

Metadata JSON Description Document (version 1.2)

Introduction

Definition

The document describes the structure of the metadata. The metadata is represented as a file in JSON format. This document describes the structure of metadata of the reporting tools for the version of the ETF Reporting Tool released on 06 November 2024. The document will be updated as and when there are changes to the metadata structure. This document provides a general understanding of the metadata and its structure.

Version history and applicability

This is version 1.2 of the document. It is applicable to the metadata file for the ETF GHG Inventory reporting tool:

Document version	Date	Applicable Metadata version	Key changes
0.1	2023-09-15	1.17.12	<ul style="list-style-type: none">Initial version
1.0	2024-07-12	1.27.4	<ul style="list-style-type: none">Updated type of dimensions included new base, added new examplesUpdated dimension with the new attributes, added new examplesUpdated dimension instance with the new attributes, added new examples<ul style="list-style-type: none">Updated drop-down attributes and example
1.2	2024-11-06	1.29.8	<ul style="list-style-type: none">Updated section with “drop-down” added new attribute to aggregate selection of multiple drop down lists. The attribute name is “dropdown_agg”Updated section with “variables” added new attribute “comment_allowed “ specifying whether the official comment can be provided (CRT), or footnote related to the cell within the data entry grid is allowed (NDC, FTC)Updated list of “operations” used for variable calculationsUpdated section with “settings” (version settings) added new attribute “roles”. The attribute is used by the application to control which role is able to change the setting

Contents

- [Introduction](#)
 - [Definition](#)
 - [Version history and applicability](#)
- [Contents](#)
- [1. Metadata](#)
 - [Example: High-level structure](#)
- [2. Version](#)
 - [Example: Version section](#)
- [3. Dimensions and fact approach](#)
 - [3.1. Dimensions](#)
 - [Example of dimensions](#)

- 3.2. Type of the dimension
 - Example of Type of Dimensions
- 3.3. Dimension Instance
 - Example of dimension instances
- 4. Drop-down
 - Example: Drop-down
- 5. Unit and GWP gas conversions
 - 5.1. Unit of measurement and coefficients
 - Example of unit conversion coefficients
 - 5.2. GWP Gas Conversion
 - Example of GWP conversions
- 6. Variable fact
 - 6.1. Variables
 - Example: variable
 - 6.2. Variable calculations
 - Example of operation definitions
 - 6.3. Operations
 - Example of operation definitions
 - 6.4. Variable validations
- 7. Navigation and Data Entry
 - 7.1. Node
 - Example: Node section
 - 7.1.1. Customization of nodes and country-specific data
 - 7.2. Grid
 - Example: Grid section
 - 7.2.1. Customization of grids
 - 7.2.2. Visual appearance
- 8. Reporting tables
 - Example: Reporting tables
- 9. Version settings
 - Example: Version setting sample structure

1. Metadata

Metadata is a data which describes the actual data. For example, a bank statement has definitions of different statuses, and the data that describes the status is metadata; coordinates of the point on the map have longitude and latitude to find the place; and the introduction of longitude and latitude is a metadata definition.

The metadata file has the following high-level entities defined:

- version - [mandatory] describes the version of the metadata
- type - [optional] describes the type of dimensions
- dimension - [mandatory] describes the list of dimensions
- dimension_instance - [mandatory] describes a tree list of dimension instances
- drop_down - [mandatory] describes the list of drop-downs
- gwp - [mandatory] describes global warming potential coefficients
- unit - [mandatory] describes the list of units of measurement
- unit_conv - [mandatory] describes the list of coefficients to convert from one unit to another unit of measurement
- variable - [mandatory] describes the list of variables (data points)
- operation - [mandatory] describes the list of operations used in the application
- variable_calc - [mandatory] describes a tree of variable calculation graph

- node - [mandatory] describes a tree list of nodes (navigation tree)
- grid - [mandatory] describes a tree list of data entry grids
- reporting_tables_tree - [mandatory] describes a reporting table tree
- settings - [mandatory] describes a list of version-specific settings

Example: High-level structure

```

1  {
2    "Metadata": [
3      {
4        "version": {...},
5        "type": [...],
6        "dimension": [...],
7        "dimension_instance": [...],
8        "drop_down": [...],
9        "gwp": [...],
10       "unit": [...],
11       "unit_conv": [...],
12       "variable": [...],
13       "operation": [...],
14       "variable_calc": [...],
15       "node": [...],
16       "grid": [...],
17       "reporting_tables_tree": [...],
18       "settings": [...]
19     }
20   ]
21 }
```

2. Version

This section describes the version of the metadata file and related information. The following attributes are defined:

- name - [mandatory] name of the version
- publication_date - [mandatory] date of the publication of the metadata file
- link - [mandatory] link to this document describing metadata and the changes
- version - [mandatory] a unique code of the version
- appId - [mandatory] application identifier (e.g. CRT)

Example: Version section

```

1  "version": {
2    "name": "1.12.9",
3    "publication_date": "10/10/2022",
4    "link": "https://unfccc.int/ETF/metadata",
5    "version": "3849a110ee80f9f13f9892f8f5f59936",
6    "appId": "CRT"
7  }
```

3. Dimensions and fact approach

Each type of metadata has a set of dimensions describing the facts. For example, if we need to define emissions from the energy sector for CO₂ in kiloton (kt) we will have the **energy sector** as a dimension instance of a *category* dimension, **emission** as a dimension instance of *measure* dimension, **CO₂** as a dimension instance of *gas* dimension, **kt** as a dimension instance of a *unit of measurement* dimension. The fact, in this case, will be a unique combination of dimension instances. If we go further, we have the following gases for the energy sector to report: **CH₄**, **N₂O**, **aggregate GHG** and so on. In this case, we will have definitions for the following facts:

- Emission from the energy sector for CO₂ in kt (described in the initial example);
- Emission from the energy sector for CH₄ in kt;
- Emission from the energy sector for N₂O in kt;
- Emission from the energy sector for Aggregate GHG in kt CO₂ equivalent;

Now, going further, we have additional combinations in addition to the **Emission: Activity Data (AD) and Implied Emission Factor (IEF)**. So the combinations will expand:

- Activity Data from the energy sector for CO₂ in kt;
- Activity Data from the energy sector for CH₄ in kt;
- Activity Data from the energy sector for N₂O in kt;
- Implied Emission Factor from the energy sector for CO₂ in kt;
- Implied Emission Factor from the energy sector for CH₄ in kt;
- Implied Emission Factor from the energy sector for N₂O in kt;
- and so on ...

