Replication Manager User Manual

1.1.4

HORIZON *2020*

Deliverable number	Short title
	Replication Manager User Manual

Long title

Replication Manager User Manual, Management at Data Centre

Short description

This document describes the Replication Manager functionalities and how to prepare your data that will be ingested by the Replication Manager. It also gives information about how to check the conformity of the coupling table and the consistency between the coupling table and the CDI metadata catalogue. Troubleshooting is also detailed.

Author	Working group
S. Brégent	

Dissemination	Copyright terms
Public	

History

Version	Authors	Date	Comments
1.0.31	S. Brégent	18/06/2019	First version
1.0.32	S. Brégent	05/07/2019	RM functionalities: ingestion, users requests, administration tools
1.0.34	S. Brégent	11/07/2019	Improve global document organisation, simplify and fix data preparation chapter.
1.0.36	S. Brégent	12/07/2019	Add Harvest URLs in §5.4 and 5.5
1.0.37	S. Brégent	08/08/2019	No change
1.0.39	S. Brégent	26/08/2019	No change
1.0.40	S. Brégent	12/09/2019	No change
1.0.41	S. Brégent	18/11/2019	Replace the term synchronize by check. Add explanations on checks and comparisons.
1.0.42	S. Brégent	05/12/2019	Updates for restricted only mode

1.0.43	S. Brégent	06/12/2019	No change
1.0.44	S. Brégent	19/12/2020	Highlight information about available formats in the XML CDIs files in §4
1.0.45	S. Brégent	25/03/2020	Add human activity steps in the ingestion workflow (§5.4.4)
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	20/08/2020	No Change
1.0.49	S. Crouzille	03/06/2021	Add BODV/MODV formats in MODUS figures.
1.1.0	S. Crouzille	22/02/2022	Add an external resources section, with a troubleshooting to force BODC vocabs updating. Add invalid Modus 2 invalid files archiving. Update menus in description and screenshots.
1.1.1	S. Crouzille	11/05/2022	No Change.
1.1.2	S. Crouzille	07/11/2022	No change.
1.1.3	L. Bruvry-Lagadec	08/03/2023	No change
1.1.4	S.Crouzille	19/01/2024	No change.

The current document can be found on SeaDataNet web site:
https://www.seadatanet.org/Software/Replication-Manager

References

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

[2] Replication Manager Installation Manual: Brégent Sophie : https://www.seadatanet.org/Software/Replication-Manager

Table of contents

Tables and Figures

1. Introduction

The Replication Manager (also called RM) replaces the Download Manager in the new SeaDataCloud system.

What's new:

- the RM is a unique Tomcat web application (no more batches)
- the RM communicates with two other components: Maris (IM/RSM) and EUDAT (cloud)
- data files are generated and archived as soon as the metadata are submitted
- submitted metadata and data are archived locally
- unrestricted data files are stored on a cloud, at EUDAT (restricted remains in Data Centre)
- the RM manages receives and processes only restricted data orders (no more orders on unrestricted data, as they are available on the cloud)

What does not change:

The coupling table is still used to organise data declaration, using modus 1 to 3.

! Data Managers shall keep in mind an important difference:

The Download Manager was generating the data files on the fly, only on user requests.

The Replication Manager generates (ie. copies or creates the files depending on modus) data files only AT THE MOMENT WHEN the CDIs metadata files are submitted. This is the same for MODUS 2 , based on SQL query in database : even if the database changes (new data for a times series for example) the file is created once, and does not change until a new version is proposed.

The data files are then

- versionned
- stored on the cloud (unrestricted) or locally (restricted)

This ensures the consistency between metadata and data distributed in the SeaDataNet infrastructure.

2. Pre-requisites and warnings

The technical instructions concerning the installation are available in the Replication Manager Installation Manual [2]. However some information in this document are also of interest for technical teams. They are highlighted by the following drawing:

Unlike the Download Manager, the Replication Manager store metadata and data, for history (versioning) or restricted data requests. For this reason, data centres have to think about disk space needs:

- metadata and data ingested will be stored (for history)
- restricted data will be stored in each version.

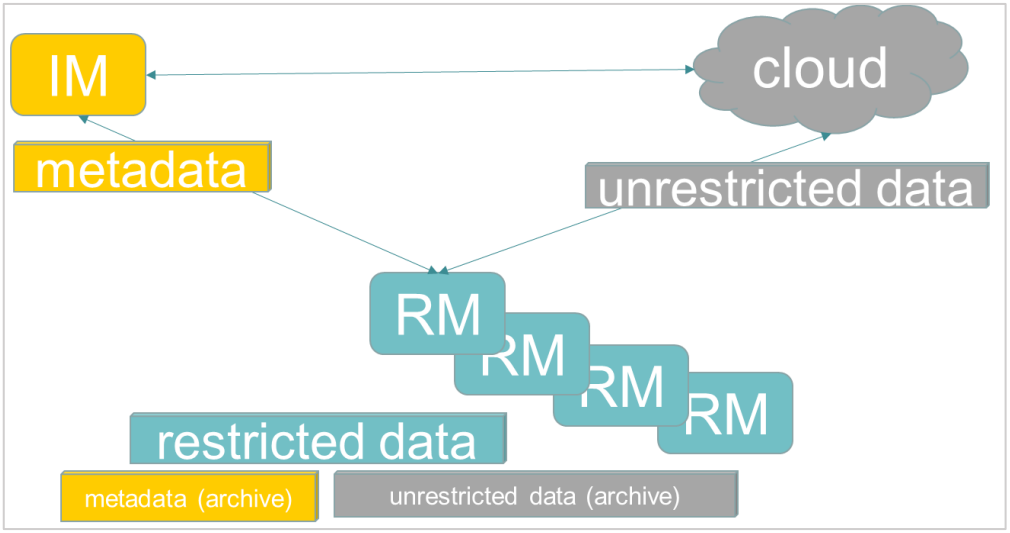


Figure 1 - Metadata and data storage in SDN infrastructure

3. Replication Manager overview

Replication Manager main functionalities are:

 Ingestion of metadata and data

 Processing of restricted data requests from the RSM (MARIS)

 Administration and maintenance tools

Each of these functionalities are accessible via the web interface.

3.1. Ingestion 

The Replication Manager allows the Data Centres to easily submit new or updated metadata and data.

The submission granularity element is called a “batch”: a batch is a set of LOCAL_CDI_IDs submitted at the same time. Metadata files are gathered in a zip, which is the element to be submitted by the Data Centre Manager.

Ingestion of metadata and data is a process composed of different steps, involving the Import Manager at MARIS and the cloud at EUDAT, in the “ingestion workflow”.

At the end of the workflow, metadata (CDI xml files) are stored in the Import Manager (CDI central catalogue at MARIS), unrestricted data files are stored in the cloud, restricted data files are stored locally, in the Data Centre.

3.2. User data requests

The RM listens to incoming restricted data requests from the RSM. It processes these requests in a workflow involving the RMS (MARIS) and the Cloud (EUDAT).

3.3. Administration tools

The RM provides tools concerning both data managers and technical teams:

- system supervision (easily read RM configuration)
- checks on coupling table consistency, locally and against the Import Manager catalogue
- external resources updates (BODC vocabularies files, CSRs…)
- workflow supervision

4. How to prepare data and workflow directories

This chapter describes how to prepare the data that will be managed by the Replication Manager. At the end of this chapter, you will find an example (§4.5).

The table below lists the information about data that will be set in the RMConfiguration.properties file.

Table 1 - Parameters concerning original data

Parameter name (in RMConfiguration.properties file)	Description
data_path	Common root of the data files paths, for modus 1 and 3. Note that the data_path can be empty; then ensure that you use absolute paths in the coupling table.
mapping_files_path	Path of the directory where are stored the mapping files (modus 2)
coupling_table_type	Coupling table type: 0 if coupling table is in a file or 1 if coupling table is in database
coupling_table_file_path	for coupling in file only: coupling table file path
coupling_table_connection	for coupling in database only: database connection string
coupling_table_user	for coupling in database only: database user
coupling_table_password	for coupling in database only: database password
coupling_table_tablename	for coupling in database only: database table name
unitsTranslationFile_path	Optional: Path of the XML file containing units translations to use in SDN NetCDF files “units” attribute.

The externalResources_path directory just needs to be created; the RM will automatically create two sub-directories and fill them.

Table 2 - Parameters concerning external resources

Parameter name (in RMConfiguration.properties file)	Description
externalResources_path	Directory where are stored external resources: • BODC vocabularies files (a directory named BODCVocabularies is automatically created) • CSR files (a directory named CSR is automatically created)

The Replication Manager can manage data using multiple storages:

Modus	Description
1	Data are already in **files** that are stored in a directory, or can be downloaded from **a web service**. Each file corresponds to **one Local CDI ID and will be copied as is by the RM, with no conversion**.
2	Data are stored in a **database**. The corresponding ODV file will be created by the Replication Manager.
3	Data are already in **files** that are stored in a directory, or can be downloaded from a **web service.** Each file contains **one or several LOCAL_CDI_ID and will be splitted by the RM to have one file per LOCAL_CDI_ID**.

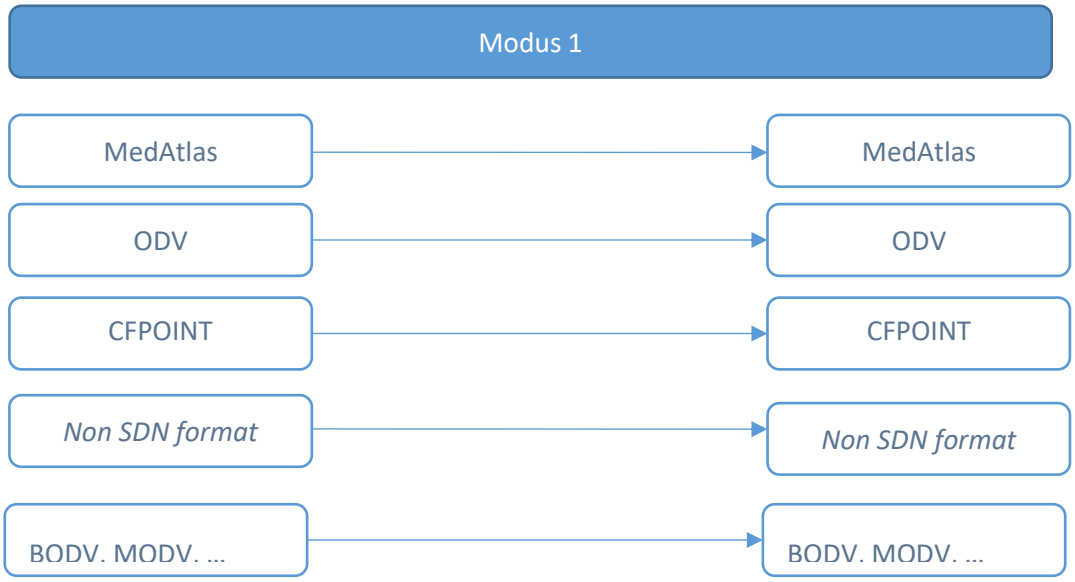
All possible formats should be described in the CDI metadata file. This will allow the RM to convert the data files to all these formats and make them available in the SeaDataNet infrastructure.

4.1. Data in modus 1 or 3

Data in files (modus 1, 3) must be readable by the Replication Manager. The files can be on the same machine or accessible via the network or a web service. It is recommended to back up your data in at least 2 different other locations after any change.

If all the files are gathered in a directory tree, this directory is considered as the “root” directory; it is possible to divide the data files over multiple sub-directories under the root. The latter must be specified as a relative path in the data set file names in the coupling table.

Figure 2 - Data processing in modus 1 and Figure 3 - Data processing in modus 3 describe the processing applied to the data files



copy

Data are already in **files** that are stored in a directory, or can be downloaded from **a web service**.

Each file corresponds to **one LOCAL_CDI_ID**.

SDN format files are only copied (no automatic updates).

