

Installation and Setup Guide

for the openMDM(R) Application

Eclipse mdmbl project

Document history:

Author	Date	Affects Version	Description
Angelika Wittek, Alexander Nehmer; Matthias Koller	9.6.2017-14.5.2019	<5.0.0	initial version User Preference Service added Review and additions Mailing Lists and ECA infos added Comments from Ganesh inserted, Glassfish Bugs added Elasticsearch version added, exported to pdf for mdmbl page Added Eclipse Hot Deploy section Parameter to skip npm install additions for new version exported to pdf for download pages Added SonarQube setup in Eclipse Extending the introduction chapter changes in service.xml, restructuring chapters, exported for V0.9 changes for version V0.10 IP Management Chapter added Chapter 6 extended
Angelika Wittek, Matthias Koller	14.05.2018-20.2.2020	5.0.0++	Updates for new version
Angelika Wittek, Matthias Koller	09.03.2020	5.1.0	Updates for new version merged Setup Guide an Installation Guide to this document.
Angelika Wittek, Simon Skoczylas	27.04.2020	5.2.0M1	Small fixes from Simon's review Added updates for database setup and roles.

Table of contents

1. Introduction	5
1.1. General	5
1.2. Requirements and Bugs	5
1.3. Use of new Frameworks / IP Management	6
1.4. Branching and versioning	6
1.5. Eclipse Infrastructure	7
1.5.1. Gerrit	7
1.5.2. Jenkins	7
1.5.3. Sonar	7
1.6. ODS Server used for Developer Tests	7
2. Prerequisites	8
2.1. Gradle	8
2.2. Eclipse IDE	8
2.2.1. Install Eclipse Copyright Tool	9
2.2.2. Use Eclipse copyright tool	10
3. Get and build the code	11
3.1. Changes	11
3.1.1. Since 5.2.0M1	11
3.2. Source Code Repositories	11
3.2.1. Manually checkout	11
3.2.2. Checkout in Eclipse IDE	12
3.3. Importing projects to the Eclipse IDE	13
3.4. Building the projects	14
3.4.1. Building projects separately	14
3.4.2. Building all projects for the core application	14
3.4.3. Building the projects without web frontend	15
3.5. Configure Git and Gerrit	15
3.5.1. Configure repositories	16
3.5.2. Configure Push	16
3.5.3. Commits to Gerrit	16
4. Preconditions and required installations	17
4.1. Glassfish	17
4.2. Database for the User Preference Service	17
4.2.1. Apache Derby Database	18
4.2.2. Other Database Products	19
4.2.3. Configure JDBC for the User Preference Service DB	19
4.3. ElasticSearch	20
4.4. Database for ODS-Server	20
4.5. ODS Server	20
5. Install, deploy and configure application	21
5.1. Configuration Files	21

5.2. Configure Login Module (Realm Configuration)	22
5.3. Property lookups for parameters values in service.xml	23
5.4. Configure system properties	23
5.5. Deployment of org.eclipse.mdm.nucleus	23
5.6. Headless Deployment	23
6. Start application	24
6.1. Troubleshooting	24
7. Known (setup) problems and solutions	24
7.1. Build	24
7.2. Glassfish	24
7.2.1. Glassfish - Inconsistent Module State	24
7.2.2. Glassfish - WEB9102: Login failed: Lookup failed	25
7.2.3. Glassfish 4 only - java.lang.ClassNotFoundException	25
7.2.4. Glassfish 4 only - org.osgi.framework.BundleException	25
7.2.5. Glassfish 4 only - java.lang.NoSuchMethodError	26
8. Extended configurations (run & IDE)	26
8.1. Deployment via Eclipse IDE	26
8.2. Angular Live Development Server	27
8.3. SonarQube	27
9. Development Rules	28
9.1. Commit comments	28
9.2. Tasks in the code	28
9.3. Working with committer feature branches	29
10. Tipps+Tricks	29
10.1. Handling of Relations	29
11. APPENDIX	30
11.1. Web Frontend: configure icons	30
11.2. Web Frontend: I18N configuration	31

This document is published under the Eclipse Public License 2.0:

<https://www.eclipse.org/legal/epl-2.0/>

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

<https://docs.google.com/document/d/1fs5p-UQceDV4dJeSAUWughCt-c1Cc82uhSxg8BRKsQs/edit?usp=sharing>

1. Introduction

1.1. General

This document serves as a setup and installation guide for the Eclipse mdmbl project. For **developers** it describes how to setup the environment, where the source code of the project can be found, retrieved, build and deployed.

For **users** interested only in deploying / installing the openMDM Application, start [here](#).

Note: This guide describes the deployment on a local machine only, for installing the application in a company infrastructure, please contact your administrators to support you with firewalls and proxy configurations, grant permissions, etc..

Mailinglists:

- For development communication we use the mdmbl mailing list:
<https://dev.eclipse.org/mailman/listinfo/mdmbl-dev>
- The openMDM Working group mailing list:
<https://dev.eclipse.org/mailman/listinfo/open-measured-data-wg>

For contributing to the mdmbl project you need to sign the ECA
(Eclipse Contributor Agreement) : <https://wiki.eclipse.org/ECA>

Helpful Links:

- Eclipse Wiki - openMDM EWG:
<https://wiki.eclipse.org/Open-Measured-Data-Management-WG>
- Eclipse Project openMDM@BL:
<https://projects.eclipse.org/projects/technology.mdmbl>
- Eclipse Bugzilla mdmbl issues:
https://bugs.eclipse.org/bugs/buglist.cgi?quicksearch=openmdm&list_id=16180271
- openMDM git repos: <http://git.eclipse.org/c/?q=mdm>

Downloads:

Get the artefacts from the Project Download Page: openMDM_application-<version>.zip
<https://projects.eclipse.org/projects/technology.mdmbl/downloads>

1.2. Requirements and Bugs

Requirements:

- Requirements are created in the Eclipse Bugzilla System [1]
- Requirements are ordered by priority and maturity by the Steering Committee
- for every REQU issue one or more Eclipse Bugzilla Tasks are created
- mdmbl dev team works on the Bugzilla issues[1]
<https://bugs.eclipse.org/bugs/buglist.cgi?quicksearch=mdmbl>

Bugzilla issues:

- Requirements
- “real” bugs from internal or external
- technical requirements / enhancements from the team

1.3. Use of new Frameworks / IP Management

To introduce a new framework to the code the QA guidelines of the EWG and the Legal Process of Eclipse (<https://www.eclipse.org/legal/EclipseLegalProcessPoster.pdf>) has to be followed:

- Get the approval for use from the the Architecture Committee. The members of the AC are listed here:
<https://wiki.eclipse.org/Open-Measured-Data-Management-WG>.
- Get the IP approval from the Eclipse Foundation, see:
[OpenMDM_IP_management_and_Software_Licensing.pdf](#)

Only if you have both approvals, you can use the libraries in the code!

1.4. Branching and versioning

Version number system changes after version 0.10. Version 0.11 was renamed in 5.0.0M1 (Version 5.0.0, Milestone 1).

This affects only the following projects:

1. org.eclipse.mdm.api.base
2. org.eclipse.mdm.api.default
3. org.eclipse.mdm.api.odsadapter
4. org.eclipse.mdm.nucleus

Stable versions are on the **master** branches.

The current development happens on the **dev** branches.

There are also several other branches, please contact us via the mailing list if you need more information.

The latest stable version is on the current master. Older stable versions are tagged with the version number, e.g.5.1.0M7, 5.1.0M8.

For all stable versions the build artefacts and documentation are made available on the mdmbl download page:

<https://projects.eclipse.org/projects/technology.mdmbl/downloads>

Documentation and release notes in Git:

<https://git.eclipse.org/c/mdmbl/org.eclipse.mdm.nucleus.git/tree/>

1.5. Eclipse Infrastructure

1.5.1. Gerrit

The mdmbl project is using Gerrit for code reviews. To configure, refer to chapter [Configure Gerrit](#).

Gerrit has a two stage reviewing system:

1. when code is checked into Gerrit, a build is automatically started. The Gerrit flag “verified” is set to +1, if the build succeeded
2. a committer has to review the code and if ok, the code review flag is set to +2

1.5.2. Jenkins

<https://ci.eclipse.org/mdmbl/>

There are nightly builds for the dev and the master branches of the following repositories:

1. org.eclipse.mdm.api.base
2. org.eclipse.mdm.api.default
3. org.eclipse.mdm.api.odsadapter
4. org.eclipse.mdm.nucleus

There are also builds for Gerrit and Sonar.