In this example, the definition of Activity Data from the energy sector for CO₂ in kt, Activity Data from the energy sector for CH₄ in kt, Activity Data from the energy sector for N₂O in kt does not make sense from a substantive point of view. A single activity data is provided for all gases and has a different unit of measurement. The combination that makes sense from a substantive point of view will be only one fact: Activity Data from the energy sector in TJ. In this example, if we use the cube approach for identifying facts (possible unique combinations), it will give us a lot of combinations which does not make sense from the substantive point of view. In order to limit the set of these combinations, we introduce an entity for the fact and call it **Variable**. When you get to this point, you may ask yourselves, why do we introduce dimensions and facts? We can have only one entity that describes the information instead of creating different dimensions and complicating life. You may be right from one side, but the challenge comes next when you need to perform the following:

- convert data from one gas into the equivalent of CO₂ gas and each gas has its own GWP coefficient for conversion.
- Aggregations of values should consider category, measure, gas, and other dimensions hierarchies.
- Formula calculation for IEF consists of **Emissions** divided by **Activity Data** (Emissions/Activity Data).
- Unit of measurement would need to be converted within the same group, for example, Gg to kg, t, Mt, and so on. TJ to calories and so on.
- Searching for information by specific dimension

Here is an example of category hierarchy under the **energy sector** used for aggregation:

Sectors/Totals

1. Energy

1.A. Fuel combustion activities (sectoral approach)

1.A.1. Energy industries

1.A.1.a. Public electricity and heat production

1.A.1.a.i. Electricity generation

1.A.1.a.ii. Combined heat and power generation

1.A.1.a.iii. Heat plants

1.A.1.b. Petroleum refining

1.A.1.c. Manufacture of solid fuels and other energy industries

1.A.1.c.i. Manufacture of solid fuels

1.A.1.c.ii. Oil and gas extraction

1.A.1.c.iii. Other energy industries

1.A.2. Manufacturing industries and construction

1.A.2.a. Iron and steel

1.A.2.b. Non-ferrous metals

1.A.2.c. Chemicals

1.A.2.d. Pulp, paper and print

1.A.2.e. Food processing, beverages and tobacco

1.A.2.f. Non-metallic minerals

1.A.2.g. Other

1.A.2.g.i. Manufacturing of machinery

1.A.2.g.ii. Manufacturing of transport equipment

1.A.2.g.iii. Mining (excluding fuels) and quarrying

1.A.2.g.iv. Wood and wood products

1.A.2.g.v. Construction

1.A.2.g.vi. Textile and leather

1.A.2.g.vii. Off-road vehicles and other machinery

1.A.2.g.viii Other

1.A.2.g.viii Other - Template

Here is an example of gas hierarchy under **Aggregate GHG** used for aggregation:

Aggregate GHGs

CO₂

CH₄

N₂O

Aggregate F-gases

HFCs and PFCs

HFCs

HFC-23

HFC-32

HFC-41

HFC-43-10mee

HFC-125

HFC-134

HFC-134a

HFC-143

HFC-143a

HFC-152

HFC-152a

HFC-161

HFC-227ea

HFC-236cb

HFC-236ea

HFC-236fa

HFC-245ca

HFC-245fa

HFC-365mfc

Unspecified mix of HFCs

PFCs

CF₄

C₂F₆

The same applies to other dimensions.

3.1. Dimensions

Entity dimension is used to describe fact (**Variable**). It has the following attributes:

- Global Unique Identifier of the dimension instance (GUID)
- Internal identifier a numeric index for internal reference (ID)
- Name of the dimension
- Reference to the type of dimension
- Order of the dimension

Examples of dimension: category, measure, gas, classification and etc.

Example of dimensions

```
1      "dimension": [  
2      {  
3          "id": 1,  
4          "uid": "584f6b65-1f6c-4b29-b643-1d1f175ed514",  
5          "name": "NAVIGATION",  
6          "order_number": 1,  
7          "type": 4  
8      },  
9      {  
10         "id": 2,  
11         "uid": "71182b9f-9e95-42b4-86f3-3ee3c15c37ed",  
12         "name": "CLASSIFICATION",  
13         "order_number": 2,  
14         "type": 4  
15     },  
16     {  
17         "id": 3,  
18         "uid": "c2d87f31-4c08-4c57-8f67-3c53b825d0e5",  
19         "name": "MEASURE",  
20         "order_number": 3,  
21         "type": 4  
22     },  
23     {  
24         "id": 4,  
25         "uid": "3331bc46-45e9-452a-a022-6700b8605cf3",  
26         "name": "GAS",  
27         "order_number": 4,  
28         "type": 4  
29     },  
30     {  
31         "id": 6,  
32         "uid": "47e8722a-08ee-4654-9df3-f40850848f54",  
33         "name": "SECTOR",  
34         "order_number": 7,  
35         "type": 4  
36     },  
37     {  
38         "id": 7,  
39         "uid": "45ea7a3a-20af-4573-bc77-6081e7cab31a",  
40         "name": "PARAMETER",  
41         "order_number": 6,  
42         "type": 4  
43     },  
44     {  
45         "id": 10,
```

```
46     "uid": "16f9737d-7044-448e-978b-7a6d7b396621",
47     "name": "KCA_NAVIGATION",
48     "order_number": 1,
49     "type": 3
50 },
51 {
52     "id": 11,
53     "uid": "ba115d58-9cf0-4fa3-a405-1e686d81b7a7",
54     "name": "KCA_FORMULA",
55     "order_number": 2,
56     "type": 3
57 },
58 {
59     "id": 77,
60     "uid": "2305b92c-629d-4dd2-a7bc-89997999d4b4",
61     "name": "VALIDATION_RULE",
62     "order_number": 1,
63     "type": 1
64 },
65 {
66     "id": 84,
67     "uid": "9be82a52-c18d-4b63-b12d-a635e36a1b69",
68     "name": "GRID_LINE_NAME",
69     "order_number": 1,
70     "type": 2
71 },
72 {
73     "id": 85,
74     "uid": "0b4b05b5-d3da-4f0c-95e1-991484102366",
75     "name": "CATEGORY_SPECIFIED",
76     "order_number": 1,
77     "type": 6
78 },
79 {
80     "id": 86,
81     "uid": "231cbb66-2b0e-4f16-97ab-65ebb7a0d8a5",
82     "name": "CLASSIFICATION_SPECIFIED",
83     "order_number": 2,
84     "type": 6
85 },
86 {
87     "id": 87,
88     "uid": "19b48dce-f5e8-4776-a4f6-ed9c39cfe4f0",
89     "name": "MEASURE_SPECIFIED",
90     "order_number": 3,
91     "type": 6
92 },
93 {
94     "id": 88,
95     "uid": "8f7e3a42-ac41-4fc8-96c2-70b17ddcb4c2",
96     "name": "GAS_SPECIFIED",
97     "order_number": 4,
98     "type": 6
99 },
100 {
101     "id": 93,
102     "uid": "bbb12ef2-7be0-4e8e-a126-425a32359c79",
103     "name": "FOOTNOTE",
```

```

104         "order_number": 2,
105         "type": 2
106     },
107     {
108         "id": 96,
109         "uid": "6698016f-84f4-4d6f-8785-782ea3bca084",
110         "name": "TOOLTIP",
111         "order_number": 3,
112         "type": 2
113     }
114 ]

