Replication Manager Installation Manual

1.1.5

HORIZON *2020*

	Short title
	Replication Manager Installation Manual

Long title

Replication Manager, Installation and configuration instructions

Short description

This document describes the installation and configuration of the Replication Manager software, part of the CDI/RSM ordering system of SeaDataNet.

Author	Working group
S. Brégent, S. Crouzille	

Dissemination	Copyright terms
Public	

History

Version	Authors	Date	Comments
1.0.13	S. Brégent	17/12/2018	First version
1.0.29	S. Brégent	07/05/2019	Updates for RM version 1.0.29
1.0.30	S. Brégent	13/06/2019	Update IM IP and port
1.0.31	S. Brégent	17/06/2019	Update version
1.0.32	S. Brégent	29/06/2019	Fix Maris IP Add new IP for EUDAT Add available RM ports
1.0.36	S. Brégent	12/07/2019	Add proxy configuration (§3.3.3)
1.0.37	S. Brégent	08/08/2019	Update java requirements (§2.2): RM supports openJDK
1.0.39	S. Brégent	26/08/2019	No change
1.0.41	S. Brégent S. Crouzille	18/11/2019	Remove proxy host and port from RMConfiguration.properties file (§3.3.2) Add troubleshooting about 403 errors with CORS request Add new IP for EUDAT Fix importmanager.seadatanet.org URLs (use https)
1.0.42	S. Brégent	05/12/2019	No change
1.0.43	S. Brégent	06/12/2019	No change
1.0.44	S. Brégent	06/02/2020	Add backup/restore procedure in §4.4 and §4.5
1.0.45	S. Crouzille	30/04/2020	Update Maris IP. Add Update procedure.
1.0.46	S.Crouzille	08/07/2020	No change
1.0.47	S.Crouzille	20/08/2020	No change

1.0.48	S. Crouzille	15/01/2021	Add possibility to change the Octopus jar file.
1.1.0	S. Crouzille	15/02/2022	Add troubleshooting to force vocabulary updates. Add new IM IP.
1.1.1	S. Crouzille	11/05/2022	No change
1.1.2	S. Crouzille	07/11/2022	Add the modus2_database_timeout in the configuration.
1.1.3	L. Bruvry-Lagadec	08/03/2023	No change
1.1.4	S. Crouzille	09/01/2024	Change EUDAT IP.
1.1.5	S. Crouzille	15/05/2024	No change

The current document can be found at:
https://www.seadatanet.org/Software/Replication-Manager

Reference documents

[1] *SeaDataNet Datafile Formats :* Lowry Roy, Fichaut Michèle, Schlitzer Reiner, Maudire Gilbert, Brégent Sophie (2018). **SeaDataNet. Datafile formats. ODV, MEDATLAS, NETCDF**. DELIVERABLE D8.5.
https://doi.org/10.13155/56547

[2] **Replication Manager User Manual**: Brégent Sophie :

https://www.seadatanet.org/Software/Replication-Manager

Table of contents

1. **Introduction**

The Replication Manager replaces the Download Manager in the new SeaDataCloud ordering system. **What does not change:**

The coupling table is still the way to organise data declaration, using moduses 1 to 3.

What's new:

- the Replication Manager is a unique Tomcat web application (no more batches)
- it communicates with three other components: MARIS Import Manager (IM – New component) and Request Status Manager (RSM – already existing component) and the EUDAT (cloud)
- it manages CDIs and unrestricted data ingestion: unrestricted data are stored in a cloud
- it manages restricted data orders (no more orders on unrestricted data, as they are available on the cloud)
- it manages data versions

We strongly recommend to use modus 3 instead of modus 1 for SeaDataNet formats (MedAtlas, ODV and SDN NetCDF). Cf Replication Manager User Manual [2]

2. Requirements

2.1. Operating system

The Replication Manager can be installed on both the Windows and Linux platforms.

2.2. Java version

Check if Java is available by executing command 'java –version' in the Windows command line or in a Linux terminal.

Due to changes in the Java Oracle licence, the RM is not supported on Java Oracle 9 and higher.

Is is supported with Java Oracle 8 (1.8u151 at least, but lower than 1.9).

One of the recommended Java SE JRE can be downloaded here:

https://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html

The RM is now supported with **OpenJDK** 11 (default version in Debian current stable distribution "Buster")

It can be downloaded here:

https://jdk.java.net/archive/

2.3. Tomcat

The Replication Manager requires at least Tomcat 8.5.31.