1.5.3. Sonar

The Eclipse SonarQube server is available via <https://sonar.eclipse.org>.

Rules: Qualitygate - sonar way with findbugs

Static code analysis is configured for the dev branches of the following projects:

1. org.eclipse.mdm.api.base
2. org.eclipse.mdm.api.default
3. org.eclipse.mdm.api.odsadapter
4. org.eclipse.mdm.nucleus

Generated and test sources are excluded.

1.6. ODS Server used for Developer Tests

All released versions are tested with the following ODS Server:

- HiQSoft GmbH
<https://www.highqsoft.com/de/avalon-asam-ods-server/>
- Peak Solution GmbH
<http://www.peak-solution.de/de/produkte-leistungen/versuchs-messdatenmanagement/softwareloesungen/peak-ods-server/>

2. Prerequisites

Already installed on your machine or install it:

- Java 8:
 - Oracle Java SE 8u201 and higher
<http://www.oracle.com/technetwork/java/javase/downloads/>
 - AdoptJDK: openJDK 8 with HotSpot
<https://adoptopenjdk.net/>
- Git: <https://git-scm.com>

Test, if programs are available from a console:

- > java -version
- > git --version

If not, add the programs to your PATH variable or set corresponding *_HOME variables.

2.1. Gradle

See <https://gradle.org/install/> for installation on you system.

Version: 4.10.2 (updated 21.2.2020)

This is optional, as the projects are using the Gradle Wrapper which is delivered with the source code.

2.2. Eclipse IDE

Get the current “Eclipse IDE for Enterprise Java Developers” Distribution from <https://www.eclipse.org/downloads>

It is recommended to use the Eclipse Installer.

Note: installation is tested with Eclipse 2019/03, Eclipse 2019/09

Install Plugins:

- SonarLint via Eclipse Marketplace
- EclEmma Code Coverage Tool (optional)

Create a new workspace for the mdmbl projects and configure it:

- set encoding to UTF-8
- set Java Code Style formatting rules to the Eclipse default rules [built in]

Formatter:

We highly recommend to use the Eclipse 2019/03 or higher IDE. Set the formatting rules to the Eclipse default rules [built in].

If you use a former version, then import the formatting rules from:

org.eclipse.mdm.nucleus/eclipse_formatter.xml

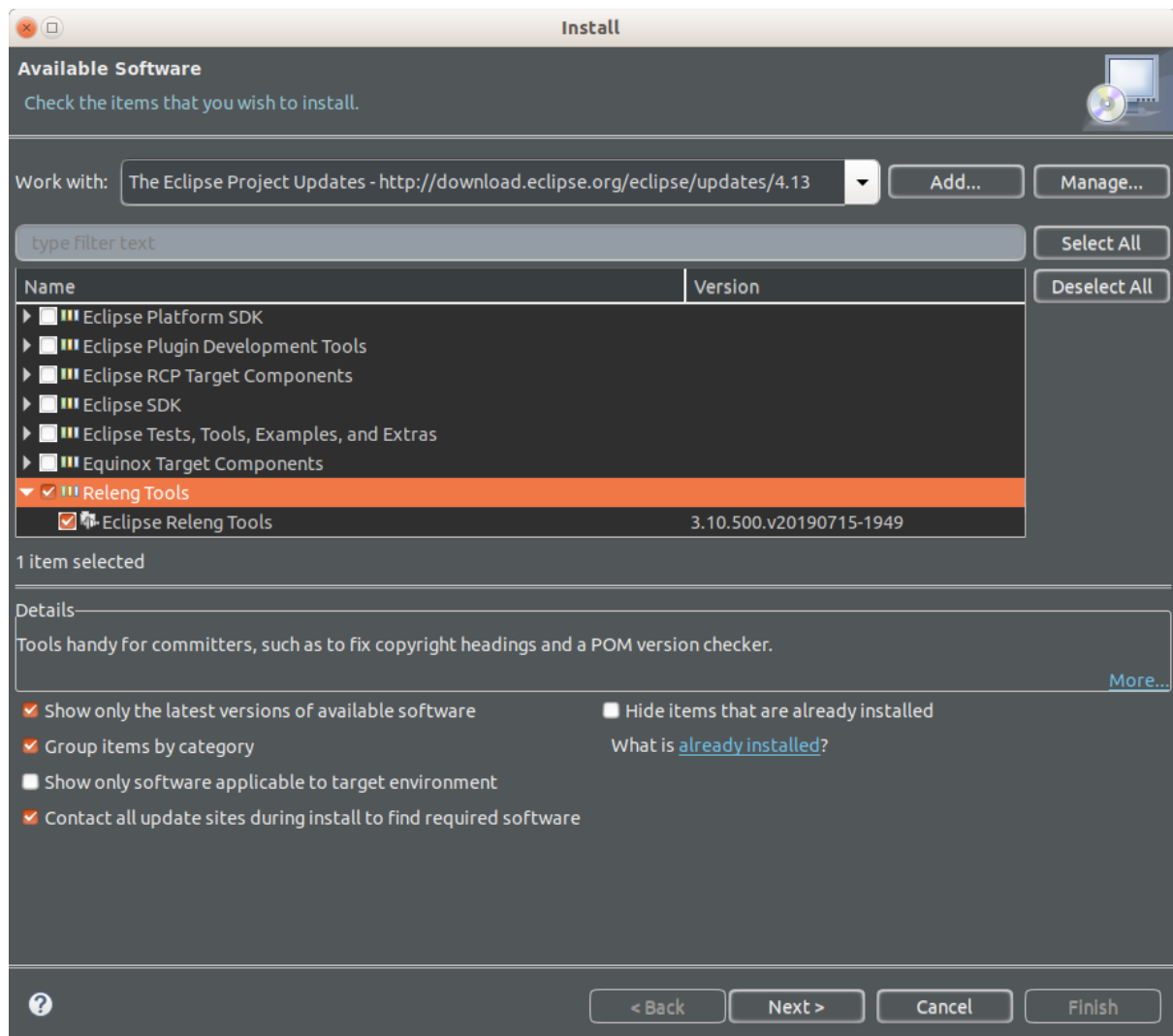
THIS IS IMPORTANT as the default rules in Eclipse have changed with the releases.

2.2.1. Install Eclipse Copyright Tool

For proper maintenance of the copyright header install this tool and run it before pushing to Gerrit.

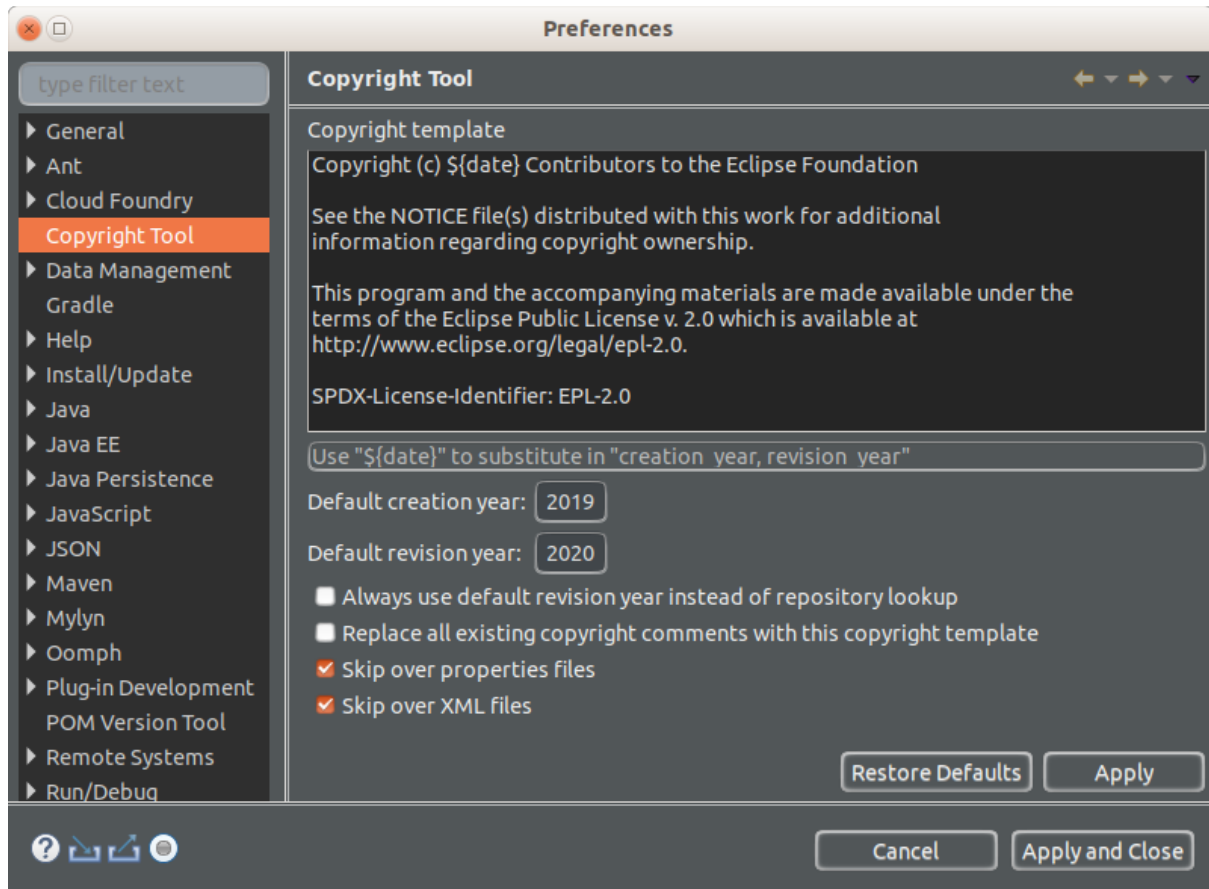
Help > Install New Software...

