IP Management and Software Licensing for the Eclipse Project mdmbl

openMDM(R) Eclipse Working Group

Document history:

Author	Date	Description
Angelika Wittek	07.03.2018	initial version, draft
Angelika Wittek	08.08.2018	Addition to copyright header

The source of this document is located in Google Docs:
https://docs.google.com/document/d/1ID6Y7pghOwfAHpf9wK3mFOqm0HnWB58272s4VFGoTZw/edit?usp=sharing

Table of Contents

1 Introduction

Developing Open Source Software (OSS) is great. Licensing and using software under an OSS License brings rights and obligations with it. The informations in this document are related to license software under the Eclipse Public License 1.0 (EPL 1.0). We will give an brief introduction to the main points.

The EPL 1.0 is the mandatory reference in all points:
https://www.eclipse.org/legal/epl-v10.html

1.1 Why using Eclipse IP Process?

Many organizations take a very naive look at open source software. Engineers are using downloads from the Internet and integrate them into the company's products. They are not aware that they may violate someone's ownership rights, and they may not be aware that they endanger the company's assets or create a solution or product that has become dangerous to the product. In the past 10 years, large organizations like IBM, Oracle, BMW, Daimler, Bosch, SAP, Siemens and many others have taken a serious look at it and have established rules that help the engineers to manage OSS that is used in their products and solutions as well as contributions these companies make to OSS projects.
All these organizations are appreciating the services that the Eclipse Foundation provides. Inbound OSS that has undergone the Eclipse IP Process is easily accepted for usage. And many of these organizations create or facilitate creation of OSS at the Eclipse Foundation for exactly these reasons.
In order to become a mature OSS user the openMDM(R) Eclipse Working Group consumer companies have very early decided that they want the same type of protection that the above mentioned companies enjoy.

Developing OSS software brings along fallacies and pitfalls. The Eclipse Process supports you not to trap in. Furthermore Eclipse provides infrastructure for the whole lifecycle of your software and according documentation and more.
See: https://www.eclipse.org/projects/handbook/

Especially look at the IP Management at the Eclipse Foundation
"The Eclipse Foundation takes a very rigorous approach to intellectual property management. As far as I know, we are the only open source foundation to have a dedicated staff whose sole responsibility is the review all of the code distributed by Eclipse Foundation projects. The easy part is the code that is written by Eclipse committers. The far more time-consuming piece is a detailed review of all of the dependencies used by or distributed with Eclipse projects. That dependency review includes checking license compatibility, scanning the code to look for potential issues, and checking in on the provenance of the code in question. That last piece ("provenance"), can be particularly time-consuming because it involves answering questions like "was the code ever re-licensed from licenseA to licenseB, and if so how was permission obtained from the contributors?". Or "how does this

project manage contributions to it?" To my knowledge, no other open source foundations or communities do the level of detailed analysis that we do."
[From Mike Milinkovich, Executive Director of the Eclipse Foundation, https://mmilinkov.wordpress.com/2016/06/29/overhauling-ip-management-at-the-eclipse-foundation/]

1.2 EPL 1.0 vs EPL 2.0

The EPL 1.0 was designed in the late part of the 1990's. At that time IBM had certain requirements that it tried to accomplish. While time went by the Eclipse Foundation team learned that there were certain shortcomings that prohibited the EPL 1.0 to become one of the dominant licenses in the open source world. Here are some of the relevant changes

- A GPL compatibility clause has been added.
- The references to the law of New York state have been removed, and definitions for Source Code, Executable Code, Derivative Works and Modified Works have been included.
- The warranty and liability sections have been changed and do now honor local applicable law (e.g., European law).

Especially the latter 2 changes shall make it easier to use the EPL 2.0 in Germany and Europe. For more information, please refer to the license and the FAQ.

2 Licensing Software under the EPL 1.0

Important points for licensing software under the EPL 1.0:

- Comply to the rules of 3rd party library licenses including compatibility to EPL 1.0
- Provide legal documentation
    - Provide EPL 1.0 copyright header
    - Provide the EPL 1.0 Licence File
    - Provide a notice file
    - Provide End User Content

Note: Licensing software under the EPL 1.0 does not mean, that the code has to be hosted at Eclipse. You can host it in any repository, e.g. www.github.com.

Explained in detail:

2.1 Comply to the rules of 3rd party library licenses

There are two main points:

- declare the usage of the 3rd party libraries as described in the according licenses
- check compatibility to the EPL 1.0

See: https://www.eclipse.org/projects/handbook/#ip-third-party

2.2 Provide /extend EPL 1.0 copyright headers

See: https://www.eclipse.org/projects/handbook/#ip-copyright-headers

Important notes:

- **Depending on your committer assignment and your working contract you either add you name OR your company name. NOT both!**
- **You never delete somebody from the copyright header!**

For a **new file** add:

```
/*
 * Copyright © {date} {owner}[ and others]
 * All rights reserved. This program and the accompanying materials
 * are made available under the terms of the Eclipse Public License v1.0
 * which accompanies this distribution, and is available at
 * http://www.eclipse.org/legal/epl-v10.html
 */
```

If you **modify a file** you have to change the copyright header accordingly:
E.g. existing header:

```
Copyright © 2017 John Doe
```

Minor changes: Update the year and add “and others”:

```
Copyright © 2017-2018 John Doe and others
```

Major changes: Add a new line

```
Copyright © 2017 John Doe
Copyright © 2018 Lisa Mustermann
```

2.3 Provide the Licence file

- provide the license file in every repo at root level.
- provide the license file in every distribution you build and deliver (zip, war, jar, ...)

