

Installation Guide for the openMDM5(R) web application

Eclipse mdmbl project

Document history:

Author	Date	Affects SW Version	Description
Angelika Wittek	15.06.2017	V 0.6	initial version
Angelika Wittek	10.07.2017	V 0.6	Deployment added
Angelika Wittek	31.07.2017	V 0.7	Elasticsearch version and section "problems and solutions" added. Exported for V0.7
Angelika Wittek	13.09.2017	V0.7 / V0.8	additions for new version exported to pdf for download pages
Angelika Wittek	09.11.2017 21.11.2017	V0.7 / V0.8 /V0.9	added information about ODS Server used for Tests
Angelika Wittek	23.11.2017	V0.9	Documented changes in service.xml, exported for V0.9
Angelika Wittek	26.02.2018	V0.10	Changes for V0.10
Angelika Wittek	02.05.2018	5.0.0M1	Chapter 6 extended
Angelika Wittek	14.05.2018	5.0.0M1	Updates for new version
Angelika Wittek	10.9.2018	5.0.0M2 / 5.0.0M3	Updates for new version
Angelika Wittek	25.10.2018	5.0.0M4	Realm configuration changes added.

Table of Contents

1 Introduction	3
1.1 ODS Server used for Developer Tests	3
2 Prerequisites	4
3 Download Application Artefacts	4
4 Installation	4
4.1 Setup User Preference Service Database	5
4.1.1 Setup default schema / user (openMDM)	5
4.1.2 Configure the JDBC resource and its dependent JDBC Connection in your Glassfish	5
4.1.3 Setup with another schema name	6
4.2 Install ElasticSearch	6
4.3 Deploy web application to glassfish	6
4.4 Configure property files	6
4.5 Install and configure Glassfish Login Realm	8
4.6 Headless Deployment	8
5 Start application	9
6 Problems and Solutions	9
6.1 Glassfish - Inconsistent Module State	9
6.2 Glassfish - java.lang.ClassNotFoundException	9
6.3 Glassfish - org.osgi.framework.BundleException	10
6.4 Glassfish - java.lang.NoSuchMethodError	10
6.5 Glassfish - WEB9102: Login failed: Lookup failed	10
7 Troubleshooting	11

This document is published under the Eclipse Public License 1.0:

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

Note: this document is written in Google Docs, location

<https://docs.google.com/document/d/13QfiZh7rQBkslrWW2GR7Kd6YIp2mnnoRSmFr8AFABsg/edit?usp=sharing>

1 Introduction

This document serves as an installation guide for the openMDM5 web application. The application is build and the artefacts to ready to deploy.

Please note: this guide describes the deployment only, for installing the application in a company infrastructure, please contact your administrators to support you with firewalls and proxy configurations, grant permissions, etc..

Interesting links:

- openMDM website: <https://www.openmdm.org/>
- Eclipse project page: <https://projects.eclipse.org/projects/technology.mdmb1>
- The openMDM Working group mailing list:
<https://dev.eclipse.org/mailman/listinfo/open-measured-data-wg>
- JIRA:
<https://openmdm.atlassian.net/secure/BrowseProjects.jspx?selectedCategory=all&selectedProjectType=software>
- JIRA REQU Issues:
<https://openmdm.atlassian.net/secure/RapidBoard.jspa?rapidView=57&projectKey=REQU&view=planning&selectedIssue=REQU-48>

For any questions please contact us via the mailinglist:

<https://dev.eclipse.org/mailman/listinfo/mdmb1-dev>

Report a bug via:

<https://bugs.eclipse.org/bugs/buglist.cgi?quicksearch=mdmb1>

To build the application from the code base, please follow this documentation:

https://wiki.eclipse.org/images/e/e0/GettingStarted_mdmb1.pdf

1.1 ODS Server used for Developer Tests

All released versions are tested with the following ODS Server:

- HiQSoft GmbH, Avalon Version 4.6:
<https://www.highqsoft.com/de/avalon-asam-ods-server/>
- Peak Solution GmbH, ODS Server Version 2.5.8
<http://www.peak-solution.de/de/produkte-leistungen/versuchs-messdatenmanagement/softwareloesungen/peak-ods-server/>

NOTE: There is an issue with the an older Avalon Server version 4.3b (from 2013): if multiple catalog sensors are deleted, the server application throws an error and leaves the application model in a broken state.

2 Prerequisites

In your infrastructure you need to have installed:

- Java 8 (JDK1.8.0_45 or higher):
<http://www.oracle.com/technetwork/java/javase/downloads/>
- Glassfish 4.1.2 : <https://javaee.github.io/glassfish/download>
some Glassfish libraries have to be patched, refer to:
 - [6.2 Glassfish - java.lang.ClassNotFoundException](#)
 - [6.3 Glassfish - org.osgi.framework.BundleException](#)

You need a running ASAM ODS Server with a compliant database configured with an ASAM Application Mode. E.g.