Download the Tomcat web server at https://tomcat.apache.org/download-80.cgi

The tomcat installation is not in the scope of this document. Please follow the Tomcat installation instructions from the official documentation:

https://tomcat.apache.org/tomcat-8.5-doc/index.html

2.4. Network access

The Replication Manager server should be accessible from the internet on port 8080 (default Tomcat port), 80 or 443.

The server shall be able to connect to:

IM:

importmanager.seadatanet.org (IP 77.87.163.212), port 443

BODC vocabulary server:
vocab.nerc.ac.uk, port 80

CSR list:
seadata.bsh.de/isoCodelists/sdnCodelists/csrCodeList.xml

The server shall allow connections from:

IM:

77.87.163.227 (old IP)
195.148.30.154 (new IP)

Maris office (for helpdesk access):

81.173.61.180

EUDAT:

195.148.30.56, 193.166.25.167 and 195.148.30.123

The Replication Manager accesses to certain services (to retrieve CSRs, BODC ...). Many of those services use HTTPS protocol, but there's no need to force any certificate, the RM should be able to use these services as is.

2.5. Internet browser

The Replication Manager Dashboard can be accessed via a web browser.

Supported browsers:

- Google Chrome >= 59.0.3071

- Mozilla Firefox >= 52.2.0

2.6. Supported databases

The Replication Manager can handle data and/or coupling table from a database, see [2].

The drivers loaded by the Replication Manager are displayed on the log file at each start.

Available drivers are:

- Oracle: ojdbc14 v12.2.0.3.0

- MySQL: mysql-connector-java v5.1.26

- PostgreSQL: jdbc3 v9.1-901

- MS SQLServer :   sqljdbc4 v4.0

- Jtds: jtds v1.2.6

Support for database 9 and 11 may require the use of an additional JVM parameter.
See §6.1 in Troubleshooting chapter.

2.7. About the RM configuration

The chapter §3.3.2 describes how to configure the RMConfiguration file and where are stored information on data paths, workflow directories paths…

The choice of these paths require to understand how the Replication Manager works. It concerns the data managers (where are the data…) as much as the technical teams (how much disk space should be used…).

This information is described in the **Replication Manager User Manual** [2].

3. Installation

This section describes how to install the Replication Manager.

If data preparation is not already done, follow the Replication Manager User Manual [2] before performing installation.

3.1. Install Sheet

This sheet aim is to gather all important information about your RM installation. Please fill it during the installation, this will be useful for assistance.

ADMINISTRATIVE INFORMATION

Partner name *Name of the Data Centre*

EDMO code *EDMO code of the Data Centre*

Technical contact *The person who will be notified by the monitoring system that the Replication Manager is no longer replying to the monitoring requests thus it is considered offline*

Name:
email:

Approbation contact *(for data requests approbation). E-mail address of the contact at the Data Centre that will deal with data requests. Login is the SeaDataNet ID obtained from the AAA authority server.*

Name:
Login:
email:

TECHNICAL INFORMATION *about the server on which the RM is installed*

Operating system

Java version

Tomcat version

Tomcat port

Server IP address and host name *Public IP address and host name if any of the server on which the RM is installed*

IP address:
Host name:

Proxy *Is your network behind a proxy?*

YES / NO

3.2. Installation

Download the latest release of the Replication Manager from

https://www.seadatanet.org/Software/Replication-Manager

Copy the Replication Manager war file (ReplicationManager.war) in your Tomcat webapps directory (overwrite the previous one if exists). The application will be automatically deployed by Tomcat.

By default, the access is allowed only for localhost and MARIS. You will see in the §3.3.1 how to modify security filters.

Open a web browser in the machine hosting the RM and check that the application is deployed successfully using the Replication Manager About page:

http://localhost:<port>/ReplicationManager

If a problem occurs, check the Tomcat catalina.out log file (default path is <tomcat_directory>/logs/catalina.out)

Check the configuration in the Replication Manager Summary page:

http://localhost:<port>/ReplicationManager/Summary

The result should be the page below:

Figure 1 Default page (Summary) with configuration not set or invalid

3.3. Configuration

The configuration is set in the application RMConfiguration.properties, web.xml and eventually log4j2.xml files.

The modifications written in the RMConfiguration.properties, log4j2.xml and web.xml file are automatically reloaded by the server. The reload takes a few seconds and can be checked in the log:

```
****************** RMListener contextDestroyed ******************
[…]
****************** RMListener INITIALIZATION ******************
```

3.3.1. Security configuration (web.xml file)

The web.xml configuration file is located in the RM WEB-INF directory:

 <tomcat_webapps>/ReplicationManager/WEB-INF/web.xml

The RM uses IP filters to allow or refuse processing requests from clients.

