Replication Manager User Manual

version 1.0.43
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.
The current document can be found on SeaDataNet web site: 
https://www.seadatanet.org/Software/Replication-Manager

References


Table of contents

1. Introduction ................................................................................................................................. 6
2. Pre-requisites and warnings ........................................................................................................ 6
3. Replication Manager overview .................................................................................................. 7
   3.1. Ingestion ................................................................................................................................. 7
   3.2. User data requests .................................................................................................................. 8
   3.3. Administration tools ............................................................................................................. 8
4. How to prepare data and workflow directories ........................................................................... 8
   4.1. Data in modus 1 or 3 ............................................................................................................. 9
   4.2. Data in modus 2 .................................................................................................................. 11
      4.2.1. Purpose of the ODV XML mapping file ........................................................................... 13
      4.2.2. SQL query ...................................................................................................................... 13
      4.2.3. XML notation description ............................................................................................... 13
      4.2.4. ODV metadata columns ................................................................................................. 15
      4.2.5. ODV data columns ......................................................................................................... 16
   4.3. Creation of the Coupling Table ............................................................................................ 17
      4.3.1. Coupling table fields ...................................................................................................... 17
      4.3.2. Coupling table in a configuration file ............................................................................ 19
      4.3.3. Coupling table configuration in a database .................................................................. 20
   4.4. Prepare the workflow directories .......................................................................................... 21
   4.5. Example of configuration ..................................................................................................... 21
5. Replication Manager functionalities ............................................................................................ 22
   5.1. Administration tools ............................................................................................................ 22
   5.2. Checks and synchronization .................................................................................................. 23
      5.2.1. Local check .................................................................................................................... 24
      5.2.2. Synchronise Central catalogue/coupling consistency check ....................................... 24
   5.3. Ingestion: submit new metadata and data ............................................................................. 26
      5.3.1. Metadata preparation ..................................................................................................... 27
      5.3.2. Submission ..................................................................................................................... 28
      5.3.3. Data preparation ............................................................................................................ 30
      5.3.4. Ingestion ....................................................................................................................... 32
   5.4. Users data requests ............................................................................................................... 35
When the coupling table is in a database, the table must use the following structure as defined below. Table 7 - Database coupling table structure .......................................................... 20

Table 8 - Workflow parameters .................................................................................................................. 21

Table 9 - Comparison between coupling table and CDI full catalogue .................................................. 24

Table 10 - Comparison between RM embedded database and CDI catalogue ........................................ 24

Table 11 - CSV file with check results ......................................................................................................... 26

Figure 1 - Metadata and data storage in SDN infrastructure .................................................................. 7

Figure 2 - Data processing in modus 1 ........................................................................................................... 10

Figure 3 - Data processing in modus 3 .......................................................................................................... 11

Figure 4 - Modus 2 process .......................................................................................................................... 12

Figure 5 - Data processing in modus 2 .......................................................................................................... 12

Figure 6 - Data and workflow directories organisation example .............................................................. 22

Figure 7 Replication Dashboard .................................................................................................................... 23

Figure 8 - Synchronisation section of Summary page in RM dashboard ................................................. 24

Figure 9 - Batches submission stages ......................................................................................................... 26

Figure 10 - Metadata and data submission (assuming data are ready) ....................................................... 27

Figure 11 - “Batches in Progress” page of the RM dashboard: batches ready to send ................................ 28

Figure 12 - Submission stage ....................................................................................................................... 29

Figure 13 - “Batches in Progress” page of the RM dashboard: batches waiting in queue ....................... 30

Figure 14 - Data preparation stage for each batch ....................................................................................... 31

Figure 15 - Ingestion stage, step 1: call the IM for metadata harvesting .................................................... 32

Figure 16 - Ingestion stage, step 2: metadata archiving ............................................................................ 32

Figure 17 - Ingestion stage, step 3: data preparation .................................................................................. 33

Figure 18 - Ingestion stage, step 4: archiving and versioning, production start-up .................................. 34

Figure 19 - Ingestion stage, step 5: batch ending - start next one .............................................................. 35