```

3.2. Type of the dimension

There are four types of the dimensions:

- Calculation- used for calculations
- Visual - used for visual representation
- Filter - used for filtering and search of the information
- base - used as based dimensions for variables

Example of Type of Dimensions

```

1     "type": [
2         {
3             "id": 1,
4             "name": "calculation"
5         },
6         {
7             "id": 2,
8             "name": "visual"
9         },
10        {
11            "id": 3,
12            "name": "filter"
13        },
14        {
15            "id": 4,
16            "name": "base"
17        }
18    ],

```

3.3. Dimension Instance

Entity Dimension Instance represents a set of instances of the same dimension in the hierarchy. It has the following attributes:

- Reference to the dimension using internal ID;
- Global Unique Identifier of the dimension instance (GUID)
- Internal identifier a numeric index for internal reference (ID)
- Name of the dimension instance;
- List of children of dimension instances (representing hierarchy)
- Order of the dimension instance to define the order of children under the parent dimension instance
- Level of the hierarchy 16-digit number, coding levels by bits. For example, level 1 will have the number "1000000000000000", level 2 "1100000000000000" and so on. (obsolete)
- Code is a short code for the dimension instance

- `is_cs_common` is used for common dimension instances across all the parties

Example of dimension instances

```
{
  "id": 1,
  "uid": "711ab9da-13cd-44d8-b8f4-33a954171186",
  "name": "Sectors/Totals",
  "dimension_id": 1,
  "code": "",
  "order_number": 1000000000,
  "level": null,
  "is_cs_common": false,
  "children": [
    {
      "id": 2,
      "uid": "3665c27e-d055-47d7-8393-5f934f3ced9d",
      "name": "1. Energy",
      "dimension_id": 1,
      "code": "",
      "order_number": 1100000000,
      "level": null,
      "is_cs_common": false,
      "children": [
        {
          "id": 3,
          "uid": "82239b33-699b-4187-b992-4a0883d99952",
          "name": "1.A. Fuel combustion activities (sectoral approach)",
          "dimension_id": 1,
          "code": "",
          "order_number": 1110000000,
          "level": null,
          "is_cs_common": false,
          "children": [
            {
              "id": 4,
              "uid": "f85540ba-43dc-4724-8d4a-f8673fdc9e0c",
              "name": "1.A.1. Energy industries",
              "dimension_id": 1,
              "code": "",
              "order_number": 1111000000,
              "level": null,
              "is_cs_common": false,
              "children": [
                {
                  "id": 5,
                  "uid": "5bab6e8b-633c-45ba-9206-82455177330a",
                  "name": "1.A.1.a. Public electricity and heat production",
                  "dimension_id": 1,
                  "code": "",
                  "order_number": 1111100000,
                  "level": null,
                  "is_cs_common": false,
                  "children": [
                    {
                      "id": 6,
                      "uid": "964719e5-8920-45e7-bb55-053a7be3bd69",
                      "name": "1.A.1.a.i. Electricity generation",
                      "dimension_id": 1,
```

```

56         "code": "",
57         "order_number": 1111110000,
58         "level": null,
59         "is_cs_common": false,
60         "children": []
61     },
62     {
63         "id": 7,
64         "uid": "afa4271d-606d-4502-b8ae-3dbf84b69920",
65         "name": "1.A.1.a.ii. Combined heat and power generation",
66         "dimension_id": 1,
67         "code": "",
68         "order_number": 1111120000,
69         "level": null,
70         "is_cs_common": false,
71         "children": []
72     }
73 ]
74 }
75 ]
76 }
77 ]
78 }

```

4. Drop-down

The drop-down entity in the metadata specifies drop-down lists that are used and referenced within the data entry grids for cells (variables) and lines (line). The entity has the following attributes:

- id - unique identifier for the drop-down
- name - name of the drop-down
- is_extendable - a flag specifies if the drop-down is extendable by the user with its own items. If set to *true* that means there are pre-defined items in the list, and it can be extended with the newly added items specified by the user. If set to *false* then only pre-defined list items can be used in the drop-down
- is_NK - a flag specifies if the drop-down used as a Notation Keys (example of "is_NK" set to true: Method, EF Used and Calorific Value)
- items - an array of pre-defined items in the drop-down
 - UID as a global unique identifier
 - name - name of the item, it is shown to the user for selection
 - order_number - order of appearance, lowest goes first
- dropdown_agg - is an array type, mandatory in case it is not used then it is set to empty array, with list of IDs comma separated referencing the list of the drop downs to be used for aggregation (used in FTC metadata, for the "recipient" field to aggregate selection from three drop downs lists)

Here is an example of the drop-down:

Example: Drop-down

```

1  {
2      "id": 2,
3      "name": "METHOD",
4      "is_extendable": false,
5      "is_NK": true,
6      "items": [
7          {

```

```
8      "uid": "c0750230-51b8-4985-93b2-c1ccc6025136",
9      "name": "D",
10     "order_number": 0
11   },
12   {
13     "uid": "c83e387f-9101-49e4-b562-2d39eec6eed3",
14     "name": "T1",
15     "order_number": 1
16   },
17   {
18     "uid": "43451736-8433-41b0-8e10-8ee9a520a0b4",
19     "name": "T1a",
20     "order_number": 2
21   },
22   {
23     "uid": "869d5c9f-ce1c-4b54-894b-9e2006daf0ef",
24     "name": "T1b",
25     "order_number": 3
26   },
27   {
28     "uid": "9018aa50-1971-4336-bbe1-a8db0cb4d9f6",
29     "name": "T1c",
30     "order_number": 4
31   },
32   {
33     "uid": "5283f8a1-9357-4216-a880-5c4052a49730",
34     "name": "T2",
35     "order_number": 5
36   },
37   {
38     "uid": "cc3cde06-66ef-47b6-8177-8de8b46dbe30",
39     "name": "T3",
40     "order_number": 6
41   },
42   {
43     "uid": "d275d7fe-b35a-4d4c-b612-20e926f9c6d4",
44     "name": "CR",
45     "order_number": 7
46   },
47   {
48     "uid": "4931c03d-9ad3-4ad7-b8e4-7e7acd2446b4",
49     "name": "CS",
50     "order_number": 8
51   },
52   {
53     "uid": "bad135e2-768e-4473-b6e3-fb8c7e58871b",
54     "name": "M",
55     "order_number": 9
56   },
57   {
58     "uid": "629ff47a-522a-49a5-a6ab-9c6062cfc9d5",
59     "name": "RA",
60     "order_number": 10
61   },
62   {
63     "uid": "bfb746b0-993a-47a9-9a91-b3a97503e795",
64     "name": "OTH",
65     "order_number": 11
```

```

66     }
67   ]
68 }