- ➔ Use modus 1 for non SDN files (segY, images, seismic…)
- ➔ Please prefer modus 3 for SDN format files (cf. Figure 3 - Data processing in modus 3)

Figure 2 - Data processing in modus 1

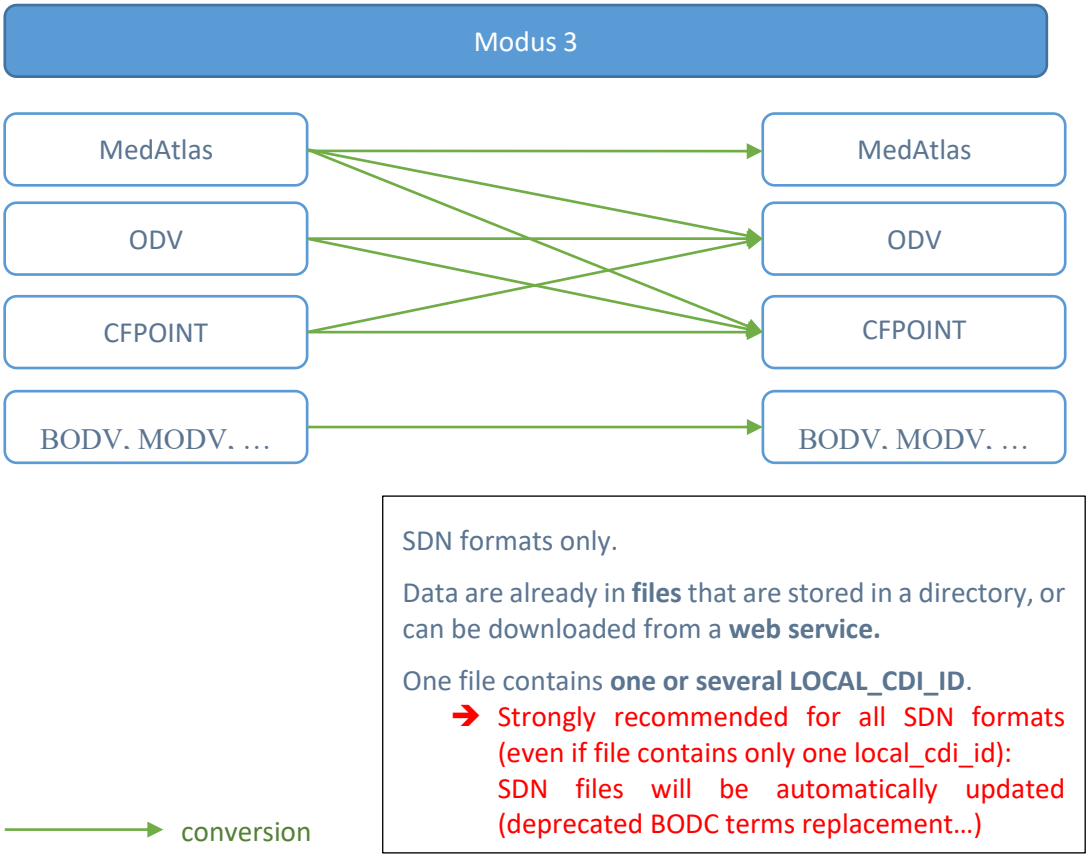


Figure 3 - Data processing in modus 3

4.2. Data in modus 2

Modus 2 consists of the generation of ODV files using

- Data retrieved from a database using the information given in the coupling table (see REF _Ref13246385 \h * MERGEFORMAT
- Table 6 - In case of modus 2 – Retrieving data from a local database
-
-
-)
- Information given in a ODV XML mapping file (An example file *odv_mapping_example.xml* is included in the installation package).

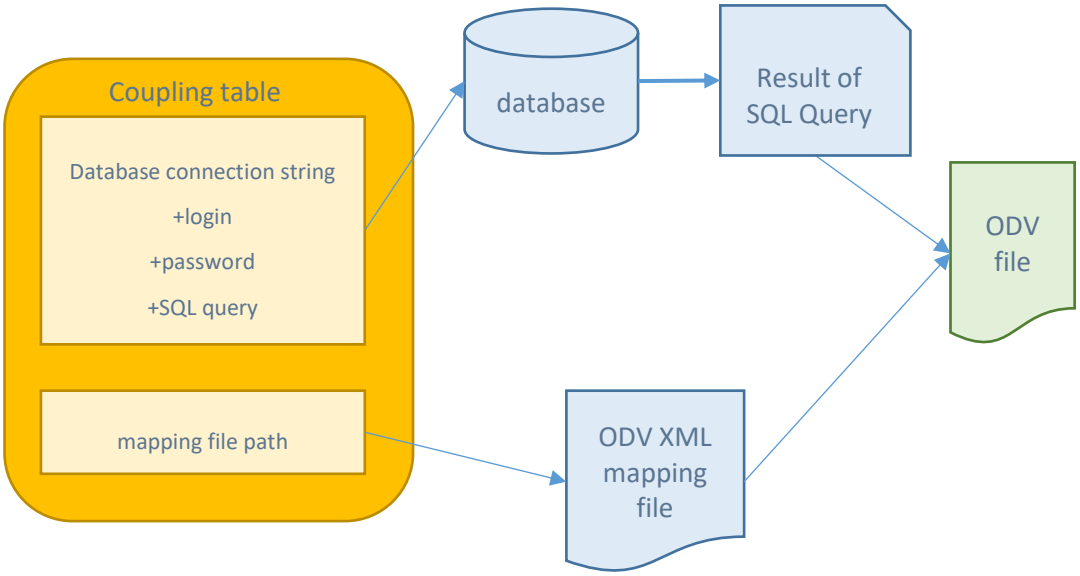


Figure 4 - Modus 2 process

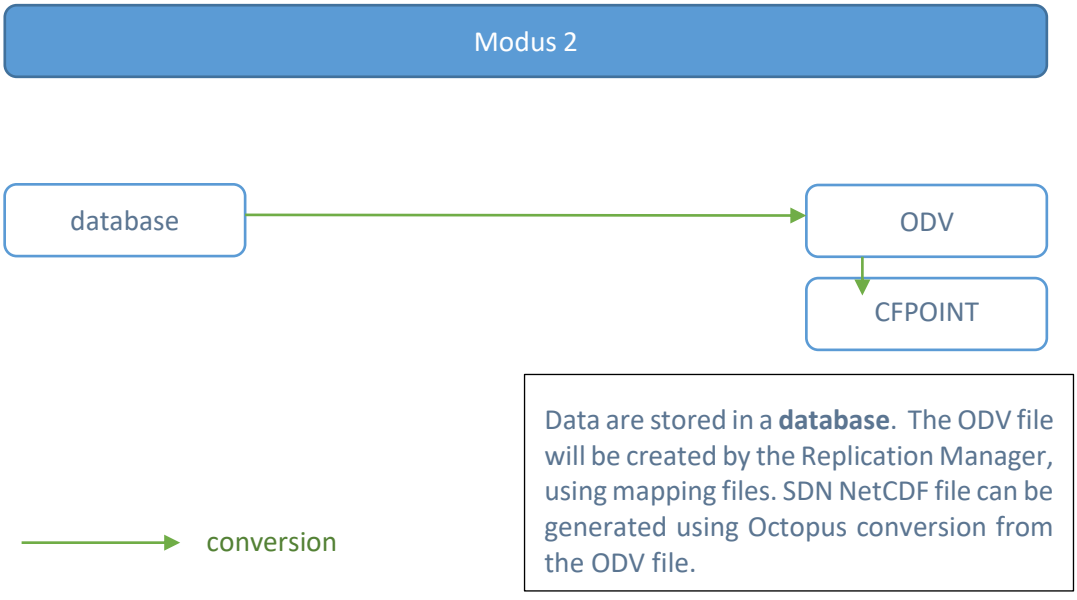


Figure - 5 Data processing in modus 2

All possible formats should be described in the CDI metadata file. This will allow rhe RM to convert the data files to all these formats and make them available in the SeaDataNet infrastructure.

4.2.1. Purpose of the ODV XML mapping file

The ODV XML mapping file describes the mapping between the data retrieved from the SQL query and the metadata/data of the ODV file to write.

For each field of the query result, the mapping file will describe:

- to which metadata or data it corresponds,
- the corresponding P01 and P06 vocabularies (for measurements only).

For example, if the SQL query is:

Select cruise, station, date, longitude, latitude, botdepth, DEPTH, DEPTH_QC, TEMP, TEMP_QC, PSAL, PSAL_QC, sdnreferences FROM tableexample

The mapping file will associate

- cruise, station, date, longitude, latitude, botdepth fields with the mandatory metadata of the SDN ODV fie
- DEPTH, TEMP, PSAL with data columns, by specifying the ODV column title, P01 and P06 mandatory codes, L22 and L33 optional codes
- DEPTH_QC, TEMP_QC, PSAL_QC with Quality Flags columns (L20 codes)

Sdnreferences field will be used to create SDN References metadata.

Checks can be performed on the XML mapping files declared in the coupling table to raise some configuration issues. See §5.3.1 for additional information.

4.2.2. SQL query

The length of the sql_query in the coupling table can be expanded up to 4000 characters maximum. . If 4000 characters are not enough, it is possible to write procedures or functions and or create view.

It must include an ORDER clause according to the ODV format specifications (see reference document [1])

A “row_group” is identified in the mentioned document as “Data rows with exactly the same metadata parameters and that are grouped together”:

- For profile data, “rows within the row_group are ordered by increasing depth or pressure”
- For time series, “rows within the row_group are ordered by increasing time”
- For trajectories, “rows within the row_group are ordered by increasing time”

As a conclusion, the ORDER clause should begin with METADATA columns (to group rows with same metadata) and then with the DEPTH/PRESSURE or the TIME column according to the type of the generated ODV file (to sort lines inside a row_group according to ODV format).

4.2.3. XML notation description

Each metadata and each pair of data value/QC flag from the generated ODV file are declared by a <code> node in the XML mapping file. Each <code> node can have the following attributes: From, to, local, qcflag

These attributes are used to create the SDN mapping lines and the column titles in the ODV file, they are described in more details below.

Attribute name	Declaration
from	Parameter name from the P01 vocabulary and units from the P06 vocabulary MANDATORY for data columns EMPTY for metadata columns from=”OBJECT, UNITS” where OBJECT: URN of a concept from the SeaDataNet P01vocabulary. It must start with SDN:P01:: UNITS: URN of the storage units in the file for the data column in the SeaDataNet P06 vocabulary. It must start with SDN:P06:: **Example:** from="SDN:P01::ADEPZZ01, SDN:P06::ULAA"
to	ODV column title: parameter local name and unit label MANDATORY for data columns MANDATORY for metadata columns to=”SUBJECT, UNITS_TITLE” Where: SUBJECT: The text user to label the ODV file column as it appears in the column row header without the units declaration. It must start with SDN:LOCAL: UNITS_TITLE: The storage units as it appears in the ODV file column row header. It can be empty. **Example:** to="SDN:LOCAL:DEPH, meters"
local	MANDATORY for data columns MANDATORY for metadata columns local=”FIELD_NAME_IN_SQLQuery” Where: FIELD_NAME_IN_SQLQuery: The name of the field in the SQL query that contains the value for the given parameter. It must not be empty. If there is no data value, then the data value column in the ODV file is left blank (value “”) and the quality flag is set to “9” (it means “missing value” in L20). **Example:** local="DEPTH"
qflag	MANDATORY for data columns NOT PRESENT for metadata columns qflag=”QUALITY_FIELD_NAME_IN_SQLQuery” Where: QUALITY_ FIELD_NAME_IN_SQLQuery: The name of the field in the SQL query that contains the SeaDataNet quality flag for the given data value. This must be based upon the SeaDataNet quality flag scale as include in the L20 vocabulary. **All measurement data published in an EU marine research project which uses the SeaDataNet Software Infrastructure must be quality checked and the “qflag” column must be defined.**

	When the attribute “qflag” is absent or its value is empty (qflag=””), then the check value will be always ‘0’ in the resulting ODV file as default value. It means “no quality check” in L20. **Example:** qflag="QC_DEPTH”
instrument	OPTIONAL for data columns NOT PRESENT for metadata columns instrument=”INSTRUMENT_FIELD_NAME” Where: INSTRUMENT_FIELD_NAME: The name of the field in the SQL query that contains the BODC L22 instrument code associated with this parameter. **Example:** instrument="DEPTH_INSTRUM"
fall_rate	OPTIONAL for data columns (only for XBT data) NOT PRESENT for metadata columns fall_rate=”FALL_RATE_FIELD_NAME” Where: FALL_RATE _FIELD_NAME: The name of the field in the SQL query that contains the BODC L33 fall rate code to associate with this parameter. **Example:** fall_rate="DEPTH_FALLRATE"

4.2.4. ODV metadata columns

ODV files must contain 9 mandatory metadata columns.