Figure 20 - Workflow for the restricted data ............................................................................................... 36
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 (i.e. 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.
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>
<thead>
<tr>
<th>Parameter name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>data_path</td>
<td>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.</td>
</tr>
<tr>
<td>mapping_files_path</td>
<td>Path of the directory where are stored the mapping files (modus 2)</td>
</tr>
<tr>
<td>coupling_table_type</td>
<td>Coupling table type: 0 if coupling table is in a file or 1 if coupling table is in database</td>
</tr>
<tr>
<td>coupling_table_file_path</td>
<td>for coupling in file only: coupling table file path</td>
</tr>
<tr>
<td>coupling_table_connection</td>
<td>for coupling in database only: database connection string</td>
</tr>
<tr>
<td>coupling_table_user</td>
<td>for coupling in database only: database user</td>
</tr>
<tr>
<td>coupling_table_password</td>
<td>for coupling in database only: database password</td>
</tr>
<tr>
<td>coupling_table_tablename</td>
<td>for coupling in database only: database table name</td>
</tr>
<tr>
<td>unitsTranslationFile_path</td>
<td>Optional: Path of the XML file containing units translations to use in SDN NetCDF files “units” attribute.</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Parameter name (in RMConfiguration.properties file)</th>
<th>Description</th>
</tr>
</thead>
</table>
| 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) |

### 4.1. Data in modus 1 or 3

The Replication Manager can manage data using multiple storages:

<table>
<thead>
<tr>
<th>Modus</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>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.</td>
</tr>
<tr>
<td>2</td>
<td>Data are stored in a database. The corresponding ODV file will be created by the Replication Manager.</td>
</tr>
<tr>
<td>3</td>
<td>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_CDID_ID and will be splitted by the RM to have one file per LOCAL_CDID_ID.</td>
</tr>
</tbody>
</table>

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.
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**
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.

4.2. 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 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).
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.
4.2.1 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 file
- `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 §5.2.1 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.

<table>
<thead>
<tr>
<th>Attribute name</th>
<th>Declaration</th>
</tr>
</thead>
<tbody>
<tr>
<td>from</td>
<td>Parameter name from the P01 vocabulary and units from the P06 vocabulary MANDATORY for data columns EMPTY for metadata columns from=&quot;OBJECT, UNITS&quot; where OBJECT: URN of a concept from the SeaDataNet P01 vocabulary. 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=&quot;SDN:P01::ADEPZZ01, SDN:P06::ULAA&quot;</td>
</tr>
<tr>
<td>to</td>
<td>ODV column title: parameter local name and unit label MANDATORY for data columns MANDATORY for metadata columns to=&quot;SUBJECT, UNITS_TITLE&quot; 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=&quot;SDN:LOCAL:DEPH, meters&quot;</td>
</tr>
<tr>
<td>local</td>
<td>MANDATORY for data columns MANDATORY for metadata columns local=&quot;FIELD_NAME_IN_SQLQuery&quot; 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 &quot;&quot;) and the quality flag is set to &quot;9&quot; (it means “missing value” in L20). Example: local=&quot;DEPTH&quot;</td>
</tr>
</tbody>
</table>
| 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"

<table>
<thead>
<tr>
<th>instrument</th>
<th>OPTIONAL for data columns</th>
</tr>
</thead>
<tbody>
<tr>
<td>NOT PRESENT for metadata columns</td>
<td></td>
</tr>
<tr>
<td>instrument=&quot;INSTRUMENT_FIELD_NAME&quot;</td>
<td></td>
</tr>
<tr>
<td>Where:</td>
<td></td>
</tr>
<tr>
<td>INSTRUMENT_FIELD_NAME: The name of the field in the SQL query that contains the BODC L22 instrument code associated with this parameter.</td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>instrument=&quot;DEPTH_INTRUM&quot;</td>
<td></td>
</tr>
</tbody>
</table>

| 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.

<table>
<thead>
<tr>
<th>ODV Metadata column</th>
<th>ODV Column title</th>
<th>Code node</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cruise</td>
<td>Cruise</td>
<td><code>&lt;code from=&quot;&quot; to=&quot;SDN:LOCAL:Cruise&quot; local=&quot;CRUISE_FIELD_NAME&quot; /&gt;</code></td>
</tr>
<tr>
<td>Station</td>
<td>Station</td>
<td><code>&lt;code from=&quot;&quot; to=&quot;SDN:LOCAL:Station&quot; local=&quot;STATION_COLUMN_NAME&quot; /&gt;</code></td>
</tr>
<tr>
<td>Type</td>
<td>Type</td>
<td>OPTIONAL</td>
</tr>
</tbody>
</table>
|                     |                  | if “Type” node is omitted, default value for Type column in ODV file will be “*”.

sdn-userdesk@seadatanet.org – www.seadatanet.org

SeaDataNet - The pan-European infrastructure for marine and ocean data management
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.