```

5. Unit and GWP gas conversions

Almost every variable has an attribute with a unit of measurement. Units are organized within the conversion group. For example, groups of mass, volume, energy, and so on. In some cases, if the unit is selected as a mass unit of CO₂ equivalent, then gas has to be taken into account. All emissions from gases can be converted to CO₂ equivalent, and for that additional coefficient, **Global Warming Potential (GWP)** needs to be included. For example, emissions from CH₄ can be converted to the CO₂ emission equivalent. The formula will look as follows:

$$\text{value in CO}_2 (t) = \text{value in CH}_4 (t) * GWP$$

Now, let's include the unit of measurement conversion

$$\text{value in CO}_2 (t) = \text{value in CH}_4 (kg) * \text{Unit Conversion Coefficient (UCC)} * GWP$$

where UCC is a conversion from the unit (kg) to (t) = 1000

5.1. Unit of measurement and coefficients

To reference the unit of measurement, a new entity Unit is introduced with the following attributes:

- name of the unit (displayed to the user), examples: "kg", "t", "Gg", "Mg", etc.

Unit conversion coefficients are kept in the entity "**unit_conv**" with the following attributes:

- coefficient is a multiplier for converting units from one to another
- from is a reference to the unit from which conversion has to occur
- to is a reference to the unit to convert

An example of the conversion from **t** to **kg** will look like:

- unit_from: unit ID of **kg**
- unit_to: unit ID of **t**
- multiplier (coefficient): 1000

Example of unit conversion coefficients

```

1  "unit_conv": [
2    {
3      "unit_from": 1,
4      "unit_to": 1,
5      "multiplier": 1.0
6    },
7    {
8      "unit_from": 1,
9      "unit_to": 2,
10     "multiplier": 0.001
11   },
12   {
13     "unit_from": 1,
14     "unit_to": 3,
15     "multiplier": 1E-06
16   },
17   {

```

```

18     "unit_from": 1,
19     "unit_to": 4,
20     "multiplier": 0.001
21 },
22 {
23     "unit_from": 1,
24     "unit_to": 5,
25     "multiplier": 1E-06
26 },
27 ....
28 ]

```

5.2. GWP Gas Conversion

Entity GWP is introduced to store the coefficient between gas conversion. Please note that there is only one conversion required from all gases to CO₂. There are no conversions required to convert CH₄ to N₂O. The following attributes are defined for gas conversion:

- coefficient is a coefficient to convert gas to CO₂ equivalent;
- gas_id: reference to the gas ID, actual gas that needs to be converted to CO₂.

In the case of CO₂ to CO₂ conversion coefficient will be equal to 1.

Example of GWP conversions

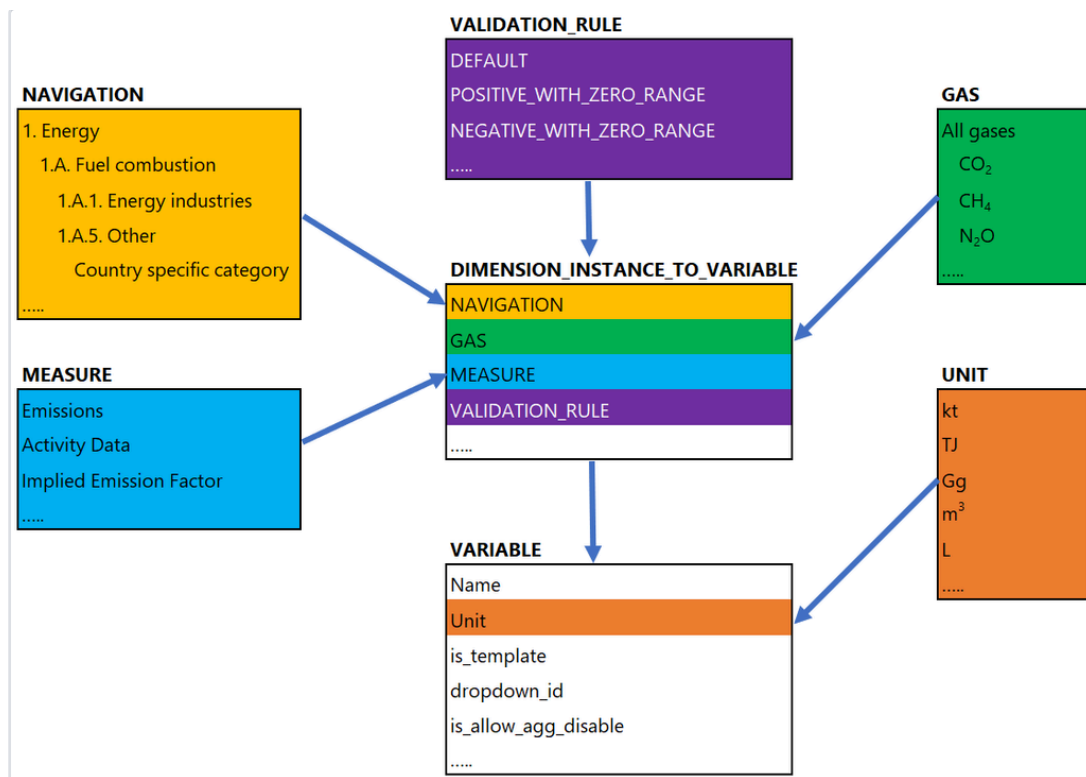
```

1     "gwp": [
2         {
3             "id": 1,
4             "gas_id": 1051,
5             "coefficient": 28.0
6         },
7         {
8             "id": 2,
9             "gas_id": 1052,
10            "coefficient": 1.0
11        },
12        {
13            "id": 3,
14            "gas_id": 1054,
15            "coefficient": 265.0
16        },
17        ...
18    ]

```

6. Variable fact

Entity variable is used to define possible combinations of dimensions and related dimension instances. The entity to store information of combination is stored within the variable "**dimension_instances**" attribute links variables with the list of dimension instances.