Mapping on a metadata column is identified in the XML mapping file by <code> node with an empty value for the attribute “from” (from=””).

ODV Metadata column do not have associated QC, then “qflag” attribute should be omitted for these nodes.

The attributes “to” have to be exactly as in this example, including casing and spaces, except for the date column.

Replace the value of each local attribute by the name of the associated field in the corresponding SQL query.

ODV Metadata column	ODV Column title	Code node
Cruise	Cruise	`<code from="" to="SDN:LOCAL:Cruise" local="CRUISE_FIELD_NAME" />`
Station	Station	`<code from="" to="SDN:LOCAL:Station" local="STATION_COLUMN_NAME" />`
Type	Type	OPTIONAL if “Type” node is omitted, default value for Type column in ODV file will be “*”.

		<code from="" to="SDN:LOCAL:Type" local="TYPE_FIELD_NAME" />
datetime	The ISO8601 format mask corresponding to the precision to which the date and time is quoted e.g. 'YYYY-MM-DDThh:mm:ss.sss' or 'YYYY-MM-DD', etc.	<code from="" to="SDN:LOCAL:*YYYY-MM-DDThh:mm:ss*" local="DATE_FIELD_NAME" /> the date pattern must be a valid ISO8601 format mask corresponding to the precision to which the date and time will be written in the output ODV file.
Longitude	Longitude [degrees_east]	<code from="" to="SDN:LOCAL:Longitude, degrees_east" local="LONGITUDE_FIELD_NAME"/>
Latitude	Latitude [degrees_north]	<code from="" to="SDN:LOCAL:Latitude, degrees_north" local="LATITUDE_FIELD_NAME"/>
LOCAL_CDI_ID	LOCAL_CDI_ID	AUTOMATIC retrieved from the value of the current LOCAL_CDI_ID that is being processed by the RM.
EDMO_code	EDMO_code	AUTOMATIC retrieved from the RM configuration file EDMO_code property).
Bot. Depth [m]	Bot. Depth [m]	OPTIONAL if omitted, default value for Bot. Depth [m] column in ODV file will be "0". <code from="" to="SDN:LOCAL:Bot. Depth, m" local="BOTTOM_DEPTH_FIELD_NAME" />

Optional Metadata SDN references

This element can be used to create SDN references (see [1])

```
<!--
Optionnal Metadata SDN references -->
<code from="" to="sdn_reference" local="SDN_REF_COLUMN_NAME" />
```

The column SDN_REF_COLUMN_NAME shall contain all sdn references (sdn_reference XML tags) as a string without separator.

4.2.5. ODV data columns

After metadata columns definition, the data columns mapping has to be declared.

Each mapping for data column is declared like this:

```
<code from="OBJECT, UNITS" to="SUBJECT, UNITS_TITLE"
local="FIELD_NAME_IN_SQLQuery" qflag="QUALITY_FIELD_NAME_IN_SQLQuery" />
```

The columns order in the generated ODV file will be the same as the declaration of their mapping in the XML mapping file.

ODV format requires mandatory columns depending on the variant:

- For physico-chemical and flow-cytometry profile data, “the primary variable is the z-co-ordinate, which for SeaDataNet is either Depth in meters or Pressure in decibars”,
- For physico-chemical and flow-cytometry time series, “the primary variable is time (UT)”,
- For physico-chemical and flow-cytometry trajectories, “the primary variable is the z-co-ordinate, which for SeaDataNet is standardised as Depth in meters”.
- For biological variants, no primary variable is required
- For microlitter variants, no primary variable is required

For biological, flow-cytometry and microlitter variants, other columns are mandatory. Please refer to the related specifications (https://www.seadatanet.org/Standards/Data-Transport-Formats).

Example:

In the mapping file:

<code from="SDN:P01::ADEPZZ01, SDN:P06::ULAA" to="SDN:LOCAL:DEPH, meters" local="depthFieldInQuery" qflag="depthQCFieldInQuery"/>

In the resulting ODV file:

- a SDN mapping line:

  //<subject>SDN:LOCAL:Depth</subject><object>SDN:P01::ADEPZZ01</object><units>SDN:P06::ULAA</units>
- a column entitled DEPH [meters], with values from depthFieldInQuery field
- a QC column entitled QV:SEADATANET, with values from depthQCFieldInQuery field

4.3. Creation of the Coupling Table

The next step is to configure the “coupling table” with the relations between the LOCAL_CDI_ID identifiers and the local data file.

The coupling table can be stored either in a configuration ASCII file called *coupling.txt* or in a database table.

Checks can be performed on the coupling table to raise some configuration issues. See §5.2 for additional information.

Each entry of the coupling table defines a relation between a LOCAL_CDI_ID of the data centre and either:

- a data file or RESTful web service (modus 1 or 3)
- a SQL query returning the data (modus 2)

4.3.1. Coupling table fields

Each entry is defined with the following fields. The first three are common, the other depends on the modus and data type.

Table 3 - Mandatory fields, in all cases

Field name	Description
LOCAL_CDI_ID	The CDI local identifier, as included in the central CDI Directory at the SeaDataNet portal.
modus	Value equals to 1, 2 or 3. See §0. and §4.2

format	Format of the original data file (ODV, CFOINT or MedAtlas, or other for some specific cases)

Table 4 - In case of modus 1 or 3 + Retrieving data files from the shelf

Field name	Description
filename	The path of the original data file, relative to the "root" directory (see data_path parameter in RMConfiguration.properties file and Table 1 - Parameters concerning original data)

Table 5 - In case of modus 1 or 3 + Retrieving data files from a RESTful web service

Field name	Description
filename	The URL where is located the file to retrieve from a RESTful web service. It must start with http:// or https://
proxy (optional)	Proxy URI uses by the Replication Manager when contacting the URL. This parameter is optional.
login (optional)	Login / username to access the web service This parameter is optional.
password (optional)	Password to access the web service This parameter is mandatory when login / username is specified ; otherwise, it is optional

Table 6 - In case of modus 2 – Retrieving data from a local database

Field name	Description
sql_query	SQL query to retrieve from local database all metadata and data, necessary for creating the ODV data file. (see §4.2) See §4.2.4 about XML mapping files, METADATA columns name should be specified in these files. See document [1] for additional information about ODV specifications.
connection_string	**Oracle** connection example *jdbc:oracle:thin:@<server>:<port>:<service_name>* *example: jdbc:oracle:thin:@195.178.224.89:1312:database_name* **MySQL** connection example *jdbc:mysql://10.1.96.214:3306/dm_test* *example: jdbc:mysql://localhost:3306/database_name* **MS-SQL** Server connection example *jdbc:sqlserver://<server>:<port>;databaseName=<database_name>* **Sybase** connection example *jdbc:jtds:sybase://<server>:<port>/<database_name>* **PostgreSQL** connection example *jdbc:postgresql://<server>:<port>/<database_name>* **Note for MS SQL Servers**: if you use Microsoft Windows authentication instead of set the login/password in the config file, you can use the jtds driver: jdbc:jtds:sqlserver://<server>:<port>/<database_name> In this case, login and password will be ignored
login	Login for database user. It cannot be empty except if MS authentication is used.
password	Password for database user. It cannot be empty except if MS authentication is used.
mapping_file	Name of the mapping file to use to generate the file in ODV. This field contains only the filename without path to its directory, as the path of the directory containing the mapping files is given in the RMConfiguration.properties file. See mapping_files_path In Table 1 - Parameters concerning original data.

4.3.2. Coupling table in a configuration file

The coupling table file must be named coupling.txt. Each line defines one entry, fields are separated by a semicolon (;).

NOTE: If a parameter value contains a semicolon (typically, JDBC connection string for a MS-SQL Server database), this value must be declared between two double quotes (").

The tables below describes which fields (ordered) must be used in each case.

In case of modus 1 or 3 + Retrieving data files from the shelf

Field Number	Field name *(see detailed description in §4.3.1)*
1	LOCAL_CDI_ID
2	modus
3	format
4	filename

In case of modus 1 or 3 + Retrieving data files from a RESTful web service

Field Number	Field name *(see detailed description in §4.3.1)*
1	LOCAL_CDI_ID
2	modus
3	format
4	filename
5	proxy (optional)
6	login (optional)
7	password (optional)

In case of modus 2 – Retrieving data from a local database

Field Number	Field name *(see detailed description in §4.3.1)*
1	LOCAL_CDI_ID
2	modus
3	format
4	sql_ query
5	connection_string
6	login
7	password
8	mapping_file

Example of a line in coupling.txt for a pre-processed file retrieved from the shelf (modus 1):
RNODC_Bottle_14828_1;1;ODV;RNODC_Bottle_14828.txt

Example of a line in coupling.txt for a record within a MedAtlas multistations file (modus 3):
RNODC_Bottle_14828_1;3;MEDATLAS;RNODC_Bottles.med

Example of a line in coupling.txt for a record within a ODV multistations file (modus 3):

ABC_Bottle_123_1;3;ODV;ABC_Bottles.txt

Examples of a line in coupling.txt for a pre-processed file retrieved from a RESTful web service (modus 1):

- No proxy and no HTTP-authentication:
  Bottle_1547;1;ODV;http://www.example.com/test_ws/odv/Bottle_1547
- Use a proxy (http://proxy.example.com:3128) and no HTTP-authentication:
  Bottle_1547;1;ODV;http://www.example.com/test_ws/odv/Bottle_1547;http://proxy.example.com:3128
- Use a proxy (http://proxy.example.com:3128) and HTTP-authentication (myUsername / myPassword):

  Bottle_1547;1;ODV;http://www.example.com/test_ws/odv/Bottle_1547;http://proxy.example.com:3128;myUsername;myPassword
- No proxy (i.e. the 5th field is empty) and HTTP-authentication (myUsername / myPassword):
  Bottle_1547;1;ODV;http://localhost:8183/test_ws/odv/Bottle_1547;;myUsername;myPassword

Example of a line in coupling.txt for database retrieval (modus 2):

*RNODC_Bottle_14828_10;2;ODV;select * from table1 where … order by metadataColumn1, …, metadataColumnN,pressureColumn;jdbc:mysql://195.178.224.89/dm_test; myUsername;myPassword;odv_mapping_2.xml*

An example of a line in coupling.txt for database retrieval (modus 2) – Case of a MS-SQL Server database:
*RNODC_Bottle_14828_10;2;ODV;select * from table1 where … order by metadataColumn1, …, metadataColumnN,pressureColumn;"jdbc:sqlserver://195.178.224.89;databaseName=dm_test"; myUsername;myPassword;odv_mapping_2.xml*

JDBC connection string value is set between two double quotes (") because of the semicolon (;) char.

4.3.3. Coupling table configuration in a database

When the coupling table is in a database, the table must use the following structure as defined below.Table 7 - Database coupling table structure

Field name *(see detailed description in §4.3.1)*	Type	Can be Null?	Mandatory column	Comment
id	int	No	Yes	Auto increment, primary key
LOCAL_CDI_ID	varchar(64)	No	Yes	Index on this column is recommended for performance
modus	int	No	Yes	Can be one of the following: 1, 2, 3
format	varchar(8)	No	Yes	
filename	varchar(128)	Yes	Yes	
sql_query	varchar(512)	Yes	Yes	The length of this field **can be expanded up to 4000 characters**.
connection_string	varchar(256)	Yes	Yes	
login	varchar(64)	Yes	Yes	
password	varchar(64)	Yes	Yes	
mapping_file	varchar(128)	Yes	Yes	
proxy	varchar(256)	Yes	No	

Length of string types may differ from the values shown in the table, but the field_name in the coupling table must be exactly the same.

Additional fields can be added if needed for internal procedures, but they will be used by the RM.

4.4. Prepare the workflow directories

The RM uses some specific directories for automatic tasks. The paths of these directories have to be set in the RM Configuration file (see reference document [2]).

Directories required for the workflow are listed below. All directories must exist and be accessible by the RM (read/write).

Table 8 - Workflow parameters

Object	Parameter name (in RMConfiguration.properties file)	Description
Ready To Send directory	readyToSendCDIs_path	To store the zips of metadata to be submitted. Needs write access for the data manager.
Temporary directory	tmpDirectory_path	To generate and temporarily store the data
Queue directory	queueDirectory_path	To store the metadata and data when waiting for Import Manager
Archive directory	archive_path	To archive the metadata and data after ingestion. Can become large!
Production directory	production_path	To store the production restricted data. Sensitive data (do not exist elsewhere in SDN architecture, this is the only copy) Can become large!
Embedded database file	embeddedDatabase_path	The file containing the RM embedded database. The file extension shall be *.odb*. This database contains the batches ingestion history and the information on how to retrieve restricted data. Sensitive data
Embedded database backup directory	embeddedDatabase_backupDirectory_path	To store the database backup file on Backup request via the dashboard.