Choose the appropriate Update Site for your Eclipse version -> Choose Releng Tools



Configuration of Eclipse copyright tool

The Copyright Tool is configured using the preferences.



Template text:

Copyright (c) \${date} Contributors to the Eclipse Foundation

See the NOTICE file(s) distributed with this work for additional information regarding copyright ownership.

This program and the accompanying materials are made available under the terms of the Eclipse Public License v. 2.0 which is available at <http://www.eclipse.org/legal/epl-2.0>.

SPDX-License-Identifier: EPL-2.0

2.2.2. Use Eclipse copyright tool

Once installed and configured, the Copyright tool can be accessed via context menu on a project -> Fix copyright

3. Get and build the code

3.1. Changes

3.1.1. Since 5.2.0M1

Changes to MDMRealm:

Up to version 5.1.0 all Users had to be assigned the group `MDM` to use openMDM. Since 5.2.0M1 users need one of the following groups: `Admin`, `DescriptiveDataAuthor`, `Guest`.
See `org.eclipse.mdm.nuclues/readme.md#Configure LoginModule` for more information.

Database schema change

In this release openMDM 5 database schema was extended.
If you have an existing installation make sure to apply the provided update script to your database.
The script can be found in `schema/update/{DATABASE}/5.2.0M1_update.sql` or see [here](#).

3.2. Source Code Repositories

The application is split into several modules, each one dedicated to a certain scope of functions. All source code has been made available in git repositories hosted by the Eclipse Foundation <https://eclipse.org/org/foundation/>
The version control system used to track all changes is Git: <https://git-scm.com/>

We strongly recommend to use a separate workspace for the openMDM projects.

3.2.1. Manually checkout

- Change to the workspace directory
- Run the following command to checkout the source code:
`git clone <repository URL>`

Checkout the following repositories for the openMDM Application:

```
git clone http://git.eclipse.org/gitroot/mdmbl/org.eclipse.mdm.api.base.git
git clone http://git.eclipse.org/gitroot/mdmbl/org.eclipse.mdm.api.default.git
git clone http://git.eclipse.org/gitroot/mdmbl/org.eclipse.mdm.api.odsadapter.git
git clone http://git.eclipse.org/gitroot/mdmbl/org.eclipse.mdm.nucleus.git
```

There are additional openMDM tools:

```
git clone http://git.eclipse.org/gitroot/mdmbl/org.eclipse.mdm.openatfx.mdf.git
git clone http://git.eclipse.org/gitroot/mdmbl/org.eclipse.mdm.mdfsoriter.git
```

3.2.2. Checkout in Eclipse IDE

Alternatively configure eGit in your Eclipse IDE and checkout there.

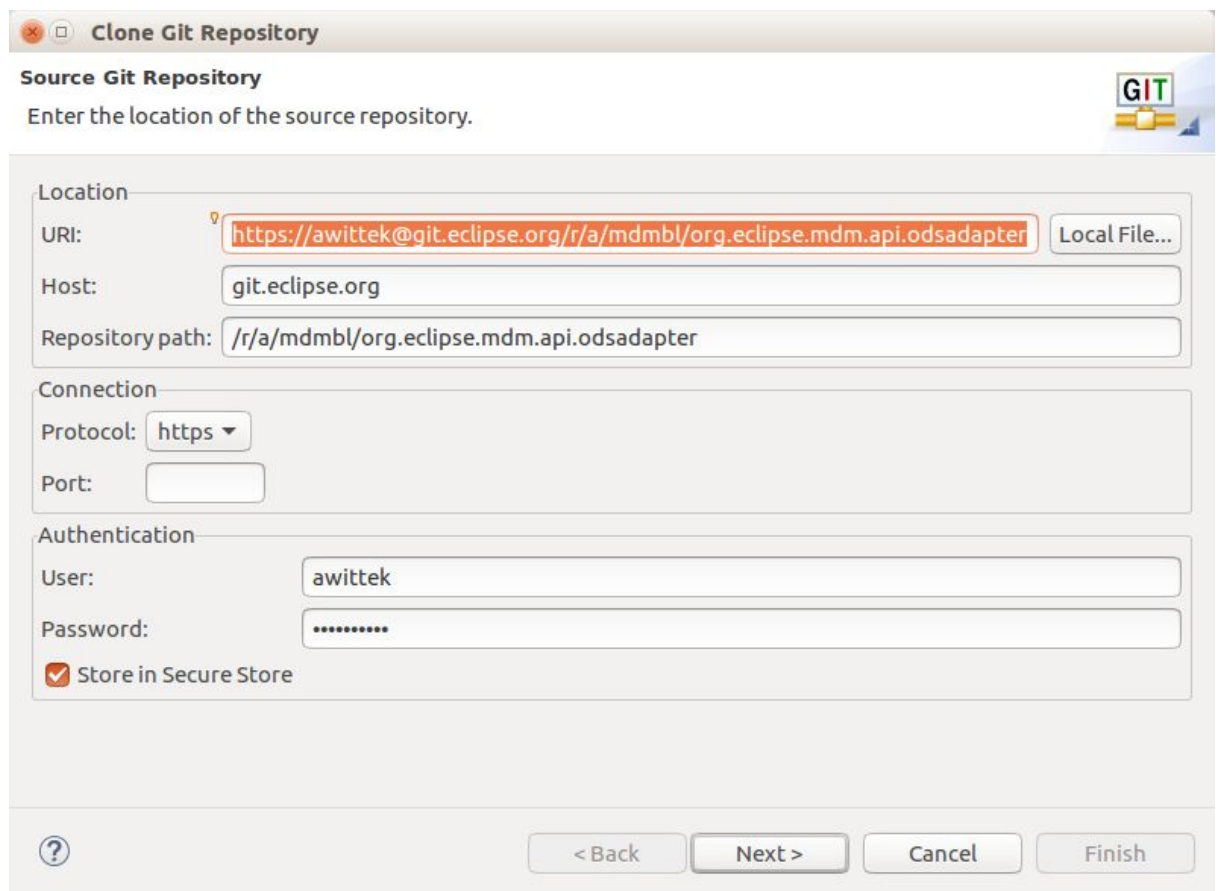
- Open the Git Perspective
- Open “Clone a Git Repository ...”
- Checkout with Username / Password:
`https://<eclipse_username>@git.eclipse.org/r/a/mdmbl/<my_repo>`

The password is the Gerrit Password, look there:

<https://git.eclipse.org/r/#/settings/http-password>

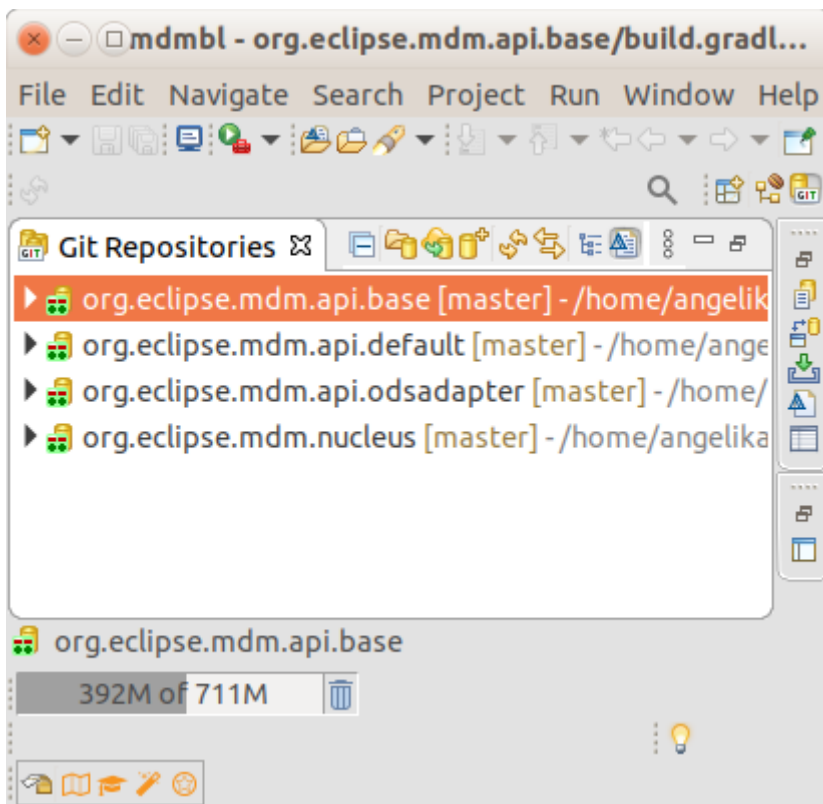
e.g. <https://awittek@git.eclipse.org/r/a/mdmbl/org.eclipse.mdm.api.base>

- Or anonymous with the URIs above.
- Screenshot:



After cloning the repositories, your Git perspective:

Note: if there are not these “green icons” means that Gerrit is not configured properly yet, follow the instructions in this [chapter](#).



3.3. Importing projects to the Eclipse IDE

Import the project to your Eclipse IDE via File -> Import -> Import as existing gradle project.

3.4. Building the projects

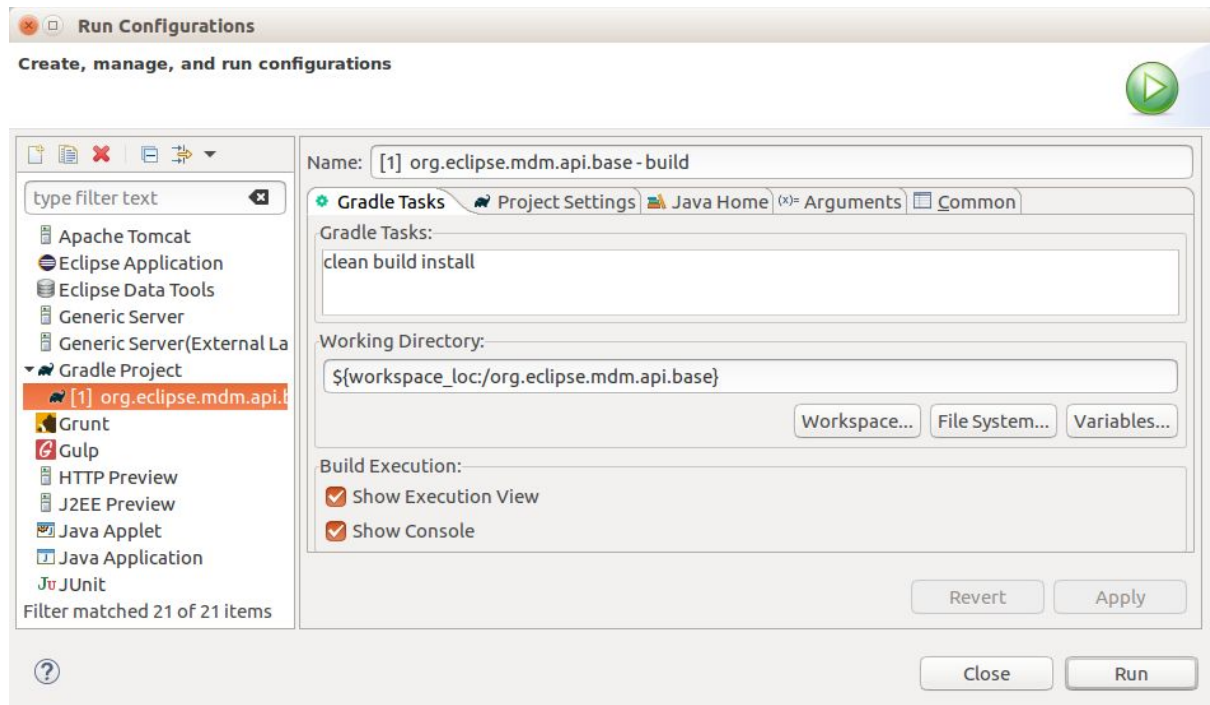
3.4.1. Building projects separately

Run `./gradlew clean build` in the root directory of each project to build it.

Execute `./gradlew clean build install` also generates a JAR to be placed in your local cache. From there it can be retrieved when subsequent projects are build.

Or you build the projects in the Eclipse IDE via a Run Configuration:

(Screenshot Eclipse 201903, beginning with Eclipse 201909 the Input Dialog changed, see below):



Note: the projects depend on each other, so you have to build them in a specific order:

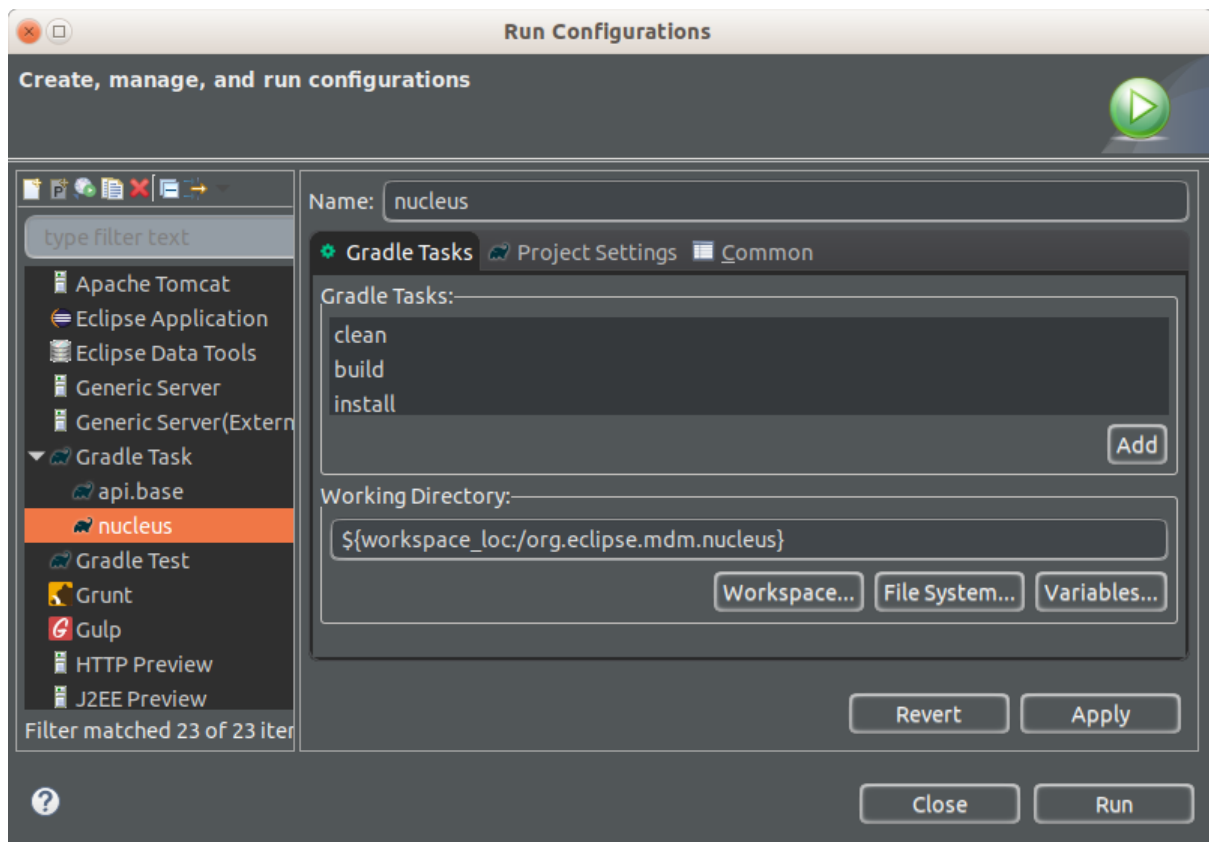
1. org.eclipse.mdm.api.base
2. org.eclipse.mdm.api.default
3. org.eclipse.mdm.api.odsadapter
4. org.eclipse.mdm.nucleus

The other projects can be built independently.