See:

- https://www.eclipse.org/projects/handbook/#legaldoc-license
- https://www.eclipse.org/projects/handbook/#legaldoc-repo

2.4 Provide a Notice File

- provide the notice file in every repo at root level.
- provide the notice file in every distribution you build and deliver (zip, war, jar, ...)

See:

- https://www.eclipse.org/projects/handbook/#legaldoc-notice
- https://www.eclipse.org/projects/handbook/#legaldoc-distribution

2.5 Provide End User Content

See: https://www.eclipse.org/projects/handbook/#legaldoc-end-user

3 The Eclipse Foundation IP Process

The Eclipse IP Process supports you that your Eclipse project is compliant to the EPL
That means, you have to follow the process. The IP Process is mainly an interaction between the Eclipse IP Team (and its proxies and tools) and the committers of a project. To keep things transparent, the communication in the IP Process is mainly via email and the Eclipse IP tracking tool IP ZIlla.The main points are discussed here, also refer to:
https://www.eclipse.org/projects/handbook/

Main points:

- committers have to follow the committer due diligence guidelines
- regularly run through an IP Check for your provided software
- make an IP Check for 3rd party libraries you want to use BEFORE you check them into the repositories
- regularly run through an IP Check for 3rd party libraries used for test & build

- Provide legal documentation
    - Provide copyright notice (provenance)
    - provide a EPL 1.0 license file
    - provide a Notice File and keep it up to date
    - provide End User content

The Eclipse Legal Process Poster:
https://www.eclipse.org/legal/EclipseLegalProcessPoster.pdf

The Eclipse IP Process in Eight Cartoons:
https://www.eclipse.org/projects/dev_process/ip-process-in-cartoons.php

Explained in detail:

3.1 The committer due diligence guidelines

Every committer has signed the committer paperwork:
https://www.eclipse.org/projects/handbook/#paperwork

That includes that **a committer has to follow the due diligence guidelines** ("Sorgfaltspflicht"):
https://www.eclipse.org/legal/committerguidelines.php

3.2 IP Checks, CQs, Type A / Type B and reuse

3.2.1 Explanation Type A / Type B IP Checks:

- Type A: License Compatibility Certification
- Type B: Full IP Due Diligence (License, Provenance, Scanning)
- Exact defintion: https://www.eclipse.org/org/documents/Eclipse_IP_Policy.pdf

For every IP Check you have to create a Contribution Questionnaire (CQ) in the Eclipse IPZilla Tool.

For every CQ you can decide if a Type A or a Type B check is performed. As long as the Eclipse projects are in the incubation phase, IP Checks of Type A are sufficient.
From experience Type A checks are performed must faster at Eclipse then Type B checks.

For 3rd party libraries reuse CQs where possible. That means if you want to use a library with version x and there is already a CQ from another project for exact this library you can create an Piggyback CQ. The advantage is that these CQs are performed automatically and run trough very fast (in general about 1 hour).

See also these blog posts:
https://waynebeaton.wordpress.com/2017/01/16/eclipse-infrastructure-support-for-ip-due-due-diligence-type/
https://mmilinkov.wordpress.com/2016/06/29/overhauling-ip-management-at-the-eclipse-foundation/

3.2.2 Getting started with CQs (Links)

This section is for commiters of the Eclipse project, only they have acces to the CQ Pages.

1. Login to IP-Zilla: https://dev.eclipse.org/ipzilla/
   Via the search button you can define filters for your project or look for existing CQs
2. Create a new CQ:
   a. Go to yout project page and log in, e.g. https://projects.eclipse.org/projects/technology.openk-platform/
   b. On the right side you have the committer tools, choose “Create a Contribution Questionnai…”

3.3 IP Checks for Eclipse projects

Regularly run through an IP Check for your provided software.

For Eclipse Projects IP Checks have to be provided from the current dev team (committers) for every release and /or requested by the Steering Committee.

As long as the projects are in the incubation phase, IP Checks of Type A are sufficient.

Regularly accomplished IP Checks minimize the risk of unclear state of the Terms of Use.

3.4 IP Checks for 3rd party libraries used for productive code

- run through an IP Check for every 3rd party library you want to use BEFORE you check them into the repositories and / or reference them via your build scripts
- As long as the projects are in the incubation phase, IP Checks of Type A are sufficient.

Note: this is not only about Java libraries, it also applies for all other libraries you use, e.g. JavaScript, Python, …

The IP Team must review third party content if:

- *the Java/OSGi manifest for one of the project bundles makes a direct reference to third party content (either a bundle or package);*
- *project code includes an import statement for a package from third party content;*
- *project code uses reflection or other means to reference APIs and implementation;*
- *project code uses OSGi Services to make a reference to a specific implementation of a service; or*
- *project code invokes a "command line" tool.*

This list is not intended to be exhaustive.

From: https://www.eclipse.org/projects/handbook/#ip-third-party

https://blogs.eclipse.org/post/wayne-beaton/decoding-eclipse-ip-policy-third-party-dependencies

3.5 IP Checks for 3rd party libraries used for test & build only

See: https://wiki.eclipse.org/Development_Resources/IP/Test_and_Build_Dependencies

3.6 Provide legal documentation

See: https://www.eclipse.org/projects/handbook/#legaldoc

That includes:

- Provide Copyright Headers, see 2.1 Provide EPL 1.0 copyright headers
- Provide the EPL 1.0 Licence File, see 2.2 Provide the Licence file
- Provide a notice file, see 2.4 Provide a Notice File
- Provide End User Content, see 2.5 Provide End User Content