4.5. Example of configuration

Figure 6 shows an example of organisation.

Yellow directories concern data (already existing in the Download Manager)

Blue directories concern workflow (new in RM). Those have to be created empty.

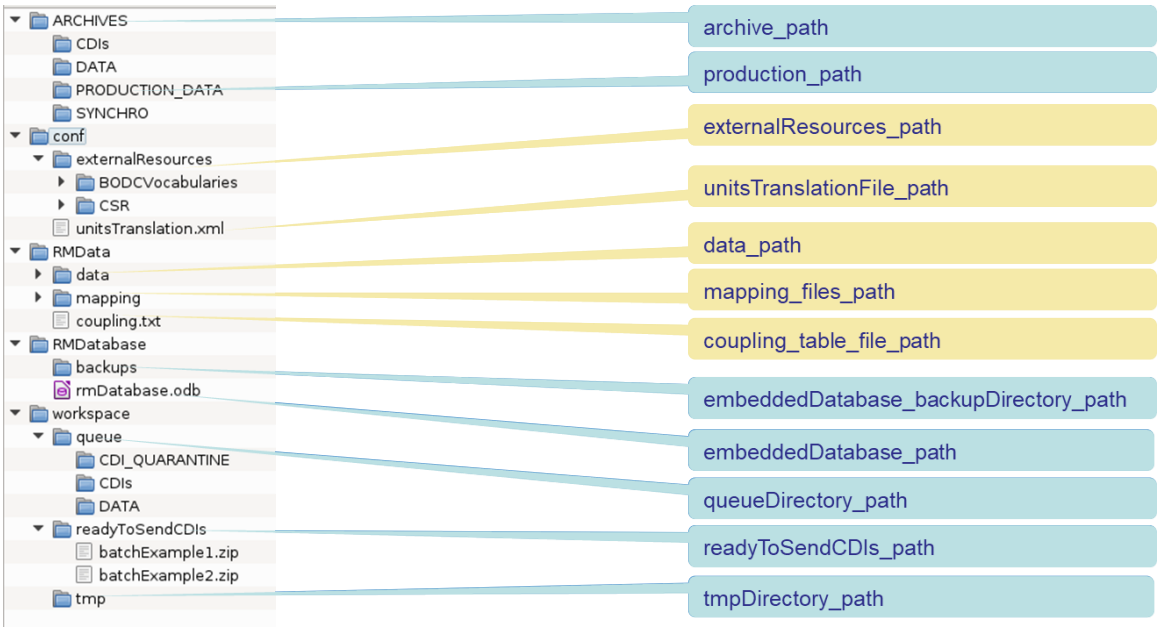


Figure 6 - Data and workflow directories organisation example

5. Replication Manager functionalities

5.1. Administration tools

The RM offers administration tools via the web interface. These tools concern technical tasks (check RM configuration) as well as data management tasks (metadata and data submission).

Web interface presentation

The dashboard is composed of 6 main pages/thumbnails:

- Summary
  - information about configuration, component versions (Octopus, Tomcat,…)
  - access to the application logs
  - checks, comparisons and populate functions
- Batches in progress
  - selectable list of “Ready to send” batches
  - current processing batch
  - list of queued batches
- Batches
- LOCAL_CDI_IDs
  - Local catalogue: list of LOCAL_CDI_IDs in production, with versions and formats
- About
  - Information about the RM, change log.

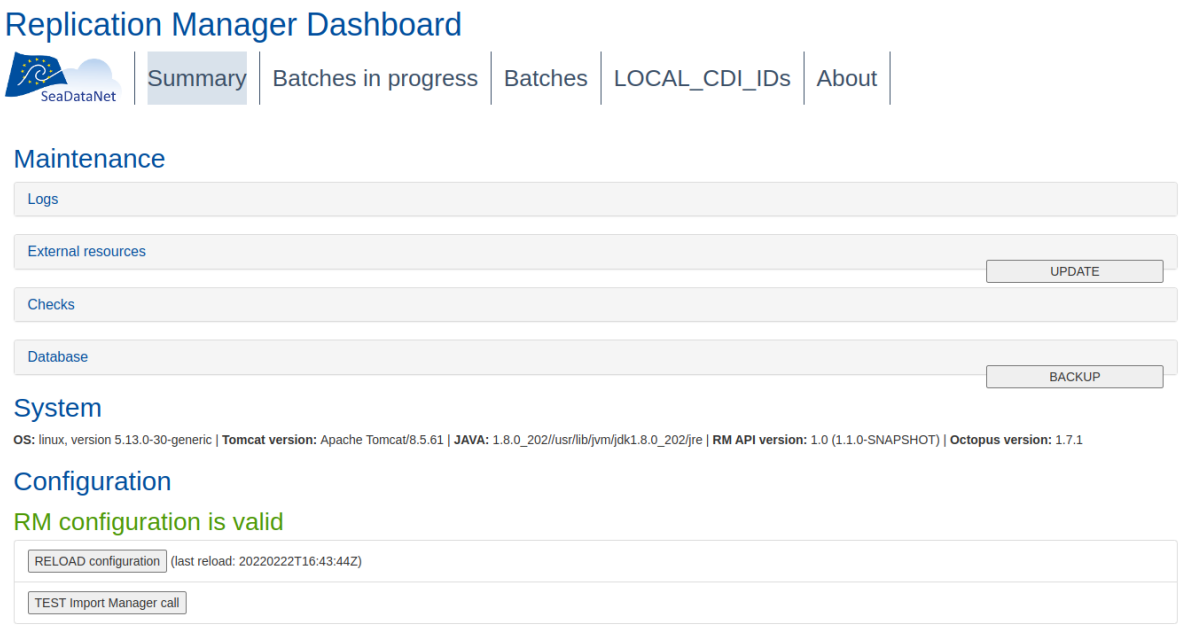


Figure 7 Replication Dashboard

5.2. External resources

This section gives information about external resources that are loaded in the RM:

- BODC vocabularies: version
- CSR list: version and date

The UPDATE button updates the external resources.

TROUBLESHOOTING: 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”.

5.3. Checks and synchronization

Tools for checks and synchronisation are accessible from the Summary page of the RM dashboard.

Three actions can be launched:

- Local check : check coupling table lines and mapping files (if exist)
- Local versus Central check: Launch comparisons between
    - coupling and central full catalogue (check if all coupling table entries are present in the catalogue)
    - local embedded database and central restricted catalogue (check if all catalogue entries are present in the local embedded database, and if data files exist)
- Population: populates the system Population (only for the first installation – hidden by default)

The population shall be done only once after first RM installation. See details in [2] Replication Manager Installation Manual.

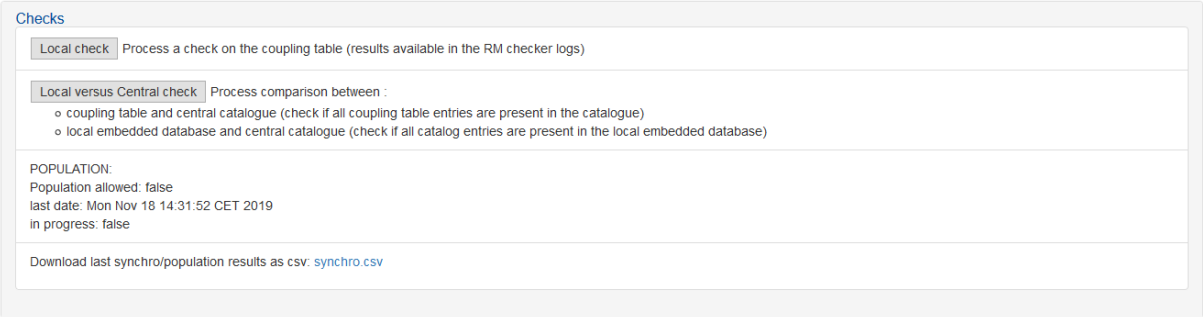


Figure 8 - Checks section of Summary page in RM dashboard

5.3.1. Local check

The check function helps Data Centres to perform several checks on:

- the coupling table
- the XML mapping files declared in this coupling table (in case of entries using modus 2)
- the availability of data

First, a test is performed on the coupling table. Then, checks on XML mapping files are done only for rows that have passed the coupling table test.

Checks on coupling table and data availaibility:

- each couple [LOCAL_CDI_ID, Format] must be unique
- for modus 3 entries, format shall be SeaDataNet (MEDATLAS, ODV, CFPOINT)
- for modus 1 or 3 entries, data file must exist and be readable (web services are not checked)
- For modus 2 entries, XML mapping files must exist and be readable

Checks on the XML mapping files defined in coupling table:

- each file must contain only LATIN-1 characters
- all mandatory metadata mapping ODV column must be declared
- for each data mapping declaration, check if OBJECT, UNITS and SUBJECT starts with an allowed characters sequence (SDN:P01::, …)
- use of NVS2 vocabularies
- detection of deprecated P01 and P06 terms

Results can be read in the RM Checker log available from the RM dashboard Summary page.

5.3.2. Central catalogue/local consistency check

This function checks the consistency between the coupling table, the RM embedded database and the Central Catalogue. The checks are listed in Table 9 and Table 10.

Table 9 - Comparison between coupling table and CDI full catalogue

Check	Result type
Each LOCAL_CDI_ID /format couple in the coupling table must exist in the full catalogue	WARNING

Table 10 - Comparison between RM embedded database and CDI catalogue

Check	Result type
Each LOCAL_CDI_ID/format couple in the restricted catalogue table must exist in embedded database	WARNING if unrestricted ERROR if restricted
Same LOCAL_CDI_ID/format couple must have the same access restriction values	ERROR
Same LOCAL_CDI_ID/format couple must have the same PID values	ERROR
Each restricted couple LOCAL_CDI_ID/format associated file must exist in PRODUCTION directory	ERROR

Results can be read in the RMLogChecker or downloaded as a csv file from the RM Summary page.

The csv file columns are detailed in the table below:

Table 11 - CSV file with check results

Column name	Description
Error level	ERROR or WARNING
Synchronization stage	COMPARE_COUPLING, COMPARE_INTERNAL_DB
message	Detailed message
localCdiId	Local_CDI_ID (read from CDI metadata catalogue or coupling table)
format	format
version	Version of the file (if exist)

5.4. Ingestion: submit new metadata and data

The process is the same for new CDIs and updates. The full process is composed of 4 stages, which will be described in this chapter.

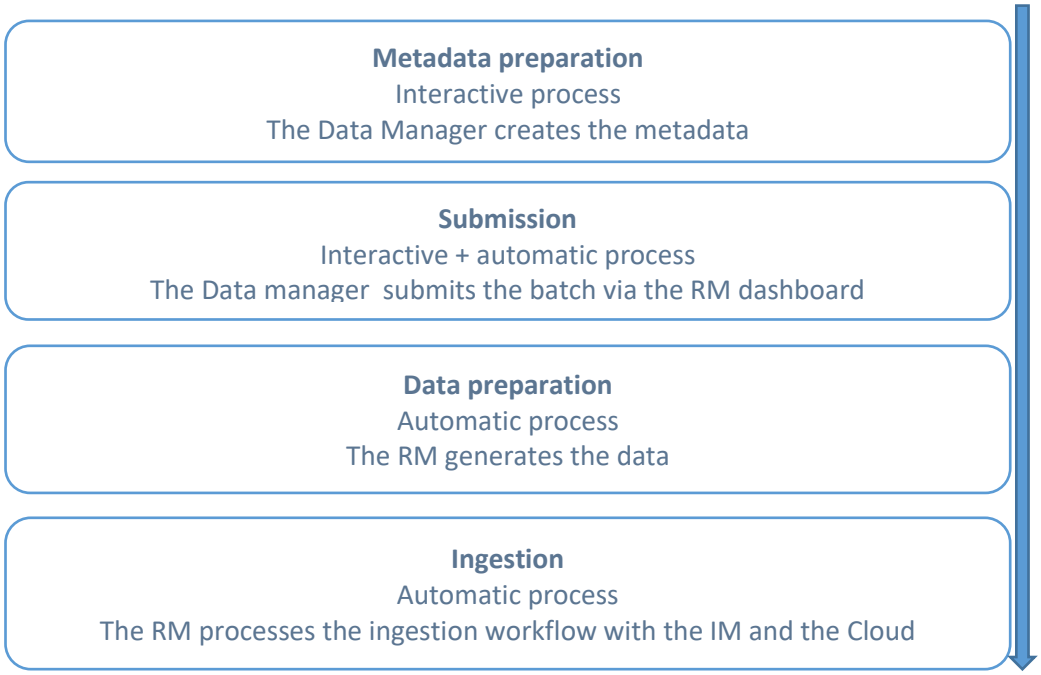


Figure 9 - Batches submission stages

The data are generated immediately when the metadata are submitted

- Data (unrestricted and restricted) and coupling must be ready (cf. §4) when you submit metadata!