3.4.2. Building all projects for the core application

Running the gradlew script from org.eclipse.mdm.nucleus project will build the following projects:

1. org.eclipse.mdm.api.base
2. org.eclipse.mdm.api.default
3. org.eclipse.mdm.api.odsadapter
4. org.eclipse.mdm.nucleus



3.4.3. Building the projects without web frontend

To build the application without the web frontend, pass the argument **skipNode** to the build script of the org.eclipse.mdm.nucleus project.

```
> ./gradlew clean build -PskipNode
```

If there are still java errors after importing and building, right click on the project -> Gradle -> Refresh Gradle Project

For other build problems. see the section [Known setup bugs](#). If your problem is not listed there, create a Bugzilla Bug for it:

https://bugs.eclipse.org/bugs/enter_bug.cgi?product=MDMBL

3.5. Configure Git and Gerrit

For checking in code to a repository you need to configure Gerrit in the Git perspective.

Additional informations:

- Gerrit can be found at <https://git.eclipse.org/r/>
- There is a tutorial, it also explains how to use Gerrit with Eclipse: <http://www.vogella.com/tutorials/Gerrit/article.html>
- See documentation from Eclipse: <https://wiki.eclipse.org/Gerrit>

Prerequisites:

- Get your Gerrit Password via: <https://git.eclipse.org/r/#/settings/http-password>

- Configure your Gerrit settings: <https://git.eclipse.org/r/#/settings/>
-> Watched Projects -> Add the mdmbl projects.

3.5.1. Configure repositories

Go to your Eclipse IDE -> Git Perspective:

For all projects do:

- Expand the project
- Remotes -> origin -> Select "Configure Push" from context menu -> OK
- URI -> click change button
- Configure URI Window:
 - URI: `https://<eclipse_username>@git.eclipse.org/r/a/mdmbl/<projectname>`
e.g.
`https://<eclipse_username>@git.eclipse.org/r/a/mdmbl/org.eclipse.mdm.api.base`
 - Protocol: choose HTTPS
 - Fill in username and password (the Gerrit one from the settings above)
 - Check the option "Store in secure store"
- Back to the "Configure Push" Window, section "Ref mapping" -> add
 - Local Branch: HEAD
 - Choose remote branch:
 - for the dev branch e.g. `refs/for/dev`
 - for the master branch: `refs/heads/master`
 - OK
- Try the "Dry-Run" Button to test your configuration
- Save
- "Gerrit Configuration" (right click on the remote)
 - URI: same as above
 - Username: same as above
 - Destination branch: dev (or another)
- Finish

3.5.2. Configure Push

To push code to Gerrit you can

- 1) As a committer
 - a) Push code directly to Gerrit without any review
(in Git repositories view: Push to upstream)
 - b) Push code to Gerrit for reviewing (in Git repositories view: Push to Gerrit)
- 2) As a non-committer
 - a) Push code to Gerrit for reviewing (in Git repositories view: Push to Gerrit)

3.5.3. Commits to Gerrit

- Add Signed-Off-By when you commit to sign off the commit
- In Eclipse: open the git staging perspective, there is a window called "Commit message" and next to the label "Commit Message" there are 3 buttons. the middle button is the "Add signed-off-by" button. Click it.

4. Preconditions and required installations

4.1. Glassfish

Supported versions:

- 4.1.2
 - Download Full Platform: <https://javaee.github.io/glassfish/download>
 - Administration-Guide: <https://javaee.github.io/glassfish/doc/4.0/administration-guide.pdf>
 - Security-Guide: <https://javaee.github.io/glassfish/doc/4.0/security-guide.pdf>
 - Some patches are necessary. Refer to this [chapter](#) for patching glassfish.
- 5.1.0 - Full Platform from <https://projects.eclipse.org/projects/ee4j.glassfish/downloads>
 - Deployment with Glassfish Administration Console is not possible, because of a Glassfish Bug: 545170 - "Archive Path is NULL" when deploying from GUI (https://bugs.eclipse.org/bugs/show_bug.cgi?id=545170)
 - Administration-Guide: <https://javaee.github.io/glassfish/doc/5.0/administration-guide.pdf>
 - Security-Guide: <https://javaee.github.io/glassfish/doc/5.0/security-guide.pdf>

Installation:

- Unzip the file into a directory without spaces (recommended).
- Test it:
 - change to the glassfish-root directory
 - invoke `./glassfish/bin/asadmin start-domain``
 - change to your browser
 - URL localhost:8080 should show you a welcome page
 - URL localhost:4848 should show you the admin page
 - Note: if you need to change the default ports 8080 or 4848 please see the glassfish documentation

4.2. Database for the User Preference Service

The Preference service stores its data to a relational database. The database connection is looked up by JNDI and the JNDI name and other database relevant parameters are specified in `src/main/resources/META-INF/persistence.xml`. The default JNDI name for the JDBC resource is set to `jdbc/openMDM`.

For the User Preference Service you need a database with a schema "openMDM".

Note: "OPENMDM" is the default schema name, it can be changed in the file:

```
/org.eclipse.mdm.nucleus/org.eclipse.mdm.preferences/src/main/resources/META-INF/persistence.xml
```

For activating the change, the artefacts have to be rebuild.

4.2.1. Apache Derby Database

The Glassfish Application Server has included the Derby libraries.

Create database and tables:

- Change to your <glassfish_root> directory
- Start the database:

```
<glassfish_root>$ ./glassfish/bin/asadmin start-database
```

- Start the ij Tool (http://db.apache.org/derby/papers/DerbyTut/ij_intro.html):

```
<glassfish_root>$ java -jar ./javadb/lib/derbyrun.jar ij
```

- Create database with default name “openMDM”

```
ij> CONNECT  
'jdbc:derby://localhost:1527/openMDM;create=true';
```

- Create the table “preference”

```
ij> CREATE TABLE OPENMDM.PREFERENCE (ID BIGINT GENERATED BY  
DEFAULT AS IDENTITY NOT NULL, keyCol VARCHAR(255), SOURCE  
VARCHAR(255), username VARCHAR(255), valueCol  
CLOB(2147483647) NOT NULL, PRIMARY KEY (ID));
```

- Add constraints

```
ij> ALTER TABLE OPENMDM.PREFERENCE ADD CONSTRAINT  
UNQ_PREFERENCE_0 UNIQUE (source, username, keyCol);
```

- Check table:

```
ij> SELECT * FROM OPENMDM.PREFERENCE;
```

- Create the table “system_process” (since 5.2.0M1):

```
ij> CREATE TABLE system_process (ID BIGINT GENERATED BY  
DEFAULT AS IDENTITY NOT NULL, process_id VARCHAR(255),  
process_key VARCHAR(255), last_locked BIGINT, PRIMARY KEY  
(ID));
```

- close ij

```
ij> exit;
```

- Stop database.

```
<glassfish_root>$ ./bin/asadmin stop-database
```

4.2.2. Other Database Products

There is also the possibility to use other database products. For Postgres DB you will find the according sql scripts after the nucleus build in the following

directory: \$workspace/org.eclipse.mdm.nucleus/build/distributions/schema/org.eclipse.mdm.preferences. Other database products supported by EclipseLink may also work, but are neither tested nor supported by the mdmbl project.