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.

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

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:

```xml
<code from="" to="" local="" />
```
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>
<thead>
<tr>
<th>Field name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LOCAL_CDI_ID</td>
<td>The CDI local identifier, as included in the central CDI Directory at the SeaDataNet portal.</td>
</tr>
<tr>
<td>modus</td>
<td>Value equals to 1, 2 or 3. See §4.1. and §4.2</td>
</tr>
</tbody>
</table>

sdn-userdesk@seadatanet.org – www.seadatanet.org

SeaDataNet - The pan-European infrastructure for marine and ocean data management
format | Format of the original data file (ODV, CFOINT or MedAtlas, or other for some specific cases)

<table>
<thead>
<tr>
<th>Field name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>filename</td>
<td>The path of the original data file, relative to the &quot;root&quot; directory (see data_path parameter in RMConfiguration.properties file and Table 1 - Parameters concerning original data)</td>
</tr>
</tbody>
</table>

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

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

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

<table>
<thead>
<tr>
<th>Field name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>sql_query</td>
<td>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.</td>
</tr>
</tbody>
</table>
| connection_string | Oracle connection example
  `jdbc:oracle:thin:@<server>:<port>:<service_name>`
  `jdbc:oracle:thin:@195.178.224.89:1312:database_name`

MySQL connection example
  `jdbc:mysql://10.1.96.214:3306/dm_test`
  `jdbc:mysql://localhost:3306/database_name`

MS-SQL Server connection example
  `jdbc:sqlserver://<server>:<port>;databaseName=<database_name>`

Sybase connection example
  `jdbc:tds: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 describe which fields (ordered) must be used in each case.

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

<table>
<thead>
<tr>
<th>Field Number</th>
<th>Field name</th>
<th>(see detailed description in §4.3.1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>LOCAL_CDI_ID</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>modus</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>format</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>filename</td>
<td></td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Field Number</th>
<th>Field name</th>
<th>(see detailed description in §4.3.1)</th>
<th>(optional)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>LOCAL_CDI_ID</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>modus</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>format</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>filename</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>proxy</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>login</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>password</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Field Number</th>
<th>Field name</th>
<th>(see detailed description in §4.3.1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>LOCAL_CDI_ID</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>modus</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>format</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>sql_query</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>connection_string</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>login</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>password</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>mapping_file</td>
<td></td>
</tr>
</tbody>
</table>

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

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>
<thead>
<tr>
<th>Object</th>
<th>Parameter name (in RMConfiguration.properties file)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ready To Send directory</td>
<td>readyToSendCDIs_path</td>
<td>To store the zips of metadata to be submitted. Needs write access for the data manager.</td>
</tr>
<tr>
<td>Temporary directory</td>
<td>tmpDirectory_path</td>
<td>To generate and temporarily store the data</td>
</tr>
<tr>
<td>Queue directory</td>
<td>queueDirectory_path</td>
<td>To store the metadata and data when waiting for Import Manager</td>
</tr>
<tr>
<td>Archive directory</td>
<td>archive_path</td>
<td>To archive the metadata and data after ingestion. Can become large!</td>
</tr>
<tr>
<td>Production directory</td>
<td>production_path</td>
<td>To store the production restricted data. Sensitive data (do not exist elsewhere in SDN architecture, this is the only copy) Can become large!</td>
</tr>
<tr>
<td>Embedded database file</td>
<td>embeddedDatabase_path</td>
<td>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</td>
</tr>
<tr>
<td>Embedded database backup directory</td>
<td>embeddedDatabase_backupDirectory_path</td>
<td>To store the database backup file on Backup request via the dashboard.</td>
</tr>
</tbody>
</table>

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.
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 cancelled**
- **Batches in production**
- **LOCAL_CDI_IDs**
  - Local catalogue: list of LOCAL_CDI_IDs in production, with versions and formats

- **About**
  - Information about the RM, change log.
5.2. Checks and synchronization

Tools for checks and synchronization 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)

5.2.1. 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 availability:

- 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.