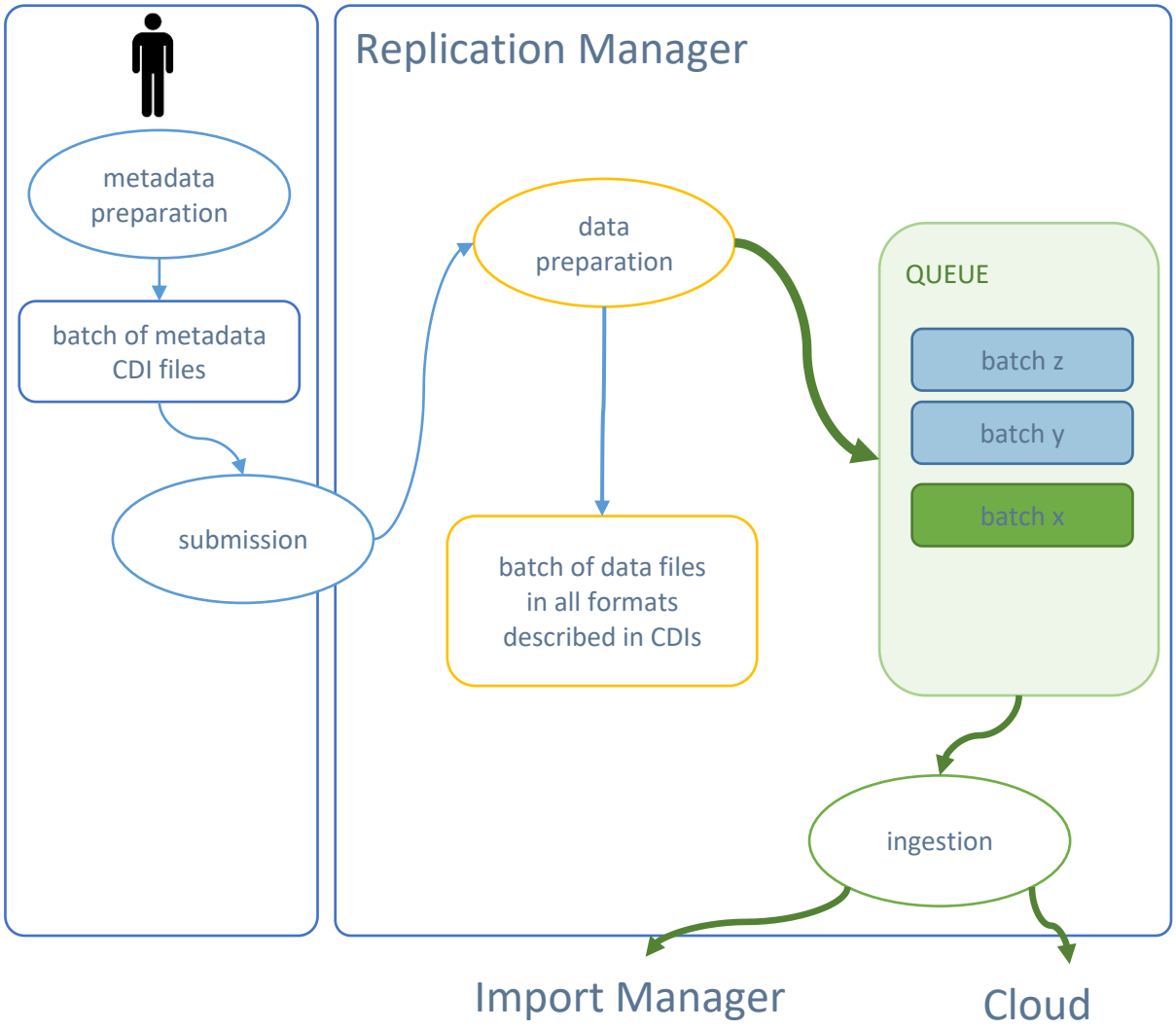


Figure 10 - Metadata and data submission (assuming data are ready)

5.4.1. Metadata preparation

Metadata preparation steps are all realised by the Data Manager:

- He/she creates CDI metadata xml files.
- He/she zips the CDI metadata files into one or multiple zip files (1 zip = 1 batch)
- He/she puts the zips files into the “Ready To Send” directory (see Table 8 - Workflow parameters)

Requirements and limitations:

- Files names: use only Uppercase [A-Z] and lowercase [a-z] English alphabet characters, Digits 0-9, dot (.) , hyphen (-) or underscore (_)
- Zip file max length is 100 Mb
- Maximum of CDI files allowed in a zip: 20 000

➔ The zip files are displayed in the “Batches in Progress” page of the RM dashboard

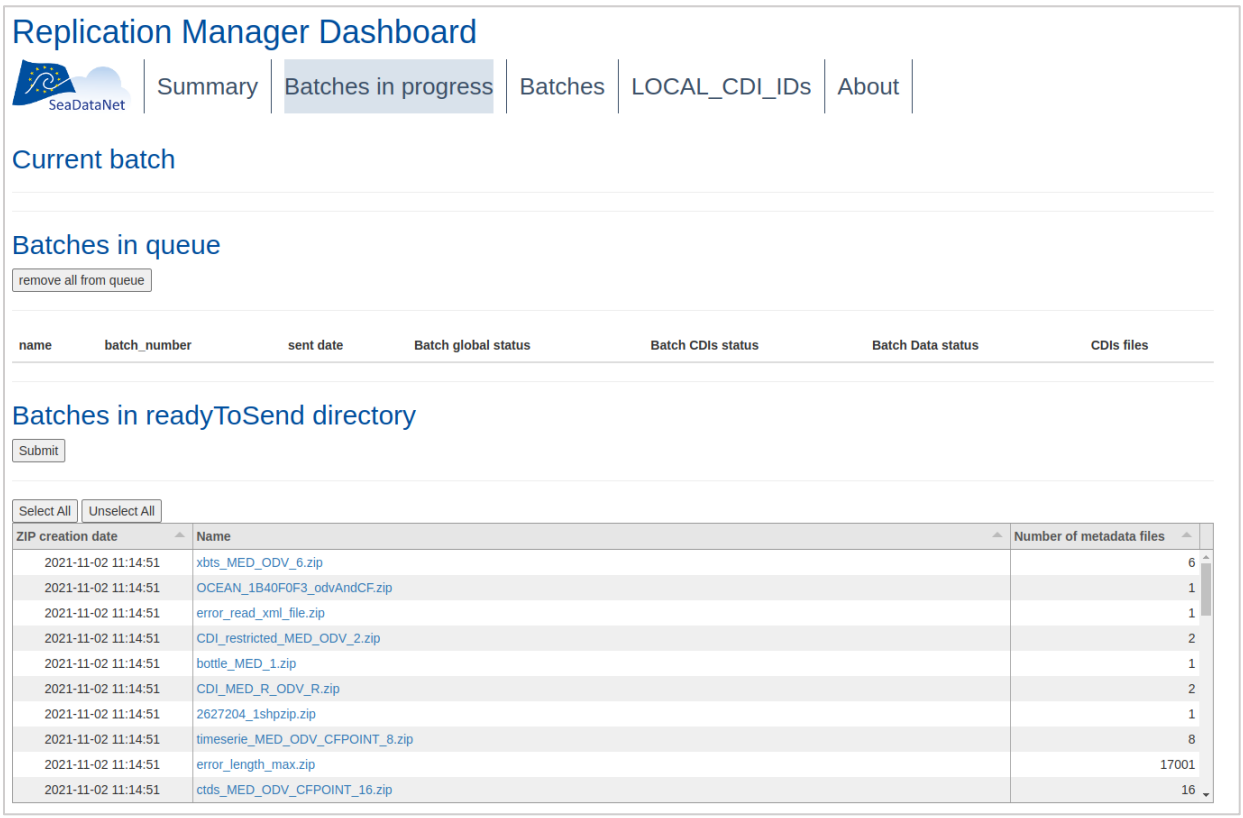


Figure 11 - "Batches in Progress" page of the RM dashboard: batches ready to send

5.4.2. Submission

The submission stage is initiated by the Data Manager.

Note: the data must be prepared before submission, see §4.

The RM processes automatically several checks in order to accept or reject the submission .

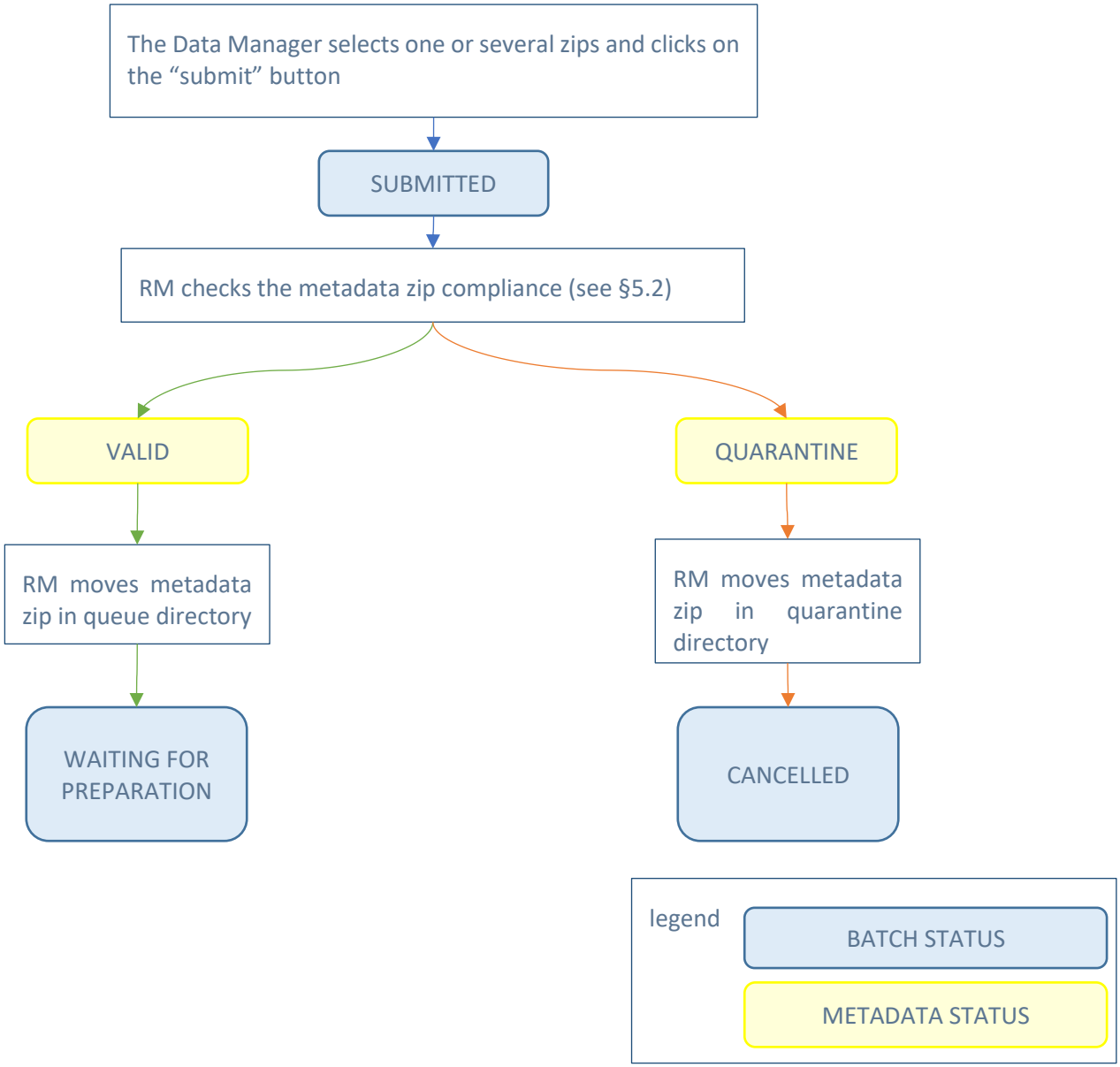


Figure 12 - Submission stage

The submitted batches appear in the “Batches in queue” section, in the “Batches in Progress” page of the RM dashboard.