4.2.3. Configure JDBC for the User Preference Service DB

For the **User Preference Service DB** configure JDBC resource and its dependent JDBC Connection Pool. It has to be created and configured through asadmin command line tool. (the glassfish web administration console is not recommended, as it is pretty buggy)

Description for the Derby Database with defaultname “openMDM”:

1. Start domain, database and admin-tool:

```
<glassfish_root>$ ./bin/asadmin start-domain
<glassfish_root>$ ./bin/asadmin start-database
<glassfish_root>$ ./bin/asadmin
```

2. Create JDBC Connection Pool “openMDM” with User, Password and DatabaseName is “openMDM”

```
asadmin> create-jdbc-connection-pool --datasourceclassname
org.apache.derby.jdbc.ClientDataSource --restype
javax.sql.DataSource --property
password=openMDM:user=openMDM:serverName=localhost:databaseName=openMDM:connectionAttributes=\\;create\\=true openMDM
```

3. Check JDBC Connection Pool via the command “ping-connection-pool” or “list-jdbc-connection-pools”
4. Create JDBC Resource:

```
asadmin> create-jdbc-resource --connectionpoolid openMDM
jdbc/openMDM
```

5. Check JDBC Resource via the command “list-jdbc-resources”
6. Stop domain, database:

```
<glassfish_root>$ ./bin/asadmin stop-domain domain1
```

```
<glassfish_root>$ ./bin/asadmin stop-database
```

4.3. Elasticsearch

Download Elasticsearch 7 for openMDM 5.1.0++:

<https://www.elastic.co/products/elasticsearch>

or

<https://www.elastic.co/de/downloads/past-releases#elasticsearch>

For openMDM 5.0.x and 5.1.Mx use version 2.x., e.g.

<https://www.elastic.co/de/downloads/past-releases/elasticsearch-2-4-2>

For testing purpose, it can be simply started by executing:

```
<elasticsearch_root>$ ./bin/elasticsearch
```

4.4. Database for ODS-Server

The database product is dependent of the ODS Server you use. To start an ODS Server you need a loaded ASAM ODS Application Model in the DB.

Embedded Apache Derby Database and Peak ODS Server

If you use a Peak ODS Server from Peak Solutions GmbH:

they provide an embedded demo Derby Database with an Application Model and Data.

Follow their instructions.

4.5. ODS Server

ASAM ODS-Server e.g. from

- Peak Solution:
<http://www.peak-solution.de/de/produkte-leistungen/versuchs-messdatenmanagement/softwareloesungen/peak-ods-server/>
- HiQSoft:
<https://www.highqsoft.com/de/avalon-asam-ods-server/>
- or another compliant data source (e.g. PAK adapter)
- Get a (test) license from the vendors and follow the installation instructions.
- If you already imported an Application Model to your database and database is running, then the ODS Server should start.

Notes for running a Peak ODS Server:

- You need two additional plugins copy them to `$odsserver_root/plugins`:
 - Put `peakcorbafileserv-VERSION.jar` into plugins
 - Put `notification-plugin-VERSION.jar` into plugins
- Configure parameters in `$odsserver_root/cfg/server.properties`:
 - Add the following line to enable the notification service:
`JMS_FORWARDER.PORT=8089`
 - The Peak ODS Server can run its own ORB daemon. Just leave the following parameter blank: `NAMESERVICE=`

- Use the Peak demo Derby DB (MDMNVH DB) in embedded mode, the database starts automatically with the Peak ODS Server startup.

Username and password are the name must be the name of the schema.

- `DB_DRIVER = DERBY_EMBEDDED`
- `DB_URL = jdbc:derby://<path_to_MDMNVH_DerbyDB>/MDMNVH;create=false`
- `DB_USER = MDMNVH`
- `DB_PASSWORD = MDMNV`

5. Install, deploy and configure application

For deployment, you need the build artefact: `org.eclipse.mdm.nucleus-<version>.war`

This file is an element of the file: `openMDM_application-<version>.zip`

Get this file from:

1. the Project Download Page:
<https://projects.eclipse.org/projects/technology.mdmbi/downloads>
2. or from your build:
`/org.eclipse.mdm.nucleus/build/distributions/`

The ZIP archive contains:

- the backend `org.eclipse.mdm.nucleus-VERSION.war`
- the configurations in `/configuration`
- sql scripts for the User Preference Service in `/schema`.

5.1. Configuration Files

Copy the content of the extracted `/configuration` folder to:

`$GLASSFISH_ROOT/glassfish/domains/domain1/config`

Configuration file: `org.eclipse.mdm.property/global.properties`

To enable globally the freetext search set the parameter: `freetext.active=true`, if enabled, make sure:

- that ElasticSearch is started and that the port in the property `elasticsearch.url` is set correct (check it in the ElasticSearch log and via your browser with the url and port, the result should be a json response)
- the freetext parameters are set in the `service.xml`
- freetext search is active for at least one service

Configuration file: `org.eclipse.mdm.connector/service.xml`

- configure the data sources, for ODS Servers look into your ODS Server log file to determine the corba URL
 - To use the freetext search configure for each datasource separately:
 - specific parameters for the NotificationService and the freetext search
 - set the `freetext.active` parameter to true (Example2)
 - Disable the freetext search for a datasource:
 - set the `freetext.active` parameter to false (Example3)
- or**
- leave away the `freetext.*` parameters, as they are optional (Example1)

```

<services>
<!-- Example1: ODS Server without freetext.* parameters -> freetext search is not active -->
<service entityManagerFactoryClass="org.eclipse.mdm.api.odsadapter.ODSContextFactory">
  <param name="nameservice">corbaloc::1.2@YOUR_HOST1:2809/NameService</param>
  <param name="servicename">YOUR_SERVICE1.ASAM-ODS</param>
</service>
<!-- Example2: Peak ODS-Sever with active freetext search -->
<service entityManagerFactoryClass="org.eclipse.mdm.api.odsadapter.ODSContextFactory">
  <param name="nameservice">corbaloc::1.2@YOUR_HOST2:2809/NameService</param>
  <param name="servicename">YOUR_SERVICE2.ASAM-ODS</param>
  <!--The indexing requires a user to get the DataItems from the ODS Server. Those are the
credentials for the user -->
  <param name="freetext.active">true</param> <!--to configure the Avalon, set to false (*) -->
  <param name="freetext.user">sa</param>
  <param name="freetext.password">sa</param>
  <param name="freetext.notificationType">peak</param>
  <param name="freetext.notificationUrl">http://YOUR_HOST2:8089/api</param>
</service>

<!-- Example3: Avalon ODS-Sever -> freetext search is not active-->
<service entityManagerFactoryClass="org.eclipse.mdm.api.odsadapter.ODSContextFactory">
  <param name="nameservice">corbaloc::1.2@YOUR_HOST3:2809/NameService</param>
  <param name="servicename">YOUR_SERVICE3.ASAM-ODS</param>
  <!--The indexing requires a user to get the DataItems from the ODS Server. Those are the
credentials for the user -->
  <param name="freetext.active">false</param> <!--to configure the Avalon, set to true (*) -->
  <param name="freetext.user">sa</param>
  <param name="freetext.password">sa</param>
  <param name="freetext.notificationType">avalon</param>
  <param name="freetext.pollingInterval">5000</param>
</service>
</services>

```

Note:

An explanation about the the corbaloc and corbaname URLs, find here:

<http://www.ciaranmchale.com/corba-explained-simply/the-corbaloc-and-corbaname-urls.html>

Restart the application server

5.2. Configure Login Module (Realm Configuration)

The realm is configured in a standardized way. Configure your LDAP, AD or other systems according to the glassfish security documentation: [Glassfish 5](#) or [Glassfish 4](#)

For local installations you can setup and configure a file realm, described in the readme.md file of the org.eclipse.mdm.nucleus project:

<http://git.eclipse.org/c/mdmbl/org.eclipse.mdm.nucleus.git/tree/README.md>

Note: The component "org.eclipse.mdm.realms" is not used any longer, since 5.0.0M4. And is not longer supported.

5.3. Property lookups for parameters values in service.xml

The service.xml contains all information necessary for the Connector-Service to connect to the available data sources/adaptor instances. Since this information includes secret information like passwords, it is possible to provide lookups, which gives you the possibility to specify tokens as references to properties defined elsewhere.

