Setup Development Environment for the openMDM(R) Application

Eclipse mdmbl project

Document history:

Author	Date	Affects Version	Description
Angelika Wittek	9.6.2017	0.6	initial version
Angelika Wittek	12.6.2017	0.6	User Preference Service added
Alexander Nehmer	22.6.2017	0.6	Review and additions
Angelika Wittek	30.6.2017	0.6	Mailing Lists and ECA infos added
Angelika Wittek	18.7.2017	0.7	Comments from Ganesh inserted, Glassfish Bugs added
Angelika Wittek	31.07.2017	0.7	Elasticsearch version added, exported to pdf for mdmbl page

Table of contents

1 Introduction

This document serves as a guide for developers of the Eclipse mdmbl project. It describes how to setup your environment, where the source code of existing projects can be found and retrieved, how the application can be built, deployed and how to start the application.

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
- JIRA REQU Issues: https://openmdm.atlassian.net/secure/RapidBoard.jspa?rapidView=57&projectKey=REQU&view=planning&selectedIssue=REQU-48
- openMDM git repos: http://git.eclipse.org/c/?q=mdm

Note: this document is written in Google Docs, location:
https://docs.google.com/document/d/1Cvu5cpm4fhlM_G4E1hdx7bT_XTATrvGRjCiBIAxcfQ8/edit?usp=sharing

2 Prerequisites

Already installed on your machine or install it:

- Java 8: http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html
- Maven: https://maven.apache.org/
- Git: https://git-scm.com

3 Installations

3.1 Gradle

(This section is copied from the openMDM@Web project)

3.1.1 Gradle - via GVM (recommended)

- Follow the instructions found at to install GVM: http://gvmtool.net/
- You need a POSIX environment if running Windows. We recommend using Babun Shell: http://babun.github.io/
- Once GVM is installed invoke `gvm install gradle 2.2`.
- Test your setup by invoking `gradle --version`.

3.1.2 Gradle - manually

- Download Gradle from http://gradle.org/downloads
- Unzip the file into a directory without spaces (recommended).
- Create a GRADLE_HOME environment variable that points to this directory.
- Adjust your PATH environment variable to include $GRADLE_HOME/bin (%GRADLE_HOME%\bin on Windows).
- Test your setup by invoking `gradle --version`.

3.1.3 Gdub (Optional, but recommended)

GDub is a wrapper script that facilitates invoking gradle tasks anywhere within a Gradle project. It's smart enough to use the gradle wrapper if available or your global gradle command. Gradle commands can be run via `gw <task name>`.

Follow the instructions found at: https://github.com/dougborg/gdub
to install gdub

3.2 Eclipse IDE

Get the current JEE Distribution from https://www.eclipse.org/downloads
It is recommended to use the Eclipse Installer.

Install Plugins:

- Gradle: follow this tutorial http://www.vogella.com/tutorials/EclipseGradle/article.html.
- 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 formatting rules to the Eclipse default rules
    -

3.3 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`

3.3.1 Apache Derby Database

There is an Quickstart Guide from Deby:
https://builds.apache.org/job/Derby-docs/lastSuccessfulBuild/artifact/trunk/out/getstart/index.html

- Download it from https://db.apache.org/derby/releases/release-10.13.1.1.cgi
- Installation Guide: http://db.apache.org/derby/papers/DerbyTut/install_software.html
    - set $DERBY_INSTALL to your installation dir
    - CLASSPATH must include $DERBY_INSTALL/lib/
        - derby.jar
        - derbyclient.jar
        - derbynet.jar
        - derbytools.jar
- Start database: $DERBY_INSTALL/bin/startNetworkServer.bat
- Create a directory for the database data and change to it

- Start the ij tool as described here: http://db.apache.org/derby/papers/DerbyTut/ij_intro.html
- Create database with default name “openMDM”

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

- Create preference table:

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

- Check table:

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

- close ij;
- Stop database.

3.3.2 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.