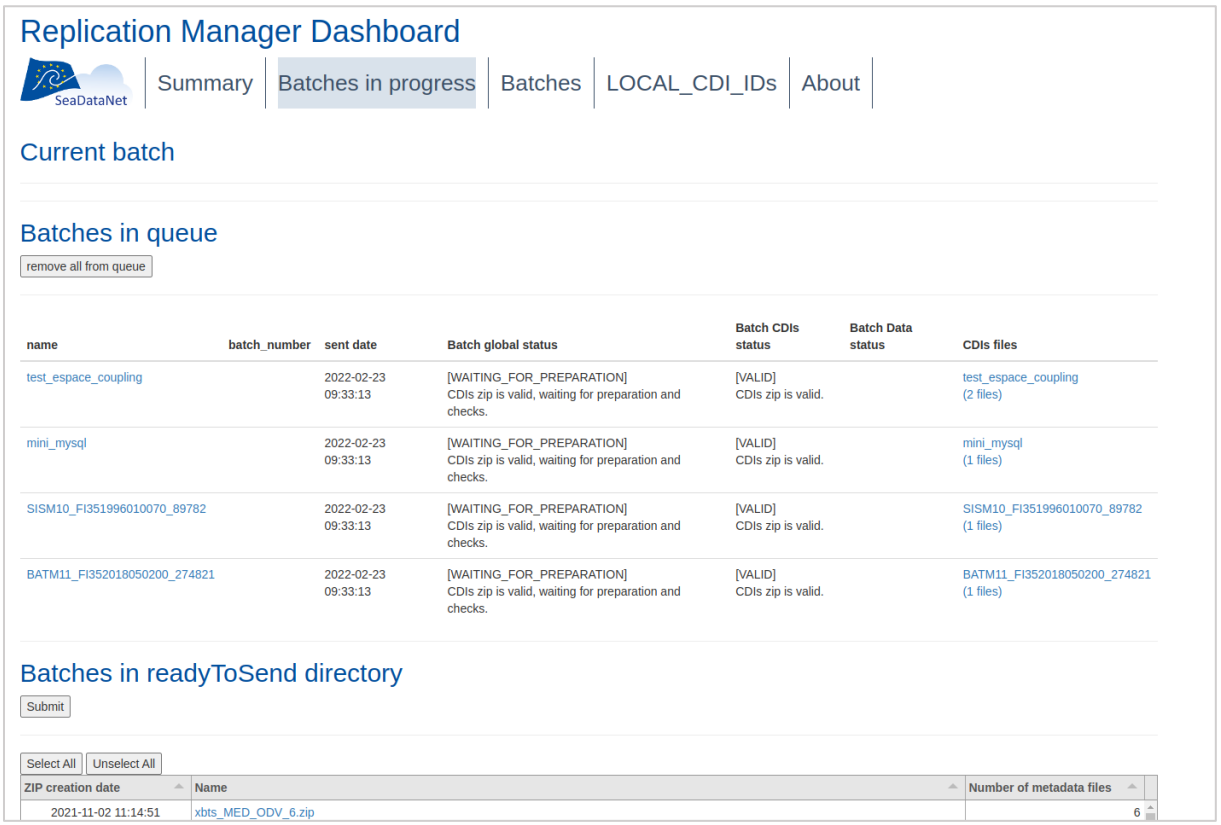


Figure 13 - "Batches in Progress" page of the RM dashboard: batches waiting in queue

5.4.3. Data preparation

This stage is totally automatic.

In Modus 2, data is generated from a database, and deleted in case of an error. In order to check the origin of an error or a data invalidity, the RM automatically copies invalid Modus 2 data files in an ARCHIVES/MODUS2 directory. This directory is automatically cleaned from files older than 1 month.

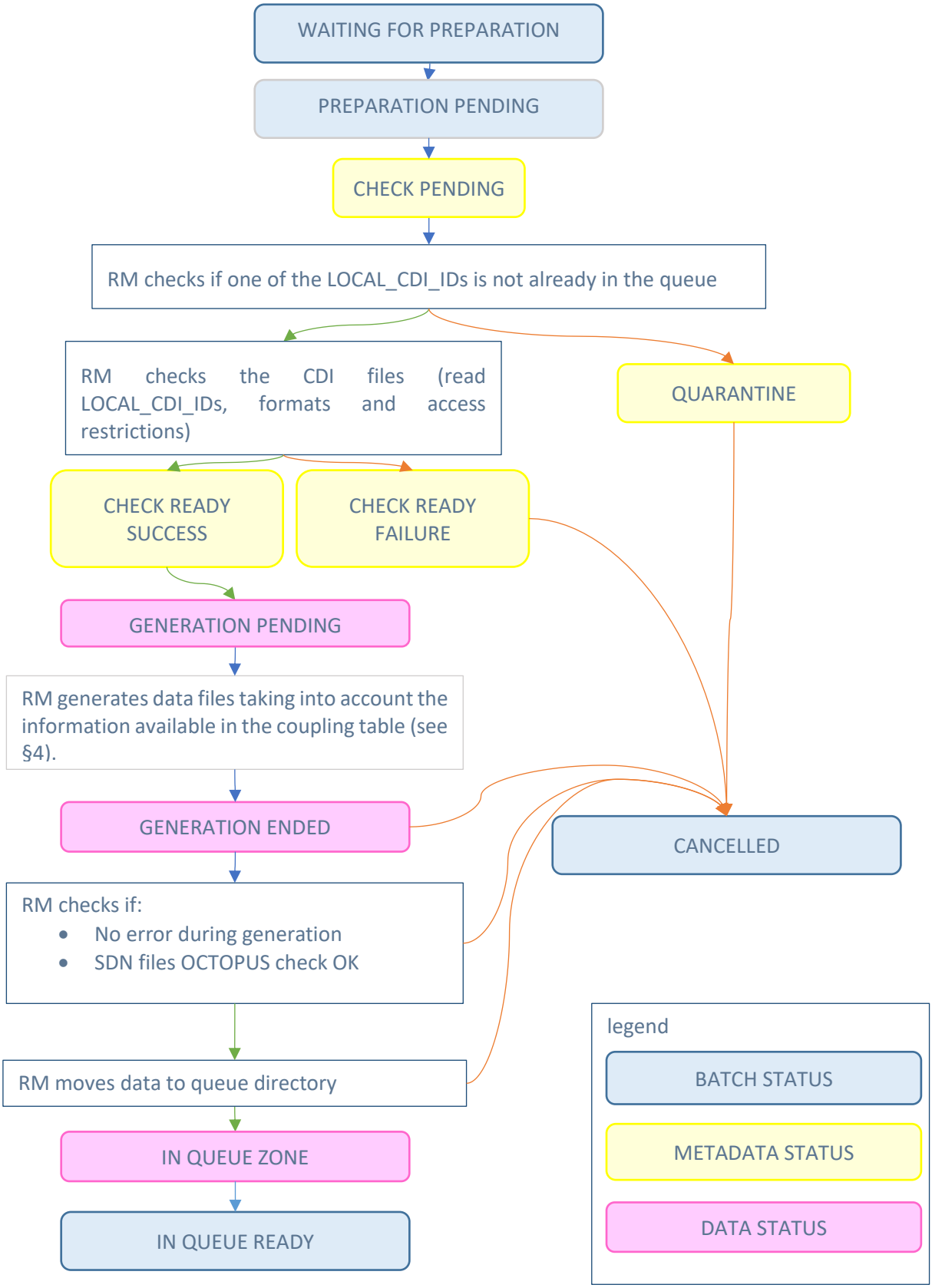


Figure 14 - Data preparation stage for each batch

5.4.4. Ingestion

The ingestion stage is an automatic workflow with IM and Cloud:interactivity is required only for validation, at certain times of the workflow. The Data Manager will be alerted by IM, and a message with a link will be displayed in the RM Dashboard.

The IM processes one batch at a time. At the end of the process, unrestricted datafiles are uploaded in the cloud and archived locally, restricted data files are stored in the local production directory. Both restricted and unrestricted files are versioned. Unrestricted files have a unique cloud identifier (PID). When a batch has ended (successfully or not), the IM informs the RM, which will start a new batch ingestion.

At each step, the batch can be cancelled in case of errors.

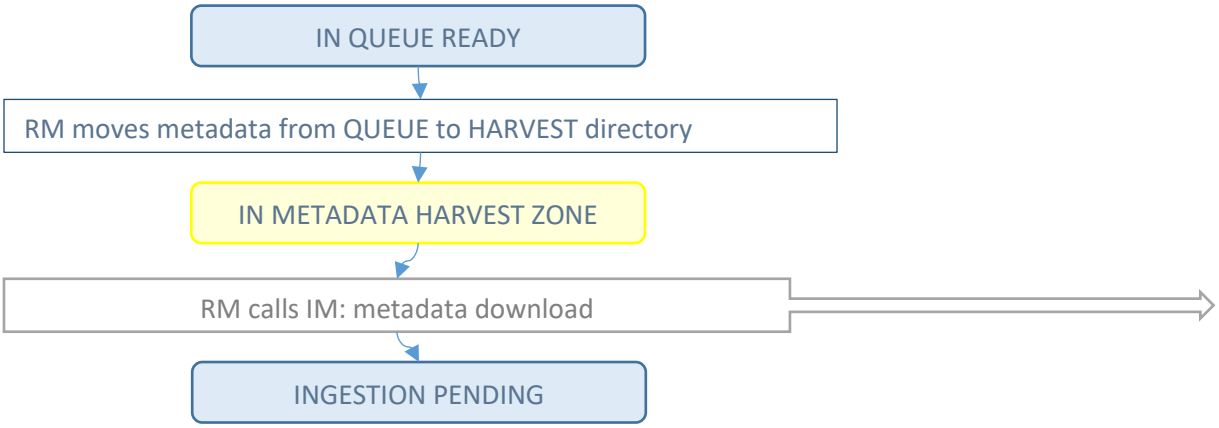


Figure 15 - Ingestion stage, step 1: call the IM for metadata harvesting

The batch zip file is in the metadata harvest zone, ready to be harvested by the IM.