- Peak Solution:
<http://www.peak-solution.de/de/produkte-leistungen/versuchs-messdatenmanagement/softwareloesungen/peak-ods-server/>
Note: you also need the notification service plugin for Peak ODS Server
- HiQSoft: <https://www.highqsoft.com/de/avalon-asam-ods-server/>

Or another compliant data source (e.g. PAK adapter).

3 Download Application Artefacts

Download zip file with artefacts from

<http://git.eclipse.org/c/mdmbl/org.eclipse.mdmbl.git/plain/Releases/V0.x/>

File: openMDM_application-<version>.zip

4 Installation

Unpack the zip file “openMDM_application-version>.zip”, you will find the following structure:

- org.eclipse.mdm.nucleus.war
- configuration/
- schema/

4.1 Setup User Preference Service Database

4.1.1 Setup default schema / user (openMDM)

The Preference service stores its data to a relational database. The default name for the schema / user is “openMDM”. For changing the default name please see section [4.1.3 Setup with another schema name](#).

At the moment there are two database products supported and tested:

- Apache Derby DB: Download it from <https://db.apache.org/derby/releases/release-10.13.1.1.cgi>
- Postgres DB: <https://www.postgresql.org/>

Create a schema / user “openMDM” on your database and use the scripts provided in the zip file to create the table:

```
schema/org.eclipse.mdm.preferences/*.sql
```

Other database products supported by EclipseLink may also work, but are neither tested nor supported by the mdmbl project.

4.1.2 Configure the JDBC resource and its dependent JDBC Connection in your Glassfish

The database connection is looked up by JNDI. The default JNDI name for the JDBC resource is set to jdbc/openMDM.

The JDBC resource and its dependent JDBC Connection Pool have been created and configured within the Glassfish web administration console or through the asadmin command line tool.

Configuration:

- start your database: (if derby DB is used: `> asadmin start-database`)
- start Glassfish: `> asadmin start-domain`
- go to your Glassfish web administration console:
if your Glassfish installation is on your machine the URL is normally:
<http://localhost:4848/>
- Menu Item: JDBC-> JDBC Connection Pools -> new
 - poolname: <mypool_name>
 - Resource Type: javax.sql.DataSource
 - Database Driver Vendor: Derby or Postgresql
 - -> next
 - set properties: User, DatabaseName to openMDM, set password
 - -> 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

4.1.3 Setup with another schema name

“openMDM” is the default schema name, if you have to change it you have to change it in the configuration and rebuild the code. Please look at the following documentation:

https://wiki.eclipse.org/images/e/e0/GettingStarted_mdmbi.pdf

Note: this will be changed in one of the following releases

4.2 Install 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

For Peak ODS Server add the following line to

```
$odsserver_root/cfg/server.properties:  
JMS_FORWARDER.PORT=8089
```

For disabling the Elasticsearch feature, see section [4.4 Configure property files](#)

4.3 Deploy web application to glassfish

Deploy the application (org.eclipse.mdm.nucleus-<version>.war, included in the zip file) on your running Glassfish server

=> do it e.g. via the admin console at <http://localhost:4848/> and adapt the Context Root and the Application Name to “org.eclipse.mdm.nucleus”.

4.4 Configure property files

- start your Glassfish server
- copy the content of the extracted /configuration folder to
\$GLASSFISH_ROOT/domains/domain1/config

- Configuration file: `org.eclipse.mdm.property/global.properties`
To enable globally the freetext search set the parameter: `freetext.active=true`
- 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)
 - 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)
 - If you do not want to use 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>
  <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">true</param>
  <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>
```

- restart the application server (Glassfish)

4.5 Install and configure Glassfish Login Realm

The realm is configured in a standardized way. Configure your LDAP, AD or other systems according to the glassfish documentation:

<https://javaee.github.io/glassfish/doc/4.0/security-guide.pdf>

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

4.6 Headless Deployment

The default configuration for the authentication method is set to "FORMULAR" . 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>FORMULAR</auth-method>
```

to

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

5 Start application

- start ORB (`$JAVA_HOME/bin/orbd -ORBInitialPort 2809`) (skip this if your ODS Server is starting this implicitly, e.g. you are using Peak ODS Server with no NAMESERVICE specified in the server.properties or another)
- start the database for the ODS Server (if necessary)
- start the ODS server
- start Elasticsearch
- start database for the User Preference Service (e.g. derby DB: `> asadmin start-database`)
- start Glassfish: `> asadmin start-domain`)

Change to your browser URL is e.g. <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 and login.

6 Problems and Solutions

6.1 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

6.2 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>

6.3 Glassfish - `org.osgi.framework.BundleException`

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>

6.4 Glassfish - java.lang.NoSuchMethodError

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.

6.5 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 Troubleshooting

Look into the Logfiles:

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