There are different lookups available:

* sys: Looks up variables defined as system properties

* env: Looks up variables defined as environment variables

Example:

```
<param name="password">${env:odsPassword}</param>
```

5.4. Configure system properties

- use system property "org.eclipse.mdm.api.odsadapter.filetransfer.interfaceName" to set a specific network interface name to be used
- use system property "org.eclipse.mdm.configPath" to redefine the location of the Glassfish config folder, e.g. "GLASSFISH_ROOT/glassfish/domains/domain1/config"

5.5. Deployment of org.eclipse.mdm.nucleus

Deploy the backend (org.eclipse.mdm.nucleus-VERSION.war) on your Glassfish server. Execute the following command (replacing PATH-TO-WAR and VERSION)

```
<glassfish_root>$ ./bin/asadmin
asadmin> deploy --force=true --name org.eclipse.mdm.nucleus
<PATH-TO-WAR>/org.eclipse.mdm.nucleus-<VERSION>.war
```

Afterwards, goto <http://localhost:8080/org.eclipse.mdm.nucleus> and Login with sa/sa

Note: Deployment in GlassFish 5 Server Administration Console is not working.

5.6. Headless Deployment

The default configuration for the authentication method is set to "FORM" . If you want to do a headless deployment change it to "BASIC".

Change this file:

org.eclipse.mdm.nuclues/org.eclipse.mdm.application/src/main/webconfig/web.xml

from

```
<auth-method>FORM</auth-method>
```

to

```
<auth-method>BASIC</auth-method>
```

6. Start application

- start ORB (`$JAVA_HOME/bin/orbd -ORBInitialPort 2809`)
(skip this if you are using Peak ODS Server with no NAMESERVICE specified in the `server.properties`)
- start the database for the ODS Server (if necessary)
- start the ODS server
- start Elasticsearch
- start Glassfish (including database for UPS)
- start Glassfish
 - `> asadmin start-database`
 - `> asadmin start-domain`

Change to your browser URL is `http://localhost:8080/org.eclipse.mdm.nucleus`.

You should see the openMDM LoginPage. Look for user/ password in the database in the `userXX` table, e.g. `sa/sa`.

6.1. Troubleshooting

Look into the Logfiles:

- Glassfish:
`$glassfish_root/domains/domain1/logs/server.log`
- Derby DB for User Preference Service:
`$glassfish_root/databases/derby.log`
- The Logfiles of your ODS compatible datasource

7. Known (setup) problems and solutions

7.1. Build

- The build of the `org.eclipse.mdm.api.odsadapter` project does need a “gradle clean install” as a “gradle install” breaks with build errors that package “`com.google.protobuf`” does not exist

7.2. Glassfish

7.2.1. Glassfish - Inconsistent Module State

- `org.glassfish.deployment.common.DeploymentException: Error in linking security policy for org.eclipse.mdm.nucleus -- Inconsistent Module State`
 - **Deployment went wrong**

- delete
\$glassfish_root/glassfish/domains/domain1/applications/org.eclipse.mdm.nucleus
- delete \$glassfish_root/glassfish/domains/domain1/generated
- restart Glassfish

7.2.2. Glassfish - WEB9102: Login failed: Lookup failed

If you cannot login to the webclient and the glassfish log shows the following error:

```
WEB9102: Web Login Failed:
com.sun.enterprise.security.auth.login.common.LoginException:
Login failed: Lookup failed for
'java:global/org.eclipse.mdm.nucleus/ConnectorService!org.eclipse.mdm.connector.boundary.ConnectorService' in
SerialContext[myEnv={java.naming.factory.initial=com.sun.enterprise.naming.impl.SerialInitContextFactory,
java.naming.factory.state=com.sun.corba.ee.impl.presentation.rmi.JNDIStateFactoryImpl,
java.naming.factory.url.pkgs=com.sun.enterprise.naming}|#]
```

check that the war file was deployed with the Application Name “org.eclipse.mdm.nucleus”

7.2.3. Glassfish 4 only - java.lang.ClassNotFoundException

If you run into "java.lang.ClassNotFoundException:

javax.xml.parsers.ParserConfigurationException not found by org.eclipse.persistence.moxy" this is a bug described in https://bugs.eclipse.org/bugs/show_bug.cgi?id=463169 and <https://java.net/jira/browse/GLASSFISH-21440>.

This solution is to replace

GLASSFISH_HOME/glassfish/modules/org.eclipse.persistence.moxy.jar with this:

<http://central.maven.org/maven2/org/eclipse/persistence/org.eclipse.persistence.moxy/2.6.1/org.eclipse.persistence.moxy-2.6.1.jar>

7.2.4. Glassfish 4 only - org.osgi.framework.BundleException

Glassfish 4 only:

If you run into “org.osgi.framework.BundleException: Unresolved constraint in bundle com.fasterxml.jackson.module.jackson-module-jaxb-annotations”: this is a compatibility problem with the installed jackson libraries.

The solution is to replace

GLASSFISH_HOME/glassfish/modules/jackson-*.jar with these files:

- <http://central.maven.org/maven2/com/fasterxml/jackson/core/jackson-annotations/2.8.1/jackson-annotations-2.8.1.jar>

- <http://central.maven.org/maven2/com/fasterxml/jackson/core/jackson-core/2.8.1/jackson-core-2.8.1.jar>
- <http://central.maven.org/maven2/com/fasterxml/jackson/core/jackson-databind/2.8.1/jackson-databind-2.8.1.jar>
- <http://central.maven.org/maven2/com/fasterxml/jackson/jaxrs/jackson-jaxrs-base/2.8.1/jackson-jaxrs-base-2.8.1.jar>
- <http://central.maven.org/maven2/com/fasterxml/jackson/jaxrs/jackson-jaxrs-json-provider/2.8.1/jackson-jaxrs-json-provider-2.8.1.jar>

Rename the jars to the replaces (remove the versions).

7.2.5. Glassfish 4 only - java.lang.NoSuchMethodError

Glassfish 4 only:

If you are using the REST-API to access the MDM entities (CREATE und UPDATE) and encounter the following error:

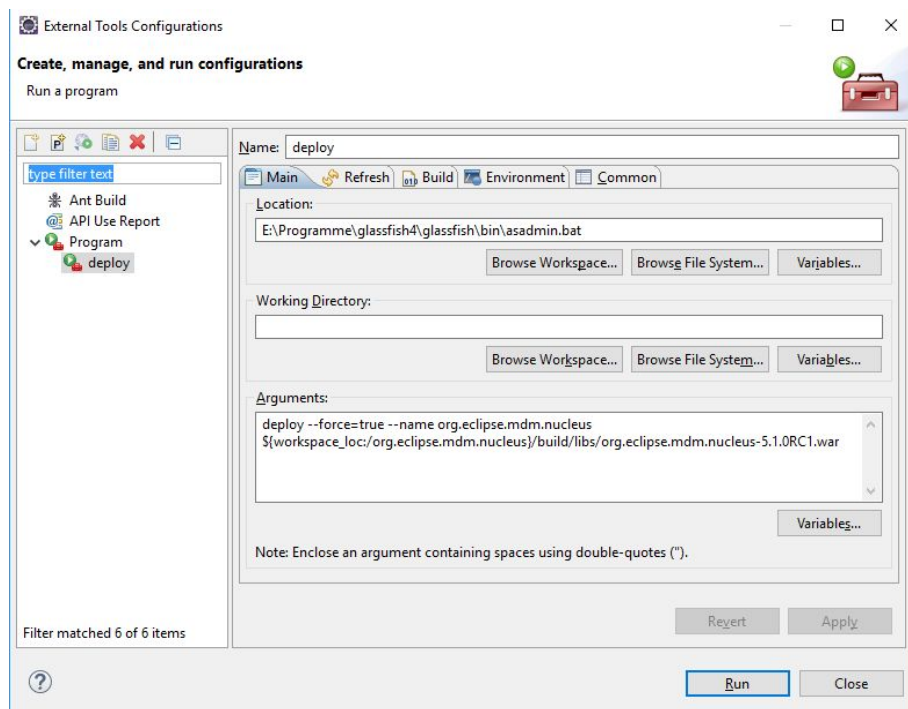
```
java.lang.NoSuchMethodError:  
com.fasterxml.jackson.databind.deser.std.UntypedObjectDeserializer  
.<init>
```

replace the jackson libraries as mentioned in 7.2.3.