See also https://tomcat.apache.org/tomcat-8.0-doc/config/filter.html#Remote_Address_Filter

The web.xml file contains

- filters definitions (<filter> tags)
- filters mappings(<filter-mapping> tags)

A filter is applied on each RM access:

- Dashboard pages ("dashboard" section)
- Logs (RM LOG and RM CHECKER LOG sections)
- Files harvesting (HARVESTING section)
- RM API calls (API section)

Three filters are defined and should be sufficient:

- SDNSystem (MARIS, EUDAT, localhost)
- MARIS (also contains localhost)
- EUDAT (also contains localhost)
- Local

Local contains only the localhost IP (i.e. allow only the machine on which the RM is installed).

You can add local IPs, or a mask representing your local network, separated by a pipe (|).

Each filter is identified by its name (<filter-name> tag).

```
<filter>
        <filter-name>local</filter-name><!-- local only; add you local IP or network mask
here -->
        <filter-class>org.apache.catalina.filters.RemoteAddrFilter</filter-class>
        <init-param>
                <param-name>allow</param-name>
                <param-value>127\.\d+\.\d+\.\d+|::1|0:0:0:0:0:0:0:1</param-value>
        </init-param>
</filter>
```

Add your local IP address (or network mask) to all filters. Example: if your IP is <mark>192.168.1.2</mark>, add it in local only filter as below:

```xml
<filter>
      <filter-name>local</filter-name><!-- local only; add you local IP or network mask
here -->
      <filter-class>org.apache.catalina.filters.RemoteAddrFilter</filter-class>
      <init-param>
            <param-name>allow</param-name>
            <param-value>127\.\d+\.\d+\.\d+|::1|0:0:0:0:0:0:0:1|192.168.1.2</param-value>
      </init-param>
</filter>
```

Then add also this IP in the MARIS, EUDAT ans SDNSystem filters.

3.3.2. RM Configuration file (RMConfiguration.properties)

This file is located in the <tomcatWebApps>/ReplicationManager/WEB-INF/classes directory.

The RMConfiguration.properties file contains the RM configuration that shall be customised by each Data Centre.

If you need to change a parameter value in the RMConfiguration.properties file, just refresh the Dashboard Summary page after you have modified the file.

If you need to change some configuration elements outside the configuration file (create a directory, for example), you will need to launch the configuration checker manually, using the "RELOAD" button on the Dashboard Summary page.

The directories paths are important elements of this configuration. Directories must exist. The figure below shows what are these paths and how they can be organised.
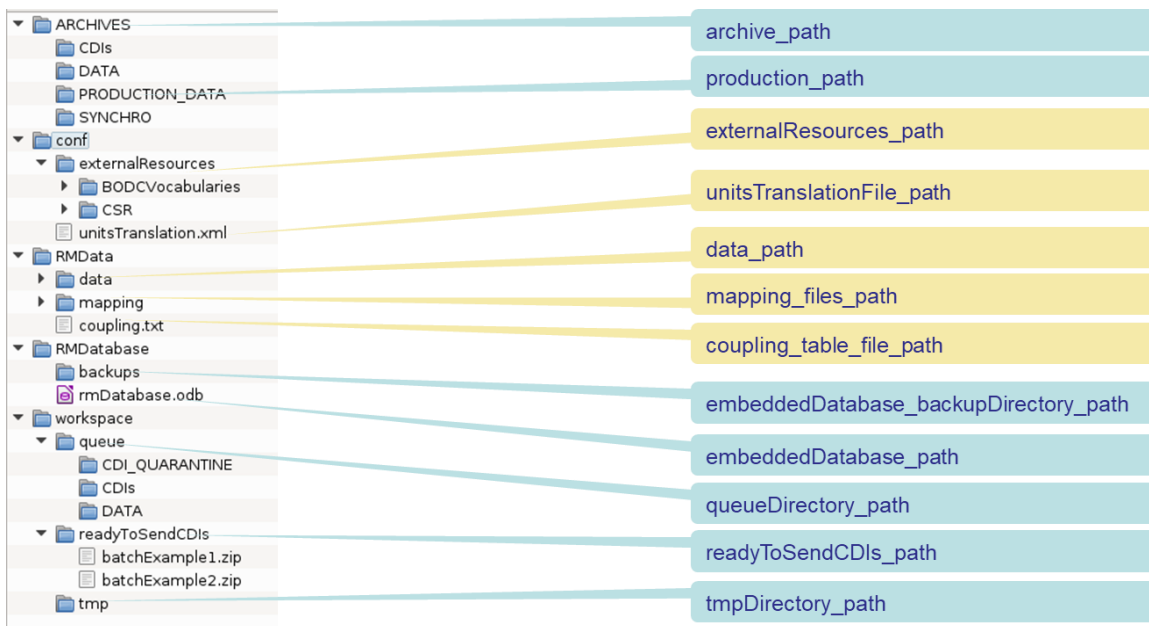
Figure 2 Organisation example

