

Guidelines for external contributions for the Eclipse mdmbl project

openMDM(R) Eclipse Working Group

Document history:

Author	Date	Version	Description
Angelika Wittek	01.05.2018	0.1	initial version
Angelika Wittek	02.05.2018	0.2	Review comments from S. Ebeling added
Angelika Wittek	02.05.2018	0.3	Review comments from S. Wartini added
Angelika Wittek	02.05.2018	1.0	Initial official version

Table of contents

1 Introduction	3
2 External contributions to the mdmbl codebase	3
2.1 Organizational Steps	3
2.2 Technical steps	4
2.3 Rejection of an external contribution	4

This document is published under the Eclipse Public License 2.0:

<https://www.eclipse.org/legal/epl-v20.html>

Note: this document is written in Google Docs, location:

<https://docs.google.com/document/d/1cFiYue7bYxLp5-rXHNj79xQWDne16mOESYNy9r5blgQ/edit?usp=sharing>

1 Introduction

The last months have shown that there are several code extensions based on the Eclipse mdmbl code that should to be contributed to the mdmbl codebase. In this document we describe a way how these contributions can be distributed to the codebase.

The Architecture / Quality Committee (AC/QC) decided for this approach, it will be extended and modified over time.

For every contribution these guidelines should be followed, in every case the AC/QC will support the contributors and specify and/or modify the guidelines if appropriate.

For upcoming external contributions the AC/QC will propose guidelines how external contributions can be planned, discussed and accepted from the beginning together with the customer and the openMDM(R) EWG.

Please note: Modifying the mdmbl code and transferring to other parties breaks the EPL. The code has to be put under the EPL and has to be available for the public.

2 External contributions to the mdmbl codebase

The decision about an external contribution becoming a part of the mdmbl codebase consists of two parts. First is the process - the openMDM(R) EWG has to decide if the contribution contains extensions / features that are valuable for the codebase and have to be added to the mdmbl project. The second part is the technical part and consists of the code quality checks and the way of merging it into the codebase.

In the openMDM(R) EWG the development of external contributions are normally assigned from a customer to a contractor (in this case the contributor), we define these two roles here for the following steps.

2.1 Organizational Steps

1. The contributor and/or the customer provide a description of the contribution (business aspects and technical overview including API changes)
2. The AC validates the technical and functional content and provides an actionable proposal to the SC.
 - a. AC needs to take special care if the new functionality works for existing subsystems and components as well as known products (Example: PAK Adapter).
 - b. The AC shall approve such contributions unless severe reasons speak against approval.
3. The SC approves or rejects the task for contribution, in which case the next step will not be started.

4. In case of approval, SC advises the Toolkit Manager (TKM) to continue with the technical steps listed below and approves cost / expenses for the TKM and the development team.

2.2 Technical steps

1. The TKM, the AC/QC and the contractor clarify the contribution rules (IP management, test coverage, code style, code comments, Key Performance Indicator (KPI) of static code analysis, ...).
2. The contributor checks the code in accordance to the rules and applies changes as needed.
3. The contributor provides information and documentation for the contribution (e.g.API Changes, text for the release notes, ...).
4. The contributor merges a defined code version into their contribution. The TKM defines the version of the mdmbl code.
5. Test cases of the mdmbl code as well as the contributed code shall execute without errors.
6. The dev team creates a new branch for the contribution in the Eclipse Infrastructure ("new_branch").
7. The contributor contributes the code to the "new_branch" via Gerrit - no committer rights are necessary, only a valid Contributor License Agreement (CLA) is required:
 1. The dev team reviews the code.
 2. If there are review comments, the contributor applies changes as needed.
8. If the contribution contains more than 1000 lines of new/changed code an IP Check from Eclipse is necessary and will be started and monitored by the TKM.
 1. If IP issues come up the contributor applies changes as needed.
9. All checks complete: a committer from the dev team approves the contribution in Gerrit. This causes the push of the contribution to the "new_branch".
10. The TKM and dev team merge the new branch into the main mdmbl code base at their discretion.
11. Final step: The contribution will be included in the next version of the mdmbl code.

2.3 Rejection of an external contribution

If the request for an external contribution to the mdmbl project codebase is rejected, the contributor and the customer are responsible on their own to follow the EPL. E.g. the extended / modified mdmbl code has to be licensed under the EPL and made available to the open.

For all licensing issues follow the EPL 1.0:

<https://www.eclipse.org/legal/epl-v10.html>