**IEDC download options: [need to code the download formatting together!]**

1. **Data package (csv), table**: returns zip folder with two csv files: One with the data as table, one with the metadata
2. **Data package (csv), list**: returns zip folder with two csv files: One with the data as list, one with the metadata
3. **Data package (JSON)**: returns a JSON file containing both the data and the metadata, in a simple format.
4. **ODYM\_MFA format, table**: returns data in ODYM format (.xlsx or .xls file with one sheet for the data and one with the metadata)
5. **ODYM\_MFA format, list**: returns data in ODYM format (.xlsx or .xls file with one sheet for the data and one with the metadata)
6. **Python (.py) array**: Python code that, when run, creates a dictionary with the metadata and a numpy array with the data
7. **Python (.py) Pandas dataframe:** Python code that, when run, creates a dictionary with the metadata and a pandas dataframe with the data.

**Below, we describe the necessary specifications of each download option.**

**All items written in [] indicate the data and metadata retrieved from the database**

**The general principles of data export are:**

* **Users can select one dataset and export all values contained in that dataset**
* **In all export options the entire dataset table entry and all data contained in that dataset are exported.**

1. **Data package (csv) as table: DELAYED! [need to first figure out how to automatically configure table]**

The export routine creates two csv files and zips them together to return a .zip folder (Do this later if zipping together is too complicated for web app!).

**File 1: [DataSet\_Name]\_Metadata.csv:**

Create csv copy of dataset table entry:

"dataset\_id","[dataset\_id]"

"dataset\_name","[dataset\_name]"

"dataset\_version","[dataset\_version]"

…

For aspect x and aspect x classification: Only write to csv if not NULL

For Reserve 1 to Reserve 5: Do not write to csv.

**File 2: [DataSet\_Name]\_Data\_Table.csv:**

…

1. **Data package (csv) as list:**

The export routine creates two csv files and zips them together to return a .zip folder (Do this later if zipping together is too complicated for web app!).

**File 1: [DataSet\_Name]\_Metadata.csv:**

Same as for option 1.

**File 2: [DataSet\_Name]\_Data\_List.csv:**

Create header line:

"aspect1","aspect2","aspect3",[…],"value","unit\_nominator","unit\_denominator","stats\_array\_1", "stats\_array\_2","stats\_array\_3","stats\_array\_4","comment"

Then, create one line for each data item in the dataset. Drop all aspects that are NULL.

1. **Data package (JSON)**:

Similar to csv file, just write one JSON file with the two sections:

{

{

// Metadata

"dataset\_id":"[dataset\_id]"

"dataset\_name":"[dataset\_name]"

"dataset\_version":"[dataset\_version]"

…

}

{

// Data

"Header":["aspect1","aspect2","aspect3",[…],"value","unit\_nominator","unit\_denominator","stats\_array\_1", "stats\_array\_2","stats\_array\_3","stats\_array\_4","comment"]

"Row1":["[aspect1]","[aspect2]","[aspect3]",[…],[value],"[unit\_nominator]","[unit\_denominator]",[stats\_array\_1], [stats\_array\_2],[stats\_array\_3],[stats\_array\_4],["comment"]]

"Row2":["[aspect1]","[aspect2]","[aspect3]",[…],[value],"[unit\_nominator]","[unit\_denominator]",[stats\_array\_1], [stats\_array\_2],[stats\_array\_3],[stats\_array\_4],["comment"]]

…

}

}

1. **ODYM\_MFA format, table**: **DELAYED! [need to first figure out how to automatically configure table]**

Tbd.

1. **ODYM\_MFA format, list:**

Create [dataset\_name].xls, with two sheets included: ‘Cover’ and ‘Values\_Master’. Below, instructions for filling cells A1, etc. are given:

1. ‘Cover’:

A1: “ODYM Data File”

A2: “Format\_Version” B2: “V\_IEDC\_1.0”

A3: “Dataset\_Name” B3: [dataset\_name]

A4: “Dataset\_Description” A4: [dataset\_description]

A5: “Dataset\_Unit” B3: “LIST”

A6: “Dataset\_Uncertainty” B3: “LIST”

A7: “Dataset\_Comment” B3: “LIST”

Then, from A8 to Axx, the complete dataset table entry, except for dataset\_name and dataset\_description. Ignore aspects that are NULL:

A8: "dataset\_id" B8: [dataset\_id]

A9: "dataset\_version" B9:[dataset\_version]

…

Then, from A77 on, specify the aspects so that the OYDM software can read it:

A77: “Dataset\_RecordType” B77: “LIST” C77: “No\_Rows” D77: [dataset\_size]

A78: “Aspect\_classifications” B78: “Aspects\_Meaning” C78: “DATA” D78: “DATA\_Info”

C79: “Value” D79: “Data item”

C80: “Unit” D80: “Unit of data item”

C81: “Stats\_Array\_String” D81: “String describing uncertainty distribution (http://stats-arrays.readthedocs.io/en/latest/)”

C82: “Comment” D82: “Comment on data item”

Then, for all non-NULL aspects, fill A79 and below with the aspect classifications and aspects meaning:

A79: [Aspect1\_classification] B79: [Aspect1]

A80: [Aspect2\_classification] B80: [Aspect2]

…

1. ‘Values\_Master’:

First row: A1: [Aspect1\_classification], B1: [Aspect2\_classification], C1: [Aspect3\_classification], […]

E1: “Value”, F1: “Unit”, G1: “Stats\_array\_string”, H1: “Comment”

Then, from row 2 onwards, insert the data table entries:

Aspects: A2: [Aspect1], B2: [Aspect2], …,

Value: [data.value] as is

Unit: “([unit\_nominator])/([unit\_denominator])”

Stats\_array\_string: “[stats\_array\_1];[stats\_array\_2];[stats\_array\_3];[stats\_array\_4]”

Comment: [Comment]

1. **Python (.py) array:**

Tbd.

1. **Python (.py) Pandas dataframe:**

Tbd.