6.1. Variables

It is a fact table and is used as a reference in other entities. It has the following attributes:

- Global Unique Identifier of the dimension instance (GUID)
- Internal identifier a numeric index for internal reference (ID)
- name (generated based on the names of dimension instances)
- unit_id is a reference to the unit of measurement for the variable
- is_calculated is an attribute when true the variable is calculated and takes part in the variable calculation
- parent_id (obsolete)
- template_var_id is a template variable internal identifier used in country-specific variables referencing the template
- template_var_uid is a template variable global unique identifier used in country-specific variables referencing the template
- is_template defines when the variable is a template the value set to true
- dropdown_id is a reference to the drop-down internal identifier used for storing values for the variable
- is_allow_agg_disabled is a flag when set to true defines that the variable can be disabled for aggregation and the aggregated value can be overwritten
- is_editable is a flag when set to true is editable
- role_node_uid is a reference to the node from where the role is inherited and applied to the current variable (used only by application)
- nke_allowed is a flag when set to true allows notation key explanation to be provided and will be included in the reporting Table 9 (CRT only)
- node_uid reference to the node that the variable belongs to
- dimension_to_change is an array to the list of dimensions available for customization (obsolete - application related)
- cs_common_node is an array of country-specific node UIDs (used only by application)
- linked_nodes is an array of node UIDs dependent on the entering of notation keys "C" and "FX" and overriding of aggregated value (used by the application)
- dimension_instances is an array to store information about unique dimension instances combination, it links variables with the list of dimension instances

- comment_allowed is a flag (True, False) in CRT to specify if official comment can be provided, for FTC and NDC to specify custom footnotes at cell level within the data entry grid

Example: variable

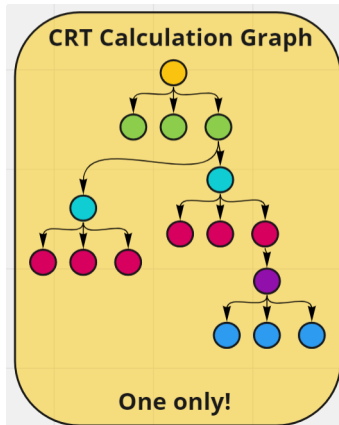
```

1      "variable": [
2      {
3          "id": 1,
4          "uid": "50b1a92e-62a4-4378-b5d7-006be67349fa",
5          "name": "[2.G.2.b. Accelerators][no classification][Emissions][PFCs][no parameter][t CO2
equivalent]",
6          "unit_id": 93,
7          "is_calculated": true,
8          "parent_id": null,
9          "template_var_id": null,
10         "template_var_uid": null,
11         "is_template": false,
12         "dropdown_id": null,
13         "is_allow_agg_disable": false,
14         "is_editable": false,
15         "role_node_uid": "fed65b84-cdad-4e38-8848-ea6af3c391bc",
16         "nke_allowed": false,
17         "node_uid": "bbf45eca-6575-4172-b26d-a62ea455e47c",
18         "dimension_to_change": [],
19         "cs_common_node": [],
20         "linked_nodes": [],
21         "comment_allowed" : True,
22         "dimension_instances": [
23             {
24                 "id": 215,
25                 "dimension_id": 1
26             },
27             {
28                 "id": 597,
29                 "dimension_id": 2
30             },
31             {
32                 "id": 837,
33                 "dimension_id": 3
34             },
35             {
36                 "id": 1056,
37                 "dimension_id": 4
38             },
39             {
40                 "id": 38538,
41                 "dimension_id": 6
42             },
43             {
44                 "id": 1011,
45                 "dimension_id": 7
46             },
47             {
48                 "id": 7390,
49                 "dimension_id": 77
50             }
51         ]
52     },
53     ...

```

6.2. Variable calculations

Calculation rules for variables are defined in the entity "**variable_calc**"; it is a B-Tree structure (calculation graph). The tree defines the dependencies of variables and calculation operations between them.



For example, we need to aggregate the results of variables B and C and place the results in variable A. In this case, variable A has two children variables B and C, and variables B and C have a reference to the calculation operation "AGG" that needs to be performed with it. For the root variables, no operations are performed, and direct children should always have the same operation to be performed. Due to business rules, in some cases, we would need to have intermediate calculation results. A specific variable (flagged as intermediate) exists for the result of intermediate calculation but is not presented or shown on the user interface. The following attributes are defined:

- `variable_uid` is a reference to the global Unique Identifier of the variable (GUID)
- `intermediate_calc` - true/false if it is an intermediate calculation
- `operation_id` is a reference to the operation
- `order_number` - order used for calculation operation. For example, if it is set to 1, then the variable should be taken first in operation (e.g. for the *DIV* divide operation)
- `is_unit_conv` is a flag when set to true requires a unit conversion for calculation
- `is_gwp` is a flag when set to true requires GWP multiplication
- `variable_calc` is an array of children variable calculations (B-Tree)

Example of operation definitions

```

1  "variable_calc": [
2      {
3          "variable_uid": "b238baa2-ad2c-4f77-889d-006c9e737f05",
4          "intermediate_calc": false,
5          "operation_id": null,
6          "order_number": null,
7          "is_unit_conv": false,
8          "is_gwp": false,
9          "variable_calc": [
10             {
11                 "variable_uid": "898baecd-d0dd-4971-bf45-cd94923c10fe",
12                 "intermediate_calc": true,
13                 "operation_id": 2,
14                 "order_number": 1,
15                 "is_unit_conv": true,
16                 "is_gwp": false,
17                 "variable_calc": [
18                     {

```



```

19     "variable_uid": "f7125b20-e2e1-4e0c-b8ce-e59c68c5752e",
20     "intermediate_calc": false,
21     "operation_id": 1,
22     "order_number": 1,
23     "is_unit_conv": false,
24     "is_gwp": false,
25     "variable_calc": [
26         {
27             "variable_uid": "684be5cd-0534-4516-bfef-28d2125f0933",
28             "intermediate_calc": false,
29             "operation_id": 1,
30             "order_number": 1,
31             "is_unit_conv": false,
32             "is_gwp": false,
33             "variable_calc": [
34                 {
35                     "variable_uid": "f02e9005-ad1d-4579-8130-9580566e3b64",
36                     "intermediate_calc": false,
37                     "operation_id": 1,
38                     "order_number": 1,
39                     "is_unit_conv": false,
40                     "is_gwp": false,
41                     "variable_calc": []
42                 }
43             ]
44         },
45         ...
46     ]

```

6.3. Operations

Entity operations define different types of calculation functions and have the following attributes:

- Internal identifier a numeric index for internal reference (ID)
- type - defines the type of operation. For example:
 - *AGG* aggregation or sum between two or more variables
 - *DIFF* difference between variables
 - *IEF* calculation of implied emission factor where emissions are divided by activity data
 - *MLP* multiplication operation between variables
 - *PERCENTAGE* The formula is "100 * value"
 - *PERCENTAGE_DIFF* The formula is "100 - value"
 - *ABS* absolute value
 - *AGG_COMMENT* used for aggregating comments and documentation boxes
 - *SUM_AC* For CRT only: Used in the intermediate calculation for the IEF (Implied Emission Factor):
 - i. If the emissionvariable is set to a NK then the intermediate calculation will return the emissionvariable no matter what in amountCapturedVariable
 - ii. The IEF will then be set to the same NK
 - *COPY_IF_NO* in case values of AD is set to "NO" then Emissions automatically are set to "NO"
 - *COPY_IF_NK* used to set the Amount Captured to same NK set in the Emissions
 - *AGG_TEXT* aggregate TEXT_ONLY variables
 - *DELETE_IF_NOT_FILLED_WITH_TT* used by the application
 - *DELETE_IF_NOT_FILLED_WITH_CB* used by the application
 - *DELETE_IF_NOT_FILLED_WITH_YES* used by the application