5.2.2. 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>
<thead>
<tr>
<th>Table 9 - Comparison between coupling table and CDI full catalogue</th>
</tr>
</thead>
<tbody>
<tr>
<td>Check</td>
</tr>
<tr>
<td>Each LOCAL_CDI_ID /Format couple in the coupling table must exist in the full catalogue</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 10 - Comparison between RM embedded database and CDI catalogue</th>
</tr>
</thead>
<tbody>
<tr>
<td>Check</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Description</td>
</tr>
<tr>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Each LOCAL_CDI_ID/format couple in the restricted catalogue table must exist in embedded database</td>
</tr>
<tr>
<td>Same LOCAL_CDI_ID/format couple must have the same access restriction values</td>
</tr>
<tr>
<td>Same LOCAL_CDI_ID/format couple must have the same PID values</td>
</tr>
<tr>
<td>Each restricted couple LOCAL_CDI_ID/format associated file must exist in PRODUCTION directory</td>
</tr>
</tbody>
</table>

Results can be read in the RMLLogChecker or downloaded as a csv file from the RM Summary page.
The csv file columns are detailed in the table below:

<table>
<thead>
<tr>
<th>Column name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Error level</td>
<td>ERROR or WARNING</td>
</tr>
<tr>
<td>Synchronization stage</td>
<td>COMPARE_COUPLING, COMPARE_INTERNAL_DB</td>
</tr>
<tr>
<td>message</td>
<td>Detailed message</td>
</tr>
<tr>
<td>localCdlId</td>
<td>Local_CDI_ID (read from CDI metadata catalogue or coupling table)</td>
</tr>
<tr>
<td>format</td>
<td>format</td>
</tr>
<tr>
<td>version</td>
<td>Version of the file (if exist)</td>
</tr>
</tbody>
</table>

### 5.3. 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.

- **Metadata preparation**
  - Interactive process
  - The Data Manager creates the metadata

- **Submission**
  - Interactive + automatic process
  - The Data manager submits the batch via the RM dashboard

- **Data preparation**
  - Automatic process
  - The RM generates the data

- **Ingestion**
  - Automatic process
  - The RM processes the ingestion workflow with the IM and the Cloud

**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!**
**5.3.1. 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
5.3.2. 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.
The Data Manager selects one or several zips and clicks on the “submit” button

RM checks the metadata zip compliance (see §5.2)

- **SUBMITTED**
- **VALID**
- **QUARANTINE**

RM moves metadata zip in queue directory

RM moves metadata zip in quarantine directory

- **WAITING FOR PREPARATION**
- **CANCELLED**

**legend**

- **BATCH STATUS**
- **METADATA STATUS**

---

**Figure 12 - Submission stage**

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

- **Summary**
- **Batches in progress**
- **Batches cancelled**
- **Batches in production**
- **LOCAL_CDI_IDs**
- **About**

#### Batches in readyToSend directory

<table>
<thead>
<tr>
<th>Created date</th>
<th>Name</th>
<th>Number of metadata files</th>
</tr>
</thead>
<tbody>
<tr>
<td>2019-08-16 11:21:06</td>
<td>enoc_length_max.zip</td>
<td>17000</td>
</tr>
<tr>
<td>2019-08-16 11:21:03</td>
<td>CDI_CESared_MED_CDU namoro.zip</td>
<td>2</td>
</tr>
<tr>
<td>2019-08-16 11:21:00</td>
<td>format_to.zip</td>
<td>2</td>
</tr>
<tr>
<td>2019-08-16 11:21:05</td>
<td>Universe_OFFPOINT_2.zip</td>
<td>2</td>
</tr>
<tr>
<td>2019-08-16 11:21:04</td>
<td>OCEAN_1844MOM3_adzoom.zip</td>
<td>1</td>
</tr>
<tr>
<td>2019-08-16 11:21:00</td>
<td>enoc_mediocetrino.zip</td>
<td>1</td>
</tr>
<tr>
<td>2019-08-16 11:21:04</td>
<td>enoc_昺ytyu_VL.zip</td>
<td>1</td>
</tr>
<tr>
<td>2019-08-16 11:21:04</td>
<td>odb_OFFPOINT_6.zip</td>
<td>1</td>
</tr>
<tr>
<td>2019-08-16 11:21:03</td>
<td>2K7F204_1.zip</td>
<td>1</td>
</tr>
<tr>
<td>2019-08-16 11:21:03</td>
<td>CDI_MED_UN_CDU_UN.zip</td>
<td>7</td>
</tr>
</tbody>
</table>