The harvest URL is

http://RMhost:RMPort/ReplicationManager/HARVEST_METADATA/<batchName>.zip

where <batchName>.zip is the name of the zip file as it was created in the Ready To Send directory

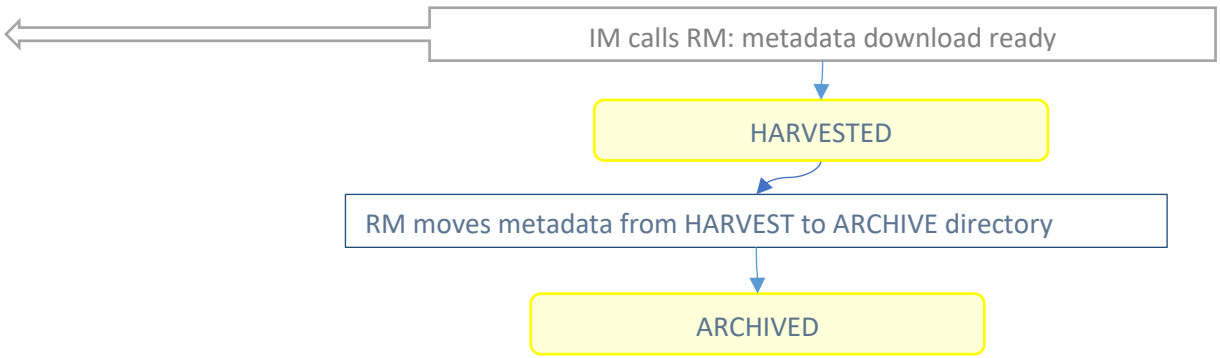


Figure 16 - Ingestion stage, step 2: metadata archiving

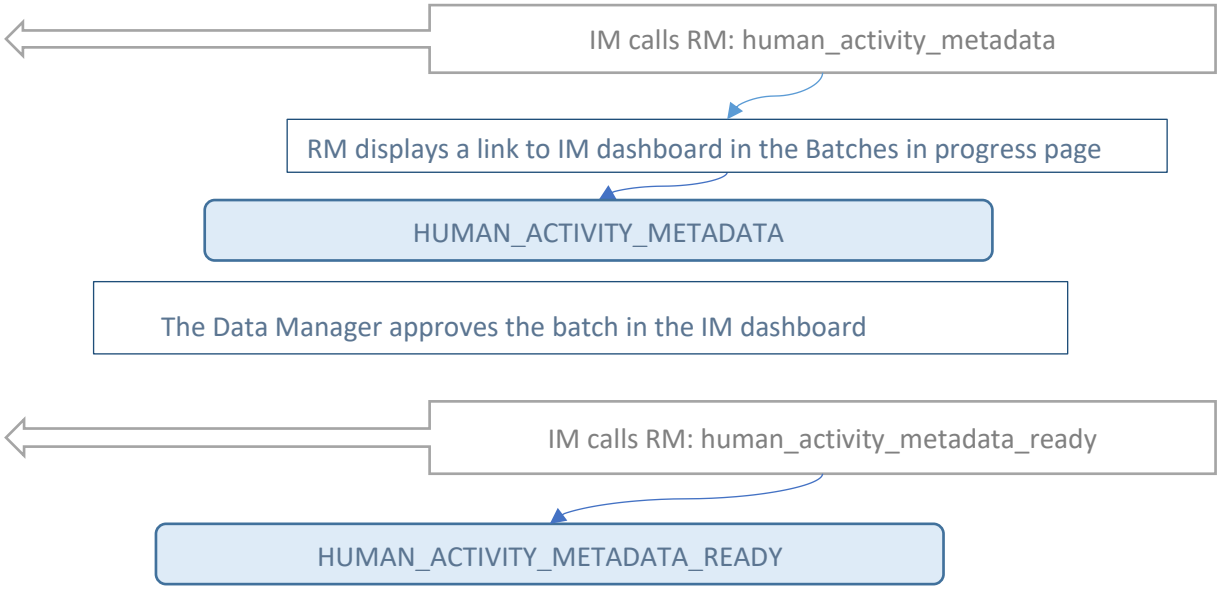


Figure 17 - Ingestion stage, step 3: metadata approbation

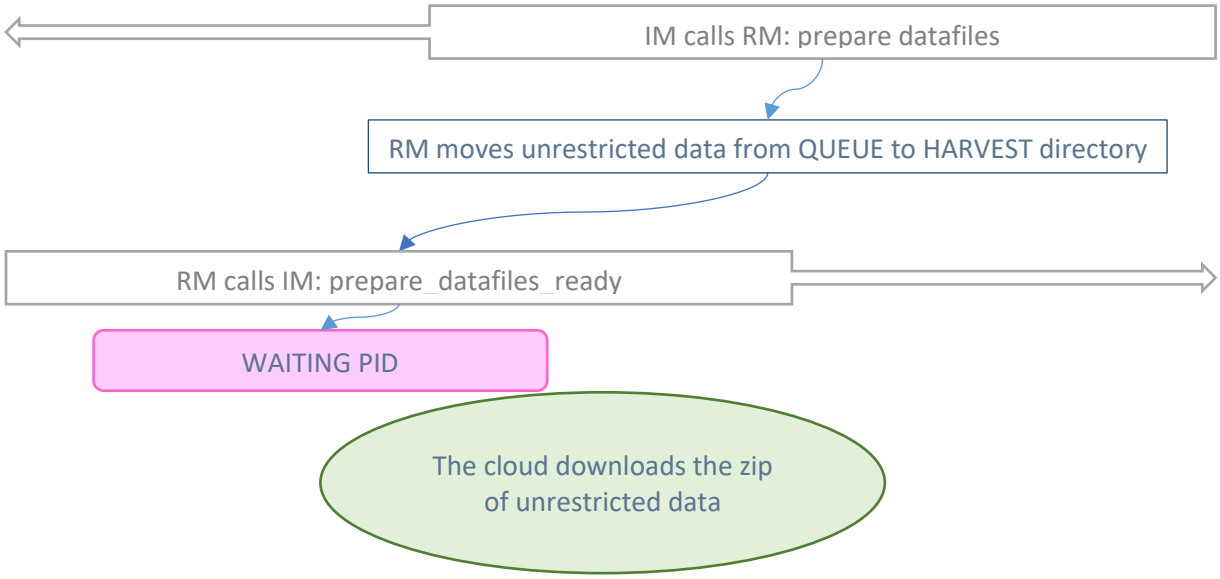


Figure 18 - Ingestion stage, step 4: data preparation

The data zip file is in the data harvest zone, ready to be harvested by the the EUDAT cloud.

The harvest URL is

http://RMhost:RMPort/ReplicationManager/HARVEST_DATA/<batchNumber>_<EDMO_CODE>.zip

where < batchNumber > is the identifier set to the batch by the IM. It is available in the batch detailed page, in the RM dashboard.

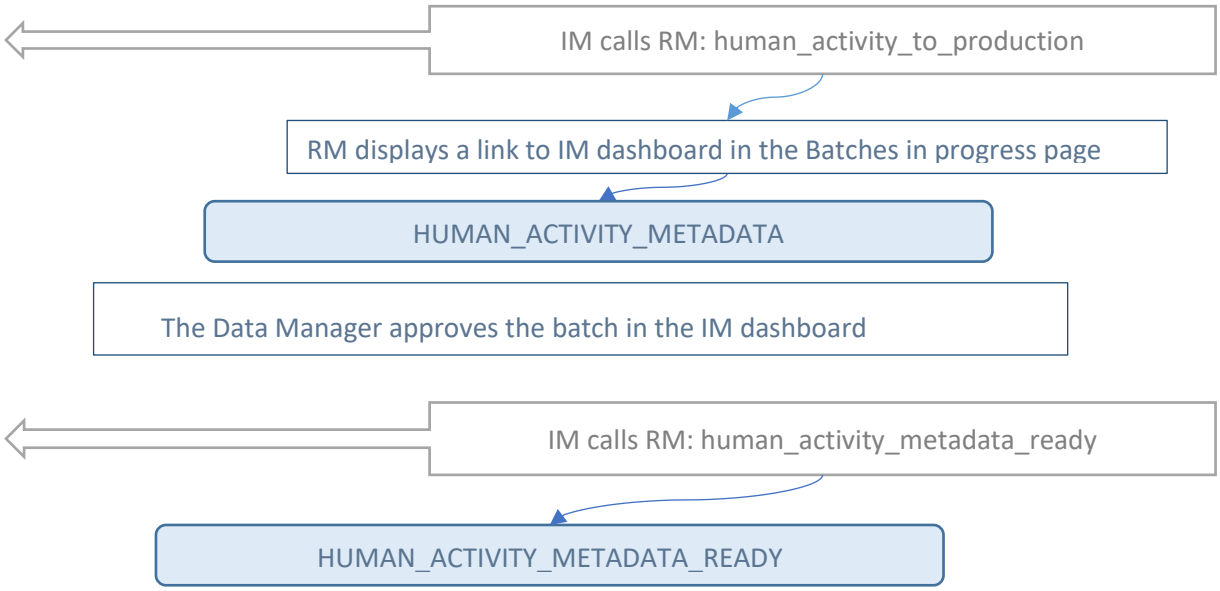


Figure 19 - Ingestion stage, step 5: data approbation

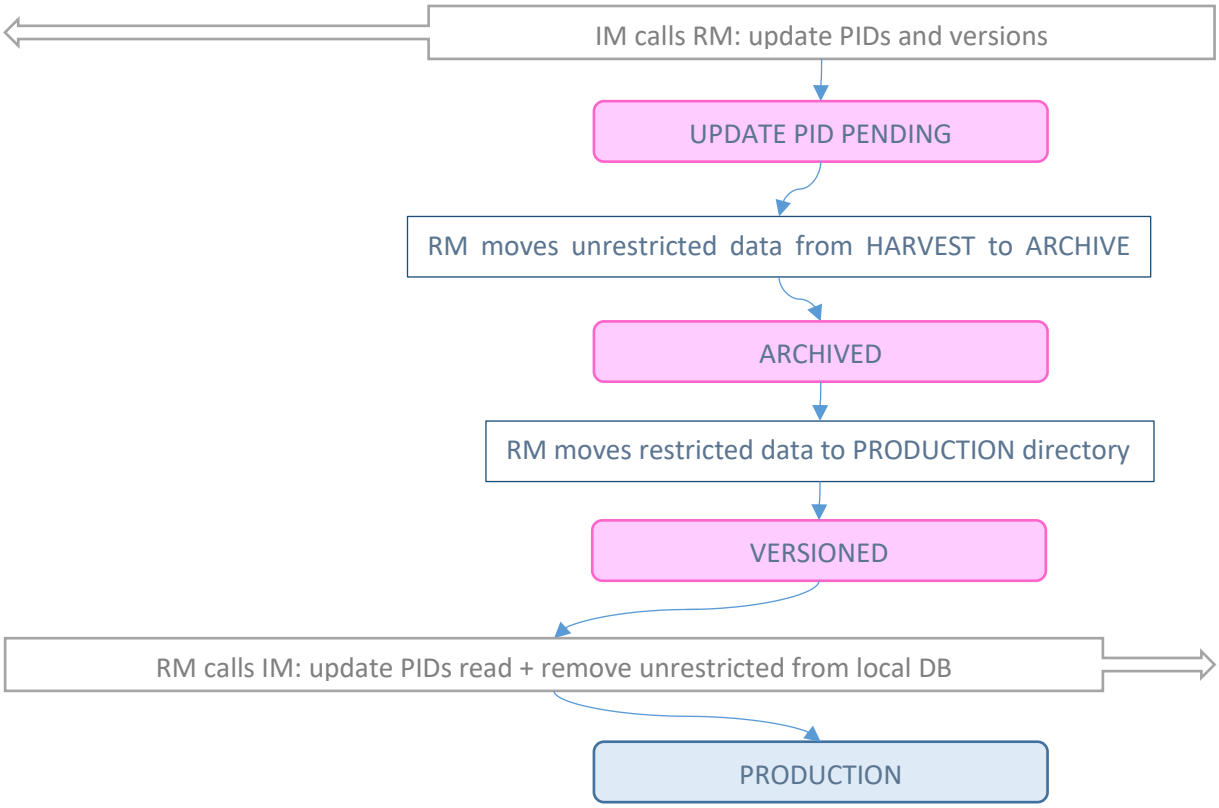


Figure 20 - Ingestion stage, step 6: archiving and versioning, production start-up

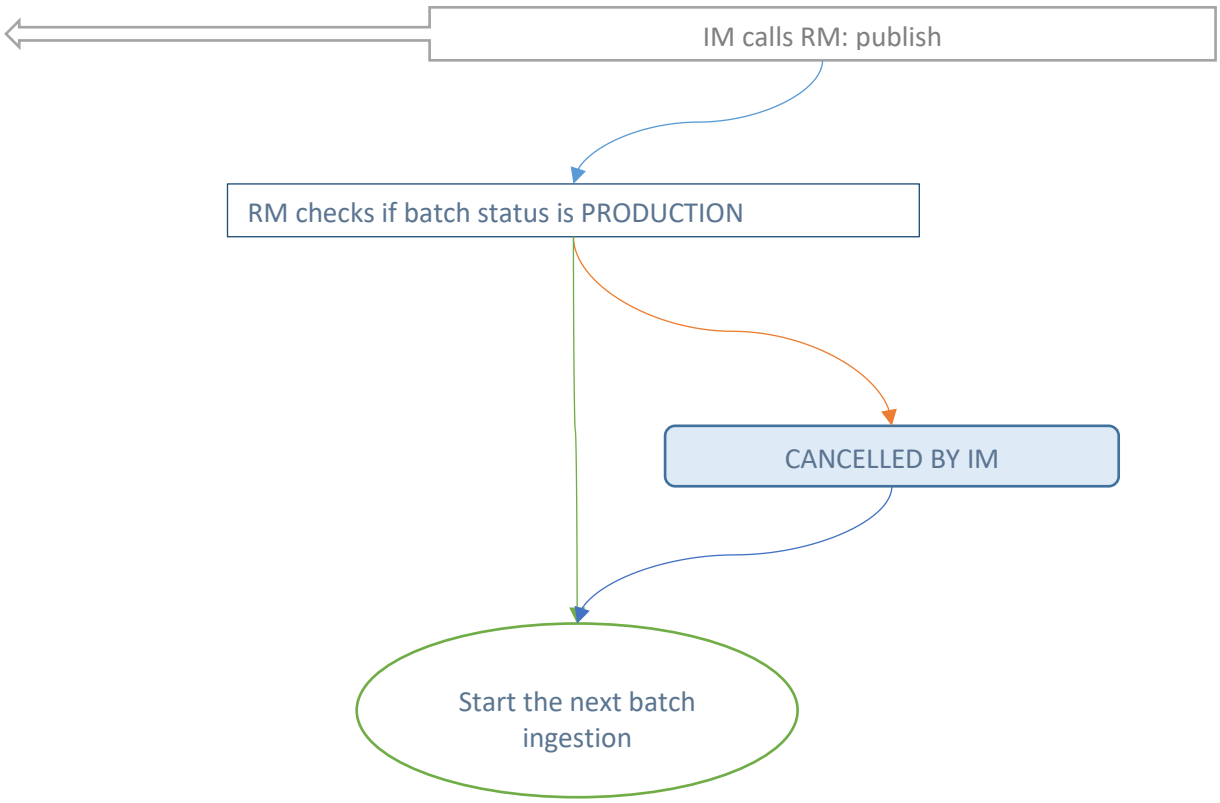


Figure 21 - Ingestion stage, step 7: batch ending - start next one

5.5. Users data requests

After the ingestion process (§5.4), data are stored in the EUDAT cloud (unrestricted) or in the Data Centre (restricted in PRODUCTION directory) (see Figure 1 - Metadata and data storage in SDN infrastructure).