3.4 Glassfish

- Version 4.1.2 required
- Download Glassfish from http://download.oracle.com/glassfish/
- Unzip the file into a directory without spaces (recommended).
- Test it:
    - change to the glassfish-root directory
    - invoke `./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

For the User Preference Sevice configure JDBC resource and its dependent JDBC Connection Pool. It has to be created and configured within the glassfish web administration console or through asadmin command line tool.

Description for the Derby Database with defaultname “openMDM”:

- Start Derby DB
- start: http://localhost:4848/
- Menu Item: JDBC-> JDBC Connection Pools -> new
    - poolname: <mypool_name>
    - Resource Type: javax.sql.DataSource
    - Database Driver Vendor: Derby
    - -> next
    - set properties: User, Password, DatabaseName to openMDM
    - -> finish
    - check it: open Connection Pool, try the ping button
- Menu item JDBC -> JDBC Resources -> new
    - JNDI NAME: jdbc/openMDM
    - Pool Name: <mypool_name>
- stop glassfish `./bin/asadmin stop-domain domain1`
- stop Derby DB

Now you can start the Derby DB via:

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

3.5 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.

3.5.1 Embedded Apache Derby Database

If you use a Peak ODS Server from Peak Solutions GmbH: they provide an embedded Derby Database with an Application Model and Data. Follow their instructions.

3.5.2 Oracle 11g XE Release 2

- Download it from http://www.oracle.com/technetwork/database/database-technologies/express-edition/downloads/index.html
- Install it.
- Note for Ubuntu Users: see this article http://tuhrig.de/3-ways-of-installing-oracle-xe-11g-on-ubuntu/
- Note for Linux Users: Installing the Docker Container from https://hub.docker.com/r/alexeiled/docker-oracle-xe-11g/ works fine (proofed by Angelika Wittek, 25.4.2017)

Load an empty Application Model to the DB: please refer to Jira-Ticket ORGA-178
TODO: check into git the dump and the descriptions and update link

3.6 ODS Server

ASAM ODS-Server e.g. from

- Peak Solutions: 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 the notification service. Check if there is the `notification-plugin-1.1.0.jar` in the `$odsserver_root/lib directory` or copy it there.
- And add the following line to `$odsserver_root/cfg/server.properties: JMS_FORWARDER.PORT=8089`
- The Peak ODS Server can run its own ORB daemon. Just leave `NAMESERVICE=` in `$odsserver_root/cfg/server.properties`
- If using Derby DB with demo data from Peak
    - The provided MDMNVH-DerbyDB Backup must be copied into the Derby database folder (besides openMDM DB folder; see 3.3.1)
    - `$odsserver_root/cfg/server.properties (username must be the name of the schema and password must be any non zero length string)`
        - `DB_URL = jdbc:derby://localhost:1527/MDMNVH`
        - `DB_USER = MDMNVH`
        - `DB_PASSWORD = dummypw`
        - `DB_DRIVER = DERBY`

3.7 ElasticSearch

ElasticSearch can be downloaded at https://www.elastic.co/products/elasticsearch.
use a version 2.x., e.g.
https://www.elastic.co/de/downloads/past-releases/elasticsearch-2-4-2

Upgrade to version 5.x is planned, see:
https://bugs.eclipse.org/bugs/show_bug.cgi?id=520297

For testing purpose, it can be simply started by executing bin/run.bat

4 Get and build the code

4.1 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.

- Create a new workspace directory
- Change to this directory
- Run the following command to checkout the source code. You have to adapt the URL for each project: `git clone <repository URL>`

Checkout the following projects

```
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
git clone http://git.eclipse.org/gitroot/mdmbl/org.eclipse.mdm.realms.git
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.mdfsorter.git
```

Alternatively configure eGit in your Eclipse IDE and checkout there.

4.2 Building the projects

Please see the README.md files in the root directory of the projects for exact information about how the projects are build.
Generally you may run `./gradlew clean build` in the root directory of each project to build it. Execute `./gradlew clean build install` to also generate a JAR to be placed in your local cache. From there it can be retrieved when subsequent projects are build.