All paths shall be absolute paths.

The resulting RMConfiguration.properties file is shown below.

In red: fields that you need to modify

In brown: fields that you may want to modify (optional)

```
#==================================================================
# REPLICATION MANAGER CONFIGURATION
#==================================================================
#TEST MODE : 0 is production, 1 is test
test_mode=1


# EDMO CODE
edmo_code=xxxx


# POPULATE ENABLED: 0 is false, 1 is true
# This is now de-activated, populate authorization is controlled only by MARIS.
#enable_populate=0



#==================================================================
# WORKFLOW ORGANIZATION
```

```
#================================================================

# readyToSend CDIs path: directory where the metadata CDIs zips will be stored for ingestion
readyToSendCDIs_path=/home/ReplicationManager/workspace/readyToSendCDIs


# tmpDirectory: directory for temporary files (where data files are created)
tmpDirectory_path=/home/ReplicationManager/workspace/tmp


# queueDirectory_path: where cdi and data files are queued during ingestion process, waiting for uploading and archiving
queueDirectory_path=/home/ReplicationManager/workspace/queue


# ARCHIVES
# archive path : Path of the directory where CDIs and DATA files are archived after ingestion process.
# This directory will contain 2 sub-directories (automatically created):
# "CDIs" containing metadata files zips copy after their ingestion
# "DATA" containing corresponding data files zips copy after their ingestion
archive_path=/home/ReplicationManager/ARCHIVES


# PRODUCTION DATA FILES
# Path of the directory where versioned restricted data files are stored
# Theses files can be requested from the RSM restricted orders
Production_path=/home/ReplicationManager/PRODUCTION


# Optional: path to a directory to use for Harvest (where zips are stored to be downloaded by maris and Eudat)
# By default, the RM will use the WEB-INF tomcat application directory
# If you do not want the RM application to write in WEB-INF directory, set your path here
# three sub-directories will be automatically created: HARVEST_METADATA, HARVEST_DATA, HARVEST_ORDERS
#harvest_root_path=


# RM DATABASE
# embedded database directory path (path is free, directory must be readable and writable by the RM, file extension shall
be .odb - file will be created on first launch)
embeddedDatabase_path=/home/ReplicationManager/RMDatabase/rmDatabase.odb
# embedded database backup directory (path is free, directory must be readable and writable by the RM)
embeddedDatabase_backupDirectory_path=/home/ReplicationManager/RMDatabase/backups


#================================================================
# DATA ORGANIZATION
#================================================================
# data path : where original data files are stored (modus 1 and 3)
data_path=/home/ReplicationManager/RMData/data


# MODUS 2 database timeout (in seconds)
modus2_database_timeout=240


# mapping files path (modus 2)
```

```
mapping_files_path=/home/ReplicationManager/RMData/mapping


#COUPLING TABLE
# Coupling table type : 0 if coupling table is in file or 1 if coupling table is in database
coupling_table_type=0


# for coupling in file only : coupling table file path
coupling_table_file_path=/home/ReplicationManager/RMData/coupling.txt


# for coupling in database only : database connexion
#coupling_table_connection=jdbc:oracle:thin:@195.178.224.89:1312:database_name
#coupling_table_user=user
#coupling_table_password=pass
#coupling_table_tablename=theTable


# The RM can update the external resources (BODC vocabs, CSR) periodically, setting these two following parameters.
# If externalResources_period_days is not set, no update will be performed, and externalResources_period_time won't be read.
# Update period in days
#externalResources_period_days=1
# Update time in H24:MM format (examples: 03:30, 23:25, 15:00 ...), server time.
#externalResources_period_time=23:30


# Path where to store external resources: BODC vocabularies local files, CSR local file
externalResources_path=/home/ReplicationManager/conf/externalResources


# file containing translation between BODC and netCDF units (optional)
#unitsTranslationFile_path=/home/ ReplicationManager/conf/unitsTranslation.xml



#====================================================================
# NETWORK
#====================================================================



# MARIS API
maris_api_url_test=https://importmanager.seadatanet.org/api_v1
maris_api_url_production=https://importmanager.seadatanet.org/api_v1
maris_cms_url_test=https://importmanager.seadatanet.org/v_import_manager_v5/content.asp?screen=4
maris_cms_url_production=https://importmanager.seadatanet.org/v_import_manager_v5/content.asp?screen=4
```