- *COPY_IF_EMPTY* used by the application
- *AGG_LINE_DESC* Used to return the line description of the variable passed
- *AGG_TEXT_DASH* Used for concatenation of all text values, with dash as a separator.
- *REF_PERIOD_AVG* used for the calculation in the reporting tables trend tables to calculate the average of the values reported in the reference period specified
- *BASE_YEAR_AVG* used for the calculation in the reporting tables trend tables to calculate the average of the values reported in the base year
- *TREND* used for the calculations in the reporting table to calculate the trend change in the trend tables
- *AGG_TEXT_SPACE* Used for concatenation of all text values, with space as a separator
- *KCA_LEVEL* used to calculate the KCA level
- *KCA_TREND* used to calculate the KCA trend
- *DELETE_IF_DIFFERENT* used by application
- *DELETE_IF_NU* used by application
- *AGG_TEXT_COMMA* Used for concatenation of all text values, with comma as a separator
- *KCA_TREND_PERCENTAGE* used for calculating the trend percentage
- *AGG_TEXT_IF_FILLED* used by application
- *RETURN_IF_FILLED* used by application
- *GET_VALUE_FROM_PREV_SUBMISSION* used to get a value from the previous submission
- *FIRST_NOT_NULL* used by application

Example of operation definitions

```

1      "operation": [
2          {
3              "id": 1,
4              "type": "AGG"
5          },
6          {
7              "id": 2,
8              "type": "DIFF"
9          },
10         {
11             "id": 3,
12             "type": "DIV"
13         },
14         {
15             "id": 4,
16             "type": "MLP"
17         },
18         {
19             "id": 5,
20             "type": "IEF"
21         },
22         ...
23     ]

```

6.4. Variable validations

It is important that each value is tested against the validation rule to ensure consistency and make sure that the text is not entered where the number is expected. Metadata defines validation rules with the name, and each variable must have a validation rule. In the case of the *emissions* validation rule is a range in the number; it can be only positive and not zero. For *removals*, the range is different; it can be only negative and not zero. For most of the variables, ranges are the same and defined as constants; for example positive range with zero

has the name "POSITIVE_WITH_ZERO_RANGE", for text only, it has the name "TEXT_ONLY" and so on. All these names are stored in the "dimension_instance" entity as part of the "dimension" entity called "**VALIDATION_RULE**".

7. Navigation and Data Entry

For the visual representation of data entry, the following entities are used:

- Node - Tree that is used for visual representation and navigation by the user
- Grid - defining rows of the grid. By default, columns present inventory years

7.1. Node

Node is a tree that is used for visual representation and navigation by the user to open the data entry grid.



It has the following attributes (see example below):

- id - unique internal identifier
- uid - global unique identifier
- name_prefix - prefix of the node, usually presented as a category code
- name - name of the node, shown to the user

- type - is a type of the node; see a screenshot from the meta-tool

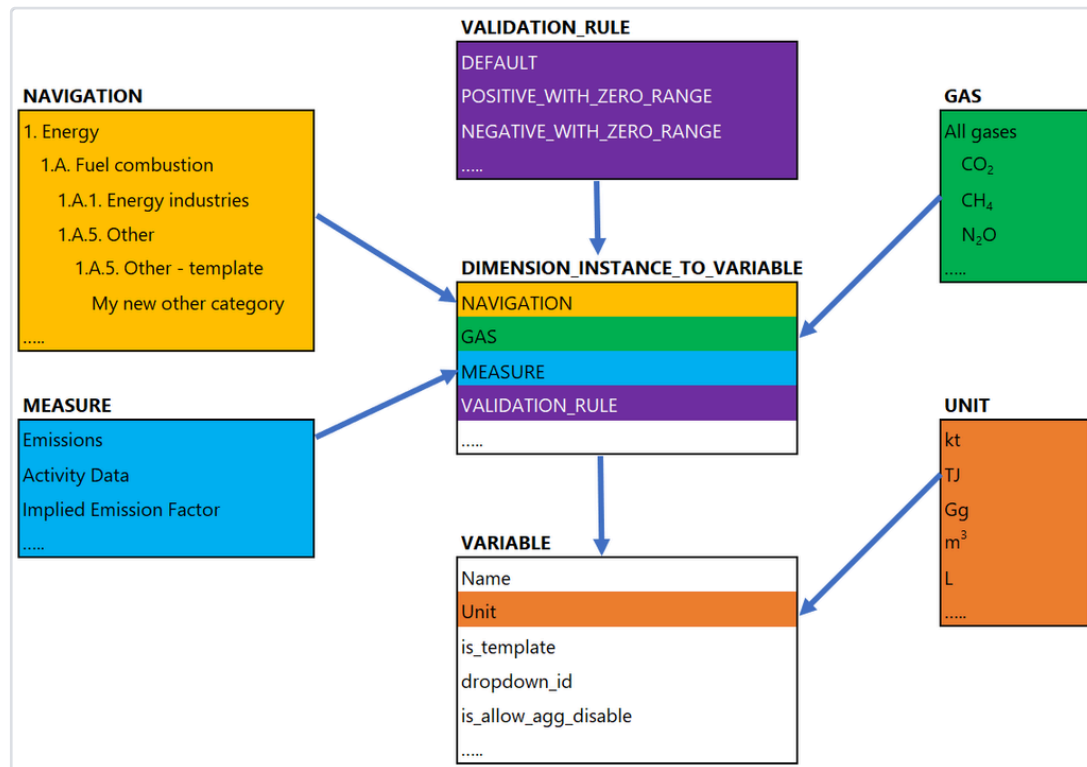
Node	Type	Customization	Dimension to change
Sectors/Totals	FIXED		
1. Energy	FIXED		
1.AA Fuel Combustion - Sectoral approach	FIXED		
1.A.1 Energy Industries	FIXED		
1.A.2 Manufacturing Industries and Construction	FIXED		
1.A.3 Transport	FIXED		
1.A.3.a Domestic Aviation	FIXED		
1.A.3.b Road Transportation	FIXED		
1.A.3.b.i Cars	FIXED		
1.A.3.b.ii Light duty trucks	FIXED		
1.A.3.b.iii Heavy duty trucks and buses	FIXED		
1.A.3.b.iv Motorcycles	FIXED		
1.A.3.b.v Other (please specify)	LIST	EDITABLE	
TEMPLATE	LIST_ITEM	TEMPLATE	CATEGORY
Gasoline	FIXED_ITEM	TEMPLATE	
Diesel Oil	FIXED_ITEM	TEMPLATE	
Liquefied Petroleum Gases (LPG)	FIXED_ITEM	TEMPLATE	
Other Liquid Fuels (please specify)	FIXED_LIST_ITEM	TEMPLATE_EDITABLE	
TEMPLATE	ITEM	TEMPLATE	CLASSIFICATION
Gaseous Fuels	FIXED_ITEM	TEMPLATE	
Biomass	FIXED_ITEM	TEMPLATE	
Other Fossil Fuels (please specify)	FIXED_LIST_ITEM	TEMPLATE_EDITABLE	
TEMPLATE	ITEM	TEMPLATE	CLASSIFICATION
Not occurring [IPCC Software]	LIST_ITEM	CS_COMMON	
Gasoline	FIXED_ITEM	CS_COMMON	
Diesel Oil	FIXED_ITEM	CS_COMMON	
Liquefied Petroleum Gases (LPG)	FIXED_ITEM	CS_COMMON	
Other Liquid Fuels (please specify)	FIXED_LIST_ITEM	CS_COMMON	
Not occurring [IPCC Software]	ITEM	CS_COMMON	
Gaseous Fuels	FIXED_ITEM	CS_COMMON	
Biomass	FIXED_ITEM	CS_COMMON	
Other Fossil Fuels (please specify)	FIXED_LIST_ITEM	CS_COMMON	
Not occurring [IPCC Software]	ITEM	CS_COMMON	
1.A.3.c Railways	FIXED		
1.A.3.d Domestic Navigation	FIXED		

