with necessary configuration.

The Tina project is set up in a way that allows to easily create a custom configuration with a preconfigured set of shipped and custom analyzers, processors and generators to fit the needs of your internal compliance processes. E.g. You might use a commercial tool to analyze your dependencies and do not rely on the results of the maven dependency plugin. In that case you can provide a custom analyzer implementation, provide a custom configuration and bundle that as the tool, which can be used by your development teams to scan their projects.  
  
Why Here?

|  |
| --- |
| *What value does this project bring to our community? What value do you expect to obtain from hosting your project at the* centre *of this community using this forge?* |

Creating a project like TINA to supplement Eclipse SW360 is the natural next step. SW360 is capable of collecting Component information and making it available at a central place. The natural next step is to provide tooling to actually use that information in a build and by that further automate the compliance processes in organizations.

## Licenses

|  |
| --- |
| Under what license(s) will the project's artifacts be distributed? If you're not sure, contact the Eclipse Management Organization (EMO) for assistance. If the initial contribution is currently distributed under a different license, please make note of the current licensing terms in the "Legal Issues" section. |

The project will have the same licensing as the parent project, EPL-1.0. In case the parent project decides to relicense to EPL-2.0 we will relicense too.

## Legal Issues

|  |
| --- |
| Please describe any potential legal issues in this section. Does somebody else own the trademark to the project name? Is there some issue that prevents you from licensing the project under a supported license? Are parts of the code available under a potentially problematic license? |

The initial contribution is sole property of the Bosch Software Innovations GmbH. It has all rights to publish and distribute the code. The existing code is already prechecked and should not contain any dependencies that conflict with the Eclipse rules.

## Initial Contribution

|  |
| --- |
| Projects are expected to arrive with existing code. Describe the existing code that will be contributed to the project. Please provide a couple of paragraphs describing the code with modest detail, including important information like code ownership (who holds the copyright?), and some consideration of community that exists around the code. Include a listing of third-party libraries and associated licenses. |

The initial contribution will be the publically releasable part of an internal tool (TINA Tool) that is already in productive use for quite a while at Bosch Software Innovations GmbH. It consists of roughly 20 KLOC and is a Maven multi module project.

## Project Scheduling

|  |
| --- |
| Describe, in rough terms, what the basic scheduling of the project will be. You might, for example, include an indication of when an initial contribution should be expected, when your first build will be ready, etc. Exact dates are not required. |

* Q1/2018
  + provide initial contribution
  + legal check of initial contribution
  + setup public build infrastructure
* Q2/18
  + Setup project page
  + first milestone release
* Q3/2018
  + Further feature implementation

## Future Work

|  |
| --- |
| What functionality do you expect to add in the next twelve to eighteen months? What sorts of activities do you plan to undertake to grow the community around your project? |

Next steps

* Provide Gradle frontend
* Enhance SW360 connector

Community work

* present project at conferences (EC France, BITKOM, ...)
* setup mailing list
* Communicate availability in already established network arround Open Source Compliance Tooling

# Source Code

## Source Repository Type

|  |
| --- |
| *Specify the type of source code repository the project will use as its primary host. Note that some external hosting providers may be supported; if you do intend to use a supported external hosting service, identify the existing repositories in the "Source Repositories" field.*   * Git * GitHub   *If you have existing publicly-accessible source repositories for this project, list the URLs for them here. Note that if you intend to continue to use an external provider as the primary host for your source code, that provider must be one that is supported by the Eclipse Foundation. You must select that provider as the "Source Repository Type". If you need to provide further explanation regarding the existing host of your project code, please do so in the "Initial Contribution" field.*   * <url> |

**github**

# People

|  |
| --- |
| A proposal must have one or more project leads. The project lead is more of a position of responsibility than one of power. The project lead is immediately responsible for the overall well-being of the project. They own and manage the project's development process, coordinate development, facilitate discussion amongst project committers, ensure that the Eclipse IP policy is being observed by the project and more. |

## Project Leads:

* Johannes Kristan
* Michael C. Jaeger ?

## Committers

1. Johannes Kristan
2. Maximilian Huber
3. Michael Jaeger
4. Tobias Rawald
5. Onur Demirci