Database connection string examples:

Oracle connection example

coupling_table_connection=jdbc:oracle:thin:@<server>:<port>:<service_name>

Example: jdbc:oracle:thin:@195.178.224.89:1312:database_name

MySQL connection example

coupling_table_connection=jdbc:mysql://10.1.96.214:3306/dm_test

Example: jdbc:mysql://localhost:3306/database_name

MS-SQL Server connection example

coupling_table_connection=jdbc:sqlserver://<server>:<port>;databaseName=<database_name>

Sybase connection example

coupling_table_connection=jdbc:jtds:sybase://<server>:<port>/<database_name>

PostgreSQL connection example

coupling_table_connection=jdbc:postgresql://<server>:<port>/<database_name>

3.3.3. Proxy configuration

If the Replication Manager needs to use a proxy to connect to the internet, you need to configure the proxy options that will be used by the java virtual machine in the Tomcat server.

Create a setenv.bat (windows) or setenv.sh (Linux) file in the Tomcat bin directory, with the following content:

Linux - setenv.sh

```
#!/bin/sh
export JAVA_OPTS="${JAVA_OPTS} -DhttpProxySet=true -Dhttp.proxyHost=proxy.example.org -Dhttp.proxyPort=3128"
echo ${JAVA_OPTS}
```

Windows - setenv.bat

```
set JAVA_OPTS=%JAVA_OPTS% -DhttpProxySet=true -Dhttp.proxyHost=proxy.example.org -Dhttp.proxyPort=3128
echo %JAVA_OPTS%
```

Replace "proxy.example.org" by your proxy host name

Replace "3128" by your proxy host port if different

If you use https, use https options instead or in addition to http:

-Dhttps.proxyHost

-Dhttps.proxyPort

3.3.4. Log configuration (log4j2.xml file)

The log4j2 configuration file is located in the RM WEB-INF/classes directory:

 <tomcat_webapps>/ReplicationManager/WEB-INF/classes/log4j2.xml

The default RM log file path is the log directory in the default tomcat path:

<tomcat_directory>/logs/

You can customise using absolute paths, as shown below.

E.g.: fileName="/home/RM/logs/replicationManager.log"

The default configuration creates a new log file as soon as the current one exceeds 250Mo.

The paths that can be customised are in red, below

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE xml>
<Configuration status="WARN">

        <Appenders>

                <Console name="Console" target="SYSTEM_OUT" follow="true">
                        <PatternLayout>
                                <pattern>%d{ISO8601}{GMT}Z -[RM] - %-5p - %m (%c{3}-%l) %n</pattern>
                        </PatternLayout>
                </Console>


                <!—Replication Manager log →
                <RollingFile name="RollingFile" fileName="${sys:catalina.home}/logs/replicationManager.log"
                        filePattern="${ sys:catalina.home}/logs/replicationManager.log.%d{yyyy-MM-dd}.%i">
                        <PatternLayout>
                                <pattern>%d{ISO8601}{GMT}Z -[RM] - %-5p - %m (%c{3}-%l) %n</pattern>
                        </PatternLayout>
                        <Policies>
                                <TimeBasedTriggeringPolicy />
                                <SizeBasedTriggeringPolicy size="250 MB" />
                        </Policies>
                         <DefaultRolloverStrategy max="20"/>
                </RollingFile>


                <!—Replication Manager checker log →
                <RollingFile name="RollingFileChecker" fileName="${sys:catalina.home
}/logs/replicationManagerChecker.log"
```

```
        filePattern="${sys:catalina.home }/logs/replicationManagerChecker.log.%d{yyyy-MM-dd}.%i">
            <PatternLayout>
                    <pattern>%d{ISO8601}{GMT}Z -[RM] - %-5p - %m (%c{3}-%l) %n</pattern>
            </PatternLayout>
            <Policies>
                    <TimeBasedTriggeringPolicy />
                    <SizeBasedTriggeringPolicy size="250 MB" />
            </Policies>
             <DefaultRolloverStrategy max="20"/>
        </RollingFile>