Type	Visibility	UI actions	Combinations	
			drop-down (type 2)	free text (type 1)
FIXED	always visible	no action	no combinations	no combinations
LIST	always visible	show button for customization	CUSTOMIZATION is set to null	CUSTOMIZATION is not null
LIST_ITEM	if the user selected me from the drop-down	show button for customization	CUSTOMIZATION is set to null	CUSTOMIZATION is not null
FIXED_LIST_ITEM	visible if the parent is visible	show button for customization	CUSTOMIZATION is set to null	CUSTOMIZATION is not null
ITEM	if the user selected me from the drop-down	no action	CUSTOMIZATION is set to null	CUSTOMIZATION is not null
FIXED_ITEM	visible if the parent is visible	no action	CUSTOMIZATION is set to null	CUSTOMIZATION is not null
HIDDEN	always hidden (used only by the application)	no action	no combinations	no combinations

- customization_type
 - CS_COMMON - a country-specific node that can be used across all countries and customizable based on the template. The only difference compared to the free text (type 1) customization is that all variables and nodes are pre-created in the metadata. The

CS_COMMON customization type is mainly used by the IPCC for mapping standard IPCC categories to the country-specific nodes in the CRT.

- EDITABLE - node that is editable when there are no country-specific child nodes added;
 - TEMPLATE - node used as a template for customization of free text **type 1 customization** only.
 - TEMPLATE_EDITABLE - node that is editable when there are no country-specific child nodes added and used as a template for customization of free text **type 1 customization** only (used by application)
 - AUTO_GENERATED - auto-generated prefix for child nodes (Type 1 customization) using the special codes (example of codes: "#.", "@.#"), used in NDC to auto-generate prefixes for indicators (used by application)
 - AUTO_GENERATED_FROM_LIST - auto-generate prefixes for the child nodes (Type 2 customization) using the codes (example of codes: "#.", "@.#"), used in NDC to autogenerate prefixes for targets (used by application)
 - SUBMISSION_SETTINGS - when the value is set the node visibility and its child nodes depend on version settings (used by the application)
 - SUBMISSION_SETTINGS_ITEM - when the value is set the node visibility depends on version settings (used by the application)
 - SINGLE_SELECT - is used for nodes with Type 2 customization to allow to selection of only one item (used by application)
- dimension_to_change - If a node is customizable, it specifies which dimension will be used to change the variable. The free text type 1 customization should be added under the parent of the template dimension instance. See the figure below, where the parent dimension of CATEGORY is "1.A.5 Other - template" and the child dimension instance is "My new other category"



- order_number - order of the nodes in the tree
- grid_name - name of the grid
- template_node_id - id of the template node attached
- template_node_uid - a reference to the global unique identifier of the template node
- grid_type is a type of visual representation of the grid that can have the following values:
 - timeseries - a time series with set of years as columns
 - form - a form without any year

- `extended_timeseries` - same as time series, but includes extra columns for additional years
- `ghg_timeseries` - used in NDC for presenting Table 6
- `projection` - used in NDC for projection table
- `projection_with_unit` - used in NDC for projection table with an extra column for the unit of measurement
- `kca_grid` - used for Key Category Assessment visual presentation
- `documentation` is a flag when set to true it has a documentation box attached (used for NDC)
- `linked_entities` is an array with references to the dependent entities with a list of global unique identifiers and the type of the entity (for example when the node should be added to the other nodes in this case the type is set to "node" or line within the grid then the type is set to "group")
- `cross_reference_variable_uid` is an array of variable UIDs that are used as a cross-reference within several nodes (for example: population variable in the agriculture sector)
- `intermediate_cals_variables` is an array of variable UIDs used for intermediate calculations (used only by the application)
- `roles` is an array of role names (used by application only)
- `node` - an array of other child nodes

Example: Node section

```

1      {
2        "id": 225,
3        "uid": "711ab9da-13cd-44d8-b8f4-33a954171186",
4        "name_prefix": "",
5        "name": "Sectors/Totals",
6        "type": "FIXED",
7        "customization_type": "",
8        "dimension_to_change": null,
9        "order_number": 1000000000000000000,
10       "grid_name": "Grid_10000000000000000000",
11       "template_node_id": null,
12       "template_node_uid": null,
13       "grid_type": "timeseries",
14       "documentation": false,
15       "linked_entities": [],
16       "cross_reference_variable_uid": [],
17       "intermediate_calc_variables": [],
18       "roles": [
19         "Expert_Agriculture",
20         "Expert_Energy",
21         "Expert_IPPU",
22         "Expert_LULUCF",
23         "Expert_Waste",
24         "Inventory_FP",
25         "Inventory_FP",
26         "Inventory_FP_alternate",
27         "Inventory_FP_alternate"
28       ],
29       "node": [
30         {
31           "id": 162,
32           "uid": "3665c27e-d055-47d7-8393-5f934f3ced9d",
33           "name_prefix": "1.",
34           "name": "Energy",
35           "type": "FIXED",
36           "customization_type": "",
37           "dimension_to_change": null,
38           "order_number": 1010000000000000000,