Import the project to your Eclipse IDE via File -> Import -> Import as existing gradle project. If importing all projects at once via the parent directory fails, import each project separately. Build the projects in the order they are listed above, because of the dependencies between them.
Refer to the documentation and follow it:

- http://git.eclipse.org/c/gerrit/mdmbl/org.eclipse.mdm.nucleus.git/tree/README.md

- http://git.eclipse.org/c/gerrit/mdmbl/org.eclipse.mdm.realms.git/tree/README.md

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 4.4 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

After importing the projects to the IDE, you have all repository connections in the Git perspective.

4.3 Configure Gerrit

Gerrit can be found at https://git.eclipse.org/r/

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

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

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

Configure your Gerrit settings:
https://git.eclipse.org/r/#/settings/
-> Watch the mdmbl projects.

4.3.1 In Eclipse

Go to your Eclipse IDE -> Git Perspective
For all project do:

- Expand the project
- Remotes -> Create Remote
- Remote Name -> e.g. CodeReview; Select “Configure Push” -> 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
    - Remote branch: refs/for/dev
    - OK
- Try the “Dry-Run” Button to test your configuration
- Save
- “Gerrit Configuration” (right click on new remote)
    - URI: same as above
    - Username: same as above
    - Destination branch: dev
- Finish

For every commit 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.

4.4 Push code

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)

4.5 Known setup bugs and problems

4.6 Bugs

- Nucleus: config-dir is missing in the build artefact mdm-web.zip (https://bugs.eclipse.org/bugs/show_bug.cgi?id=518063)
  Workaround is described in the bug comments. in the .
- The build of openatfx.mdf breaks because of the Download Task (https://bugs.eclipse.org/bugs/show_bug.cgi?id=519453)
  Workaround is described in the bug comments. in the .
- 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

4.7 Problems and Solutions

4.7.1 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 `$glassfish_root/glassfish/domains/domain1/applications/org.eclipsemdm.nucleus`
        - delete `$glassfish_root/glassfish/domains/domain1/generated`
        - restart Glassfish

4.7.2 Glassfish - 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

5 Deploy and configure application

before deploying the 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 ODS Server with corresponding database
- start Elasticsearch
- start Glassfish
    - `> asadmin start-domain`
    - `> asadmin start-database`

The build of the nucleus module creates:
`/org.eclipse.mdm.nucleus/build/distributions/mdm_web.zip`

The ZIP archive contains:

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

Deploy the backend (org.eclipse.mdm.nucleus.war) on your Glassfish server
=> do it via the admin console at http://localhost:4848/

Configuration:

- copy the content of the extracted /configuration folder to `$GLASSFISH_ROOT/domains/domain1/config`
- edit the `org.eclipse.mdm.connector/service.xml` file to configure the data sources, look into your ODS Server log file to determine the corba URL.
  e.g. for the Peak ODS Server with embedded Derby Database use:

```
<service
entityManagerFactoryClass="org.eclipse.mdm.api.odsadapter.ODSEntityManagerFactory">
    <param name="nameservice">corbaloc:iiop:127.0.0.1:2809/NameService</param>
   <param name="servicename">MDMNVH.ASAM-ODS</param>
</service>
```

- There is a config parameter to enable / disable the elasticsearch in `$GLASSFISH_ROOT/domains/domain1/config/org.eclipse.mdm.property/global.properties`
  Parameter: `freetext.active=[true|false]`
- if you enable the freetext.active parameter (=true), 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)
- install and configure the LoginModule (see org.eclipse.mdm.realms - README.md)
- restart the application server
- goto http://localhost:8080/org.eclipse.mdm.nucleus and Login with sa/sa

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)

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.

7 Development Rules

- prefix every commit with the Bugzilla ID if there is one, e.g. your commit message would be “518433: made some changes”

8 Troubleshooting

Look into the Logfiles:

- Glassfish:
  `$glassfish_root/domains/domain1/logs/server.log`
- Derby DB for User Preference Service:
  `$glassfish_root/databases/derby.log`
- Peak ODS Server:
  `$peakodsserver_root/logs/`