    </Appenders>


    <Loggers>
            <Logger name="org.apache.http"        level="INFO">
                    <AppenderRef ref="RollingFile" level="INFO" />
                    <AppenderRef ref="Console" level="INFO" />
            </Logger>
            <Root level="DEBUG">
                    <AppenderRef ref="RollingFile" level="INFO " />
                    <AppenderRef ref="Console" level="INFO " />
            </Root>
            <Logger name="fr.ifremer.rm.common.check"    level="INFO">
                    <AppenderRef ref="RollingFileChecker" level="INFO" />
                    <AppenderRef ref="Console" level="INFO" />
            </Logger>
    </Loggers>


</Configuration>
```

3.3.5. Configuration validation

Once the configuration is done, go to the RM dashboard *Summary* page and click on the "RELOAD" button:

http://<host>:<port>/ReplicationManager/

The Configuration in the *Summary* page shall be valid:

Configuration

RM configuration is valid

RELOAD configuration (last reload: 20190712T16:55:39Z)

TEST Import Manager call

If the configuration is not valid

- check the error messages in the "Custom Parameters" table (*status* column)
- fix the parameter(s) value(s) in the RMConfiguration.properties file if needed
- check the files and directories
- click on RELOAD button

Once the RM

3.4. Checks

To check the RM status, use the following GET HTTP request:

http://RMhost:RMport/ReplicationManager/api/api_v1/status

The response should be like

```
{
  "datetime":"20190711T15:51:34",
  "configIsValid":"true",
  "edmo_code":"3367",
  "version":"1.0.33"
}
```

To check the connection to the Import Manager, click on the Test Import Manager call button, in the Summary page. The result should be a pop-up like below:

[SUCCESS] datetime: 20190712T19:20:57 / request_id: 6c0250a7-92ab-434e-97d0-86637ae517e3

OK

4. Maintenance

Information about the system is available on the *Summary* page, "System" section:

- Operating system
- Tomcat version
- Java version
- RM version
- Octopus version

Several maintenance tools are available from the *Summary* page, "Maintenance" section:

- logs access
- BODC and other external resources versions and manual update launcher
- Check tools
- database tools

Figure 3 Summary Page, Maintenance section

4.1. Log access

In the dashboard *Summary* page, the "Logs" section gives URLs to access to the current RM log and RM Checker log.

Figure 4 Logs access

The log paths are given in the log4j2 xml configuration file (§3.3.3).

Direct URLs are:

http://<host>:<port>/ReplicationManager/RMLog

http://<host>:<port>/ReplicationManager/RMLogChecker

Older logs can be accessed using URL parameters date and index:

- Date = the log file date
- Index: the log file number in this day

The logs pages are not auto-refresh. Click on F5 to refresh.

The log section RM logs list opens a page listing all the existing log files.

4.2. External resources

This section displays the BODC vocabularies version and the CSR list file last update date.

To update all, click on the "UPDATE" button.

Note that you should also define an automatic update using the externalResources_period_days and externalResources_period_time parameter in the RM configuration file (§3.3.2)

Figure 5 external resources maintenance

No message is displayed on the page, but the updates can take a while.

Wait about 10 minutes and refresh the page to display the new versions.

4.3. Checks

The RM dashboard Checks section offers the following functionalities:

- Local check : check coupling table lines and mapping files if exist)
- Local versus Central check: Launch comparisons between

- o coupling and central catalogue (check if all coupling table entries are present in the catalogue)
- o local embedded database and central catalogue (check if all catalogue entries are present in the local embedded database)
- Population: populates the system (cf. §5.1)

Checks

Local check   Process a check on the coupling table (results available in the RM checker logs)

Local versus Central check   Process comparison between :
    o coupling table and central catalogue (check if all coupling table entries are present in the catalogue)
    o local embedded database and central catalogue (check if all catalog entries are present in the local embedded database)

POPULATION:
Population allowed: false
last date: Mon Nov 18 14:31:52 CET 2019
in progress: false

Download last synchro/population results as csv: synchro.csv

WARNING: The population should be launched only on first RM installation!

See [2] Replication Manager User Manual for more details.

4.4. Backup

This chapter describes the elements that must be backed up in order to be able to restore a functional Replication Manager server in case of crash. The list described here do not include the elements concerning the original data (files, modus 2 database, modus 2 mapping files). See [2] **Replication Manager User Manual**.

4.4.1. Configuration files

The configuration files contains all the paths and eventually specificities that you have chosen for the ReplicationManager installation. If the RM crashes, it could be very useful and a significant time saver to have these files.

We **very strongly** recommend to backup the following configuration files:

- <tomcat_webapps>/ReplicationManager/WEB-INF/web.xml
- <tomcatWebApps>/ReplicationManager/WEB-INF/classes/RMConfiguration.properties
- <tomcat_webapps>/ReplicationManager/WEB-INF/classes/log4j2.xml

If you use a setenv file for proxy configuration, it is strongly recommended to also backup it (see 3.3.3).

4.4.2. Database

The RM embedded database is stored in an odb file. When the server is running, an associated odb$ file exists.

The backup of the database is MANDATORY to be able to restore the system.

The database can be properly backed up manually or automatically:

Manually from the *Summary* page, in Database section, using the database button:

Automatically: a backup can be scheduled with an external script:

```
curl -d '{"request_id": "0", "edmo_code": "3367", "datetime": "20170712T15:33:11", "api_function": "backup_database", "version": "1", "test_mode":"false"}' -H 'Content-Type: application/json' http://localhost:8080/ReplicationManager/api/api_v1
```

Replace 3367 by your EDMO code, and localhost:8080 by your RM hostname:port.

Datetime can be an old date, this is not a problem.

We strongly recommend an automatic backup.

In both cases, a backup will be created in the directory indicated by the *embeddedDatabase_backupDirectory_path* property in the RMConfiguration.properties file.

The backup is a odb file, created in a subdirectory named from the date and time YYYYMMDDHHmm.

4.4.3. Workflow and production directories

Directory	property of the RMConfiguration.properties file.	Backup policy
PRODUCTION If your data centre distributes data with restricted access, the only copy of these data files are stored in the PRODUCTION directory (they are not in the Cloud).	*production_path*	MANDATORY

Ready to send CDIs The place where the data centre managers put the zipped CDIs to submit. If not saved, the batches will have to be put again in this directory (check with the data manager if he keeps a copy on his side).	*readyToSendCDIs_path*	Recommended
Queue The place where the zips are stored when waiting for enter in the ingestion workflow, or after having be put in Quarantine. If not saved, the batches will have to be submitted again.	*queueDirectory_path*	Recommended
Archive Copy of the submitted metadata files and unrestricted data files. This archive is used to keep the history of the submissions.	*archive_path*	Optional

4.5. Restore

1/ Download the latest Replication Manager

2/ Install the new Replication Manager following the §3.2 and §3.3, using the configuration files that have been saved (§4.4.1).

WARNING: the configuration files may have changed from a version to another. If new installation version is newer than the backup version, prefer copy/paste properties from the backup file to the new one than override the file.

3/ stop the Tomcat server

4/ restore the backup directories and database

5/ start the Tomcat server

4.6. Update

The Update process is globally the same as the Restore one, except that it is not required to backup the embedded database (it's recommended though).

It's not necessary to stop/start the server too: once updated, go to the Summary page, and run a configuration reload (a server restart is also recommended though).

4.7. Update Octopus

You can update the Octopus JAR file used in the RM, in order to use its new functionalities without reinstalling the RM.

To do so:

1. Get the last version of the Octopus JAR file.

2. Shutdown the RM instance.

3. In the RM directory, go to ./WEB-INF/lib.

4. Find the current Octopus JAR file (named "octopus-x.y.z.jar, depending on the version), and either delete it or move it for backup.

5. Replace it by the new Octopus JAR file.

6. Restart the RM. On the Summary page, you can check the new version of Octopus.

5. Usage

This chapter presents briefly the RM usage. You will find more details in The Replication Manager User Manual (see Reference Document [2]).

5.1. Population (first installation only)

Once the RM is correctly configured FOR THE FIRST TIME IN YOUR DATA CENTER, it has to be populated using metadata and data already existing before RM installation

By default, to avoid accidents, the population is not allowed. It has to be allowed by the CDI support.

Once the population is allowed, you can run the process by clicking on the "POPULATE" button.