When a user requests restricted data via the SDN portal, an automatic workflow involving the RM, the IM and the cloud starts. No human intervention is required.

Note: user requests on unrestricted data no longer involve the RM as the data are stored on the cloud.

You will find below a detailed presentation of the restricted order workflow:

When a user requests restricted data on the SDN portal, the Import Manager notifies the RM. The RM creates a zip with requested restricted data that will be harvested by the cloud.

The restricted data are stored in a specific zone of the cloud, available only for the user, until he/she downloads the files.

The RM creates one directory for each order, named: < orderNumber>_<Marine-ID loginCode>

Once the files are harvested by the cloud, the RM is notified and deletes them.

Once the user has downloaded the files, they are deleted from the cloud.

The workflow is:

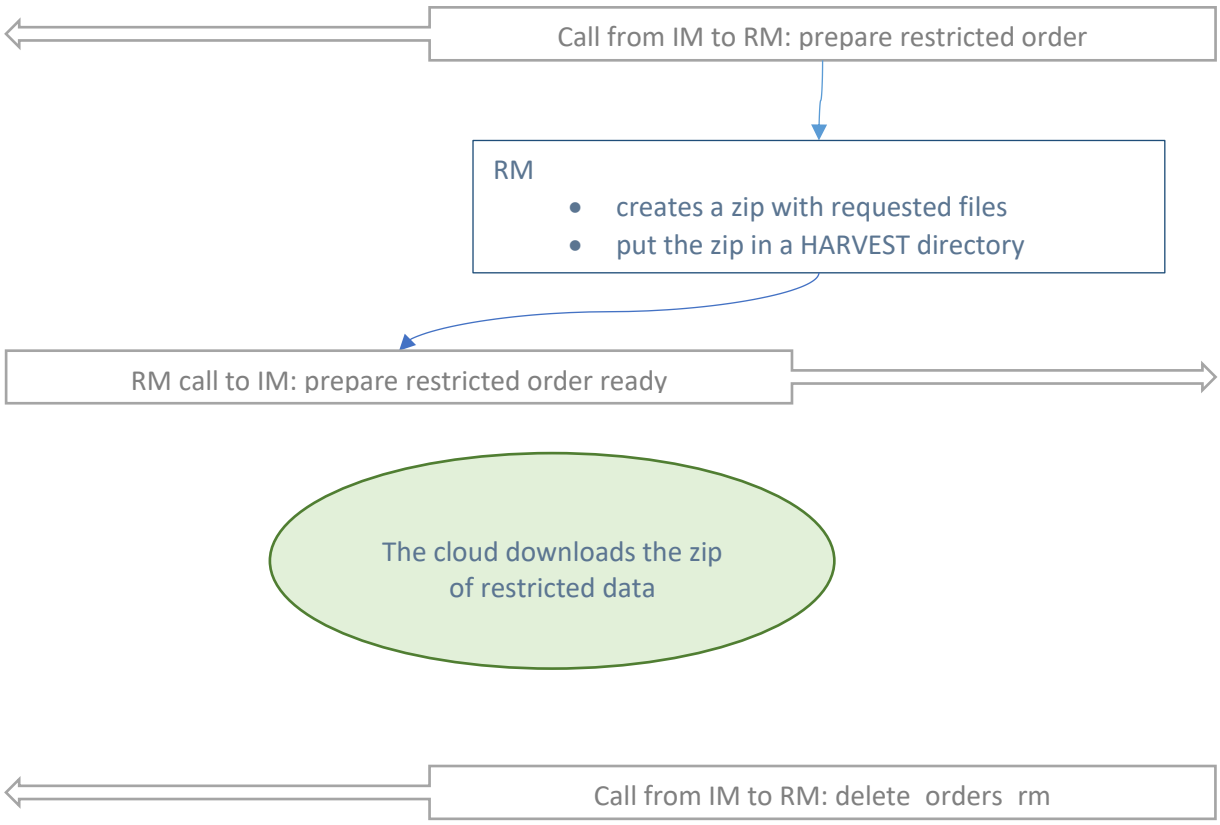


Figure 22 - Workflow for the restricted data

The data zip file is in the orders harvest zone, ready to be harvested by the the EUDAT cloud.

The harvest URL is

http://RMhost:RMPort/ReplicationManager/HARVEST_ORDERS/<filename>.zip

where < filename > is the filename requested by the IM during the prepare_restricted_order call.

Appendix – example of ODV XML mapping file

```
<?xml version="1.0" encoding="UTF-8"?>
<!--
================================================
This is a mapping file example, to create ODV files with DM modus 2.
Associated SQL query is:
        SELECT PLATFORM, STATION, STATION_DATE, LONGITUDE, LATITUDE,
        BOTTOM_DEPTH,
        TYPE_COLUMN_NAME,
        LOCAL_CDI_ID,
        DEPTH, QV_DEPTH, TEMP, QV_TEMP, SALINITY, QV_SALINITY, OXYGEN, QV_OXYGEN, LITHO, QV_LITHO
        DEPTH_INSTRUMENT, DEPTH_FALL_RATE, TEMP_INSTRUMENT, TEMP_FALL_RATE,
        SDNREF_COL
        FROM dm_test.modus2_prof_data
        WHERE LOCAL_CDI_ID="myExample" ORDER BY BOTTOM_DEPTH;
================================================
 -->
<root>
 <updated>2009-09-22T10:30:00</updated>
 <codes type="odv">

  <!—
  ============================================================================
  Mandatory Metadata columns that can not be retrieved from batch
  ============================================================================
  attributes:
  from          : [MANDATORY] leave empty
  to                    : [MANDATORY] ODV column header
  local         : [MANDATORY] SQL column name (or alias in query) containing metadata

  Date pattern:
  use an ISO8601 patter appropriate for the wanted output ODV file precision
  Available patterns are:
        YEAR ("YYYY"),
                YEAR_MONTH ("YYYY-MM"),
                YEAR_MONTH_DAY ("YYYY-MM-DD"),
                YEAR_MONTH_DAY_HOUR ("YYYY-MM-DDThh"),
                YEAR_MONTH_DAY_HOUR_MIN ("YYYY-MM-DDThh:mm"),
                YEAR_MONTH_DAY_HOUR_MIN_SEC ("YYYY-MM-DDThh:mm:ss"),
                YEAR_MONTH_DAY_HOUR_MIN_SEC_FRACTION1("YYYY-MM-DDThh:mm:ss.s"),
                YEAR_MONTH_DAY_HOUR_MIN_SEC_FRACTION2("YYYY-MM-DDThh:mm:ss.ss")         ,
                YEAR_MONTH_DAY_HOUR_MIN_SEC_FRACTION3("YYYY-MM-DDThh:mm:ss.sss");
        -->
  <code from="" to="SDN:LOCAL:Cruise" local="PLATFORM" />
  <code from="" to="SDN:LOCAL:Station" local="STATION" />
  <code from="" to="SDN:LOCAL:yyyy-mm-ddThh:mm:ss.sss" local="STATION_DATE" />
  <code from="" to="SDN:LOCAL:Longitude, degrees_east" local="LONGITUDE" />
  <code from="" to="SDN:LOCAL:Latitude, degrees_north" local="LATITUDE" />
```

```
<!--
=============================================================================
Optional Metadata columns
=============================================================================
attributes:
from            : [MANDATORY] leave empty
to                      : [MANDATORY] ODV column header
local           : [MANDATORY] SQL column name (or alias in query) containing metadata
-->
<!-- Bot. Depth [m]  (If not given, value is set to 0)  -->
<code from="" to="SDN:LOCAL:Bot. Depth, m" local="BOTTOM_DEPTH" />
<!--Type (If not given, value is set to *)-->
<code from="" to="SDN:LOCAL:Type " local="TYPE_COLUMN_NAME" />


        <!--
        =============================================================================
        Optional Metadata SDN references
        =============================================================================
        use this element to create sdn_reference lines

        attributes:
        from            : [MANDATORY] leave empty
        to                      : [MANDATORY] "sdn_reference"
        local           : [MANDATORY] SQL column name (or alias in query) of column containing all sdn_reference for this file as a string
                                                references must be stored in a unique String without separator.
                                                        Example:
                                                        <sdn_reference xlink:href="http://seadatanet.maris2.nl/v_cdi_v3/print_xml.asp?edmo=486&identifier=FI35200345015_03AC0_H09" xlink:role="isDescribedBy" xlink:type="SDN:L23::CDI"/><sdn_reference xlink:href="http://www.ifremer.fr/CSR" xlink:role="isObservedBy" xlink:type="SDN:L23::CSR"/><sdn_reference xlink:href="http://exampleHttp"/><sdn_reference xlink:href="doi:exampleDoi"/>
        -->
<code from="" to="sdn_reference" local="SDNREF_COL" />

<!--
=============================================================================
Data columns
=============================================================================
attributes:
from            :       [MANDATORY] "SDN:P01::xxxxxxxx, SDN:P06::yyyy"
                                                P01 and P06 BODC vocabularies codes associated to the parameter
                                                where xxxxxxxx is P01 code and yyyy is P06 code
to                      :       [MANDATORY] "SDN:LOCAL:xxxx, yyyy" or  "SDN:LOCAL:xxxx"
                                                ODV column header where xxxx is local parameter name and yyyy local unit name
                                                unit can be omitted if type=INDEXED_TEXT (see below)
local           :       [MANDATORY] SQL column name (or alias in query) containing data
qflag           :       [MANDATORY] SQL column name (or alias in query) containing data QC values
instrument      :       [OPTIONNAL] SQL column name containing BODC L22 instrument code
fall_rate       :       [OPTIONNAL] SQL column name containing BODC L33 fall rate code
type            :       [OPTIONNAL] set to "INDEXED_TEXT" if data are strings
```

```
                                                    unit in "to" attribute will be ignored
                                                    you shall use UUUU as P06 code in "from" attribute for
this kind of data
  ========================================================================== -->


   <code from="SDN:P01::ADEPZZ01, SDN:P06::ULAA" to="SDN:LOCAL:DEPH, meters" local="DEPH"
qflag="QV_DEPH" instrument="DEPH_INSTRUM" fall_rate="DEPH_FALLRATE" />
   <code from="SDN:P01::TEMPPR01, SDN:P06::UPAA" to="SDN:LOCAL:TEMP, Celsius degree" local="TEMP"
qflag="QV_TEMP" />
   <code from="SDN:P01::PSLTZZ01, SDN:P06::UUUU" to="SDN:LOCAL:PSAL, P.S.U." local="PSAL" qflag="QV_PSAL"
/>
   <code from="SDN:P01::SIMPLITH, SDN:P06::UUUU" to="SDN:LOCAL:LITHO" local="LITHO" qflag="QV_LITHO"
type="INDEXED_TEXT"/>
  </codes>
</root>
```