8. Extended configurations (run & IDE)

8.1. Deployment via Eclipse IDE

To deploy the nucleus in Glassfish asadmin can be configured as an External Tool in Eclipse:



8.2. Angular Live Development Server

During development a full build of the MDM5 Webclient, which is part of the nucleus, is quite time consuming. It is more efficient to skip the client build entirely by using the parameter `-PskipNode`, which excludes the JavaScript-Code from being build and included in the war file. For the UI you can start the "NG Live Development Server".

It is started by invoking "npm start" in the root path of the angular application (org.eclipse.mdm.nucleus\org.eclipse.mdm.application\src\main\webapp). After successful compilation you are able to see the webclient under <http://localhost:4200/>.

Changes to the angular application will trigger an incremental compile of the changed sources.

Additionally the the live development server implements a proxy, which forwards all requests to http://localhost:4200/org.eclipse.mdm.nucleus/* to <http://localhost:8080/>. So if the Glassfish with the MDM5 backend does not run on port 8080, you have to change the proxy configuration in

org.eclipse.mdm.nucleus/org.eclipse.mdm.application/src/main/webappproxies/development.config.json.

To avoid issues with the form based authentication, it is easier to switch to basic authentication by changing the web.xml in this scenario. Otherwise you may get errors for the backend requests, because you are not automatically forwarded to the login page.

8.3. SonarQube

The Eclipse SonarQube server is available via <https://sonar.eclipse.org>.

Note: Do not use SonarLint plugin as the current version does not support the old SonarQube version at Eclipse and the old one has less features that the SonarQube plugin

- Install SonarQube plugin version 3.5
 - Install new Software...
 - Add <http://downloads.sonarsource.com/eclipse/eclipse/>
 - Select all plugins to be installed
 - Oxygen might complain about Erlang Configuration Helper to not be installable
- Configure SonarQube in Eclipse: Eclipse Preferences -> SonarQube -> Servers
 - Edit the default server or create a new one if none present
 - SonarQube Server ID: Eclipse
 - SonarQube Server URL: <https://sonar.eclipse.org>
 - Username: your username for eclipse.org
 - Password: your password for eclipse.org
- Associate projects with Eclipse SonarQube:
 - Edit Project Properties for all projects from
 - org.eclipse.mdm.api.base
 - org.eclipse.mdm.api.default
 - org.eclipse.mdm.api.odsadapter
 - org.eclipse.mdm.nucleus
- Configure
 - Associate with SonarQube: Association should be proposed automatically
- Available Views are
 - SonarQube Issues
 - SonarQube Rule Description
- Analysis can be performed via
 - Project context menu
 - SonarQube -> Analyze
 - The results of the analysis are then written to Eclipse SonarQube

9. Development Rules

9.1. Commit comments

Prefix every commit with the Bugzilla ID if there is one, e.g. your commit message would be "518433: made some changes".

9.2. Tasks in the code

TODO / FIXME and other tasks have to follow this naming schema:

```
// {TASK_NAME} {my_committer_id}, {yyyy-MM-dd}: {my_comment}
```

e.g.

```
// TODO mkoller, 2017-11-14: Currently only the first Adapter is indexed.
```

You can do that manually or you can import the following snippet In the Eclipse IDE via

Preferences ->Java -> Editor -> Templates

```
<!-- Import: Eclipse IDE -> Window -> Preferences -> Java -> Editor Templates -> Import --><!-- User Variable: add to eclipse.ini -> "-Duser.name=<my name>" ->
<templates>
  <template autoinsert="true" context="java" deleted="false" description=""
enabled="true"      name="TODO">// TODO ${user}, ${d:date('yyyy-MM-dd')}:
  </template>
  <template autoinsert="true" context="java" deleted="false" description=""
enabled="true" name="FIXME">// FIXME ${user}, ${d:date('yyyy-MM-dd')}:
  </template>
</templates>
```

Usage in code: "TOD"+ [Ctrl+Space]

You have to set your Eclipse user to your committer ID:

edit the file eclipse.ini and set the parameter:

-Duser.name=your_eclipse_committer_id

9.3. Working with committer feature branches

Develop on your committer feature branch. Make code reviews as usual via Gerrit on your branch.

Committer feature branches have a special name scheme:

`/${my_committer_id}/my_branch`.

Only for those references under refs/heads (and refs/tags) with your Committer ID you will be able to delete the branch afterwards.

- To create a branch: create the branch with the name scheme -> **`/${my_committer_id}/my_branch`**
- To delete a branch in Eclipse: Project → Team → Remote → Push ...
 - “Next” on first wizard screen
 - Select branch to delete in “Add delete ref specification”
 - Click “Add Spec” (the branch is now listed in “Specifications for push”)
 - “Finish”
- When integrating your branch into the dev branch
 - Checkout the dev branch
 - Merge your feature branch into the dev branch, but with having Git a merge commit created although it was a fast-forward merge

```
git merge --no-ff your_feature_branch
```
 - Push the dev branch to Gerrit

Note: Without the no-ff option Gerrit refuses to accept the push due to “no new changes”.

10. Tipps+Tricks

10.1. Handling of Relations

- First, you should consider if you really need to access relations directly. For most use cases there already is a function in the OpenMDM 5 business logic available which does what is required.
- The BaseEntityManager interface allows you to traverse parent and children relations.
- If you need to read other relation types, you can implement and use a BaseEntitySearchQuery. For details look at the method changeRelation in org.eclipse.mdm.api.odsadapter.RelationTest.
- For setting relations you need to access to BaseEntityFactory.get*Store. Typically this is only possible for the adapter implementation, because the contents of the stores might not be 100% consistent over different adapters. The adapter may choose make these methods publicly accessible, as is the case for the ODSAdapter, but be aware that this is currently adapter specific. See org.eclipse.mdm.api.odsadapter.RelationTest for an example. Please note that you can only modify mutable relations (in the MutableStore), e.g. no parent child relations.

11. APPENDIX

11.1. Web Frontend: configure icons

All icons for the web frontend are taken from the FAMFAMFAM Silk Icons library, version 1.3 (<http://www.famfamfam.com/lab/icons/silk/>) This library is licenced under the Creative Commons Attribution 3.0 License (<https://creativecommons.org/licenses/by/3.0/>). This library was approved by the Eclipse Foundation, see CQ 17759.





Note:















A lot of users of other mdm applications are used to the icons from this document

https://www.highqsoft.com/download/ao_base.htm

These icons are not open source, so we do not use them in our application.

The mapping from the ao elements to FAMFAMFAM icons:

ASAM ODS 5.3.0 Base Element Definitions		FamFamFam Silk Icons, V1.3
https://www.highqsoft.com/download/ao_base.htm#AoFile		page_white_stack
https://www.highqsoft.com/download/ao_base.htm#AoEnvironment		database
folder		folder
https://www.highqsoft.com/download/ao_base.htm#AoMeasurementQuantity		chart_curve_go

https://www.highqsoft.com/download/ao_base.htm#AoMeasurement		chart_curve
https://www.highqsoft.com/download/ao_base.htm#AoParameter		shape_square
https://www.highqsoft.com/download/ao_base.htm#AoParameterSet		shape_move_forwards
project		house
structure level		paste_plain
https://www.highqsoft.com/download/ao_base.htm#AoSubmatrix		calendar
https://www.highqsoft.com/download/ao_base.htm#AoTestEquipmentPart		monitor
https://www.highqsoft.com/download/ao_base.htm#AoTestEquipment		computer
https://www.highqsoft.com/download/ao_base.htm#AoTest		brick_add
https://www.highqsoft.com/download/ao_base.htm#AoTestSequencePart		shape_move_front
https://www.highqsoft.com/download/ao_base.htm#AoTestSequence		page_white_stack
test step		brick
https://www.highqsoft.com/download/ao_base.htm#AoUnitUnderTestPart		cog
https://www.highqsoft.com/download/ao_base.htm#AoUnitUnderTest		cog_go

The mapping is defined in:

org.eclipse.mdm.nucleus/org.eclipse.mdm.application/src/main/webapp/src/styles.css

11.2. Web Frontend: I18N configuration

I18N support for web frontend, languages available: English, German.

For adding a new language see:

org.eclipse.mdm.nucleus/org.eclipse.mdm.application/src/main/webapp/README_I18N.md

Files with translations:

org.eclipse.mdm.nucleus/org.eclipse.mdm.application/src/main/webapp/src/assets/i18n/*.json