```
POPULATION:
Population allowed: true
last date: Mon Nov 18 14:31:52 CET 2019
in progress: false
  Populate    populate the RM (WARNING: if the system is already populated, it will be reset)
```

Figure 6 Population button

The population process is automatic. It

- Calling the IM to ask for a catalogue containing all restricted entries

- Download the catalogue from the IM

- Save metadata information in the RM embedded database

- Generate and store in ARCHIVE directory the restricted data files

The metadata CDI files and unrestricted data are stored on the cloud by the Import Manager.

During and after the Population process, you can see the local_cdi_ids already processed in the LOCAL_CDI_Ids page.

The population can end with errors. The list of errors is available from the sychro.csv file in the Summary page Checks section.

In all cases, the authorization must be asked again to the CDI support to launch a new population process.

5.2. New metadata and data submission

The "Ready To Send" directory is the directory defined by the "ReadyToSendCDIs_path" parameter in the RMConfiguration.properties file. This path can be read from the S*ummary* page, in the "Custom Parameters table".

Zips shall be placed in the "Ready To Send" directory, so they become visible in the *Batches In Progress* page.

From this page, Data Centre Manager can select and submit batches. He can follow the workflow using "Batches cancelled", "Batches in production" and "Local CDI Ids" pages.

Figure 7 Batches ready to send

Choose the CDIs batches that you want to submit (by clicking on the rows) , then click on the "Submit" button.

The batch will start the automatic workflow:

- Local preparation   -> in queue
- Ingestion   -> Current batch (one at a time)
- Workflow ended   -> cancelled or in production

At any time, click on the batch name to display a detailed page.

5.3. Restricted data requests process

Incoming restricted data requests will automatically be processed by the RM.

Logs shall be accessed in log file (no information in the dashboard).

6. Troubleshooting

6.1. Cannot load coupling table in an Oracle database

This problem has been observed on a machine using openJDK 8 and Oracle 11.

The following error is appears in the Replication Manager log:

java.sql.SQLException: ORA-00604: error occurred at recursive SQL level 1 ORA-01882: timezone region not found

The problem is known in Oracle:

https://www.ibm.com/support/pages/connect-oracle-9i-and-11g-using-ojdbc6jar-driver

Solution:

Add a JVM argument in the command line:
  `-Doracle.jdbc.timezoneAsRegion=false`

Create a setenv.bat (windows) or setenv.sh (linux) file in the tomcat bin directory, with the following content:

```
echo Setenv Configuration
set JAVA_OPTS="-Doracle.jdbc.timezoneAsRegion=false"
echo JAVA_OPTS: %JAVA_OPTS%
```

Warning: you may have created this file for proxy configuration (§3.3.3).

In this case, keep the proxy information and add the oracle parameter:

```
echo Setenv Configuration
set JAVA_OPTS="-DhttpProxySet=true -Dhttp.proxyHost=proxy.example.org -Dhttp.proxyPort=3128 -Doracle.jdbc.timezoneAsRegion=false"
echo JAVA_OPTS: %JAVA_OPTS%
```

replace "proxy.example.org" by your proxy host name
replace "3128" by your proxy host port if different

6.2. 403 errors with CORS request

If you experiment 403 ("Forbidden") HTTP responses, you may change the CORS configuration in the Tomcat web.xml file like this:

```xml
<!-- CORS filter -->
<filter>
    <filter-name>CorsFilter</filter-name>
    <filter-class>org.apache.catalina.filters.CorsFilter</filter-class>
    <init-param>
        <param-name>cors.allowed.origins</param-name>
        <param-value>*</param-value>
    </init-param>
    <init-param>
        <param-name>cors.allowed.methods</param-name>
        <param-value>GET,POST,HEAD,OPTIONS,PUT</param-value>
    </init-param>
    <init-param>
        <param-name>cors.allowed.headers</param-name>
        <param-value>Content-Type,X-Requested-With,accept,Origin,Access-Control-Request-Method,Access-Control-Request-Headers</param-value>
    </init-param>
    <init-param>
        <param-name>cors.exposed.headers</param-name>
        <param-value>Access-Control-Allow-Origin,Access-Control-Allow-Credentials</param-value>
    </init-param>
</filter>
```

6.3. Force vocabulary updates

In some specific configurations, the BODC vocabularies are not updated correctly and you cannot see new terms that are already included in the vocabularies.

If this happens, follow these 2 steps:

1. delete the file [externalResources_path]/BODCVocabularies/vocabularies.xml

2. update the vocabulary lists: Summary > click the UPDATE button near "External Resources".

If this still does not resolve your problem, it is recommended to force the vocabulary update following these 2 steps:

1. delete all files in the [externalResources_path]/BODCVocabularies

2. update the vocabulary lists: Summary > click the UPDATE button near "External Resources".