#### Current batch

#### Batches in queue

<table>
<thead>
<tr>
<th>Name</th>
<th>Batch number</th>
<th>Status</th>
<th>CDI status</th>
<th>CDI Files</th>
</tr>
</thead>
<tbody>
<tr>
<td>CDI_MED_R</td>
<td>2310-07-01 11:23:24</td>
<td>[Waiting for preparation]</td>
<td>[Valid]</td>
<td>CDI_MED_R (1 file)</td>
</tr>
<tr>
<td>CDI_MED_R_R</td>
<td>2310-07-01 11:23:24</td>
<td>[Waiting for preparation]</td>
<td>[Valid]</td>
<td>CDI_MED_R_R (3 files)</td>
</tr>
<tr>
<td>CDI_MED_UN_CDU_R</td>
<td>2310-07-01 11:23:24</td>
<td>[Waiting for preparation]</td>
<td>[Valid]</td>
<td>CDI_MED_UN_CDU_R (3 files)</td>
</tr>
</tbody>
</table>

**Figure 13** - “Batches in Progress” page of the RM dashboard: batches waiting in queue

#### 5.3.3. Data preparation

This stage is totally automatic.
Figure 14 - Data preparation stage for each batch

- WAITING FOR PREPARATION
- PREPARATION PENDING
- CHECK PENDING
- RM checks if one of the LOCAL_CDI_IDs is not already in the queue
- RM checks the CDI files (read LOCAL_CDI_IDs, formats and access restrictions)
- CHECK READY SUCCESS
- CHECK READY FAILURE
- GENERATION PENDING
- RM generates data files taking into account the information available in the coupling table (see §4).
- GENERATION ENDED
- RM checks if:
  - No error during generation
  - SDN files OCTOPUS check OK
- RM moves data to queue directory
- IN QUEUE ZONE
- IN QUEUE READY
- QUARANTINE
- CANCELLED
- legend
  - BATCH STATUS
  - METADATA STATUS
  - DATA STATUS
5.3.4. 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.

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

**Figure 16 - Ingestion stage, step 2: metadata archiving**
The data zip file is in the data harvest zone, ready to be harvested by 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 18 - Ingestion stage, step 4: archiving and versioning, production start-up

The cloud downloads the zip of unrestricted data

IM calls RM: update PIDs and versions

UPDATE PID PENDING

RM moves unrestricted data from HARVEST to ARCHIVE

ARCHIVED

RM moves restricted data to PRODUCTION directory

VERSIONED

RM calls IM: update PIDs read + remove unrestricted from local DB

PRODUCTION
### 5.4. Users data requests

After the ingestion process (§5.3), 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 20 - Workflow for the restricted data

Call from IM to RM: prepare restricted order

RM
- creates a zip with requested files
- put the zip in a HARVEST directory

RM call to IM: prepare restricted order ready

The cloud downloads the zip of restricted data

Call from IM to RM: delete_orders_rm
The data zip file is in the orders harvest zone, ready to be harvested by 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
<?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 pattern 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.s"),
    YEAR_MONTH_DAY_HOUR_MIN_SEC_FRACTION3("YYYY-MM-DDThh:mm:ss.sss");
    -->
</codes>
</root>
```

```xml
<code from="" to="SDN:LOCAL:Station" local="STATION" />
<code from="" to="SDN:LOCAL:Latitude, degrees_north" local="LATITUDE"/>
<code from="" to="SDN:LOCAL:Longitude, degrees_east" local="LONGITUDE"/>
<code from="" to="SDN:LOCAL:Cruise" local="PLATFORM"/>
<code from="" to="SDN:LOCAL:Station" local="STATION"/>
<code from="" to="" local=""/>
```
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) containing all sdn_reference for this file as a string

references must be stored in a unique String without separator.

Example:

-->
<code from="" to="sdn_reference" local="SDNREF_COL" />

Data columns

attributes:
from : [MANDATORY] "SDN:P01::xxxxxxxx, SDN:P06::yyyyy" P01 and P06 BODC vocabularies codes associated to the parameter
to : [MANDATORY] "SDN:LOCAL::xxx, yyy" or "SDN:LOCAL::xxxx" ODV column header where xxx is local parameter name and yyy local unit name
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::ADEFZU01, SDN:P06::ULAA" to="SDN:LOCAL:DEPH, meters" local="DEPH" qflag="QV_DEPH" instrument="DEPH_INSTRUM" fall_rate="DEPH_FALLRATE" />
<code from="SDN:P01::TEMPXU01, SDN:P06::UPAA" to="SDN:LOCAL:TEMP, Celsius degree" local="TEMP" qflag="QV_TEMP" />
<code from="SDN:P01::PSLZU01, 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>