```

```

39     "grid_name": "Grid_101000000000000000",
40     "template_node_id": null,
41     "template_node_uid": null,
42     "grid_type": "timeseries",
43     "documentation": false,
44     "linked_entities": [],
45     "cross_reference_variable_uid": [],
46     "intermediate_calc_variables": [],
47     "roles": [
48         "Expert_Energy",
49         "Inventory_FP",
50         "Inventory_FP_alternate"
51     ],
52     "node": [
53         {
54             "id": 287,
55             "uid": "82239b33-699b-4187-b992-4a0883d99952",
56             "name_prefix": "1.A.",
57             "name": "Fuel combustion activities (sectoral approach)",
58             "type": "FIXED",
59             "customization_type": "",
60             "dimension_to_change": null,
61             "order_number": 101010000000000000,
62             "grid_name": "Grid_101010000000000000",
63             "template_node_id": null,
64             "template_node_uid": null,
65             "grid_type": "timeseries",
66             "documentation": false,
67             "linked_entities": [],
68             "cross_reference_variable_uid": [],
69             "intermediate_calc_variables": [],
70             "roles": [
71                 "Expert_Energy",
72                 "Inventory_FP",
73                 "Inventory_FP_alternate"
74             ],
75             "node": [
76                 {
77                     "id": 154,
78                     "uid": "f85540ba-43dc-4724-8d4a-f8673fdc9e0c",
79                     "name_prefix": "1.A.1.",
80                     "name": "Energy industries",
81                     "type": "FIXED",
82                     "customization_type": "",
83                     "dimension_to_change": null,
84                     "order_number": 101010100000000000,
85                     "grid_name": "Grid_101010100000000000",
86                     "template_node_id": null,
87                     "template_node_uid": null,
88                     "grid_type": "timeseries",
89                     "documentation": false,
90                     "linked_entities": [],
91                     "cross_reference_variable_uid": [],
92                     "intermediate_calc_variables": [],
93                     "roles": [
94                         "Expert_Energy",
95                         "Inventory_FP",
96                         "Inventory_FP_alternate"

```

```

97         ],
98         "node": [
99             {
100                 "id": 147,
101                 "uid": "5bab6e8b-633c-45ba-9206-82455177330a",
102                 "name_prefix": "1.A.1.a.",
103                 "name": "Public electricity and heat production",
104                 "type": "LIST",
105                 "customization_type": "",
106                 "dimension_to_change": null,
107                 "order_number": 101010101000000000,
108                 "grid_name": "Grid_101010101000000000",
109                 "template_node_id": null,
110                 "template_node_uid": null,
111                 "grid_type": "timeseries",
112                 "documentation": false,
113                 "linked_entities": [],
114                 "cross_reference_variable_uid": [],
115                 "intermediate_calc_variables": [],
116                 "roles": [
117                     "Expert_Energy",
118                     "Inventory_FP",
119                     "Inventory_FP_alterate"
120                 ],
121                 "node": [
122                     {
123                         "id": 288,
124                         "uid": "a2b5241f-b208-4f2c-82cf-3d01c78eb983",
125                         "name_prefix": "1.A.1.a.i.",
126                         "name": "Electricity generation",
127                         "type": "ITEM",
128                         "customization_type": "",
129                         "dimension_to_change": null,
130                         "order_number": 101010101000000000,
131                         "grid_name": "Grid_101010101000000000",
132                         "template_node_id": null,
133                         "template_node_uid": null,
134                         "grid_type": "timeseries",
135                         "documentation": false,
136                         "linked_entities": [],
137                         "cross_reference_variable_uid": [],
138                         "intermediate_calc_variables": [],
139                         "roles": [
140                             "Expert_Energy",
141                             "Inventory_FP",
142                             "Inventory_FP_alterate"
143                         ],
144                     }
145                 ]

```

7.1.1. Customization of nodes and country-specific data

In some of the nodes, it is possible to add a new child node (customize) with detailed information. In some cases, the country has additional information, and there are specific nodes where it is possible. There are different types of customization possible for different dimensions.

The node under which the customization is available has an attribute about the **type** of customization, the dimension that is used for customization (**dimension_to_change**), and the **dimension instance ID** where children nodes are added.

The following **types** are identified:

- Type 1: Template: The user can add a new node and enter the name. In this case, the node should be added as a child.
- Type 2: Fixed list: The user can select a node to add from a predefined list, e.g., animal type, gases, and so on. In this case, the list of available nodes are children of dimension instances where customization is performed.

In the case of the type template node, the set of variables should be the same as in the parent node, but the dimension to change will have a new dimension instance, and new variables need to be created. Those variables need to be flagged as country-specific and linked to the variables from the parent node so that vertical aggregation rules are applied, and horizontal calculation is also taken into account and inherited from the variables in the parent node.

Examples of node customization scenarios

▼ Examples of node customization scenarios

Scenario 1 (type 2 customization)

Name		Type	Custo mizati on	Dimens ion to change	Action
node 1		LIST	null	null	show the button to present the drop-down list
	node 1.1	ITEM	null	null	if I am selected from the drop-down, make me visible
	node 1.1.1	FIXED _ITEM	null	null	if my parent is selected, make me visible
	node 1.1.2	FIXED _ITEM	null	null	if my parent is selected, make me visible

Scenario 2 (type 2 customization with type 1 customization)

Name		Type	Custo mizati on	Dimens ion to change	Action
node 1		LIST	null	null	show the button to present the drop-down list
	node 1.1	LIST_I TEM	null	null	if I am selected from the drop-down, make me visible and show the button to present the drop-down list
	node 1.1.1	ITEM	null	null	if I am selected from the drop-down, make me visible
	node 1.1.2	ITEM	null	null	if I am selected from the drop-down, make me visible
	node 1.1.3	LIST_I TEM	EDITA BLE	null	if I am selected from the drop-down, make me visible and show the button to enter free text
	node 1.1.3.1 - template	ITEM	TEMP LATE	CLASSI FICATI ON	do NOT show me, but instead use me as a template for creating children
	node 1.1.3.1 (free text)	ITEM	null	null	create me as a child of the parent, copy me from the template, create variables and grids from the template

Scenario 3 (type 2 customization with type 2 customization)

Name		Type	Custo mizati	Dimens ion to	Action
------	--	------	-----------------	------------------	--------

				on	change	
node 1				LIST	null	show the button to present the drop-down list
	node 1.1			ITEM	null	if I am selected from the drop-down, make me visible
		node 1.1.1		FIXED _ITEM	null	if my parent is selected, make me visible
		node 1.1.2		FIXED _ITEM	null	if my parent is selected, make me visible
		node 1.1.3		FIXED _LIST_ ITEM	null	if my parent is selected, make me visible and show the button to present the drop-down list
			node 1.1.3.1	ITEM	null	if I am selected from the drop-down, make me visible

Scenario 4 (type 2 customization with type 1 customization)

Name				Type	Custo mizati on	Dimens ion to change	Action
node 1				LIST	null	null	show the button to present the drop-down list
	node 1.1			ITEM	null	null	if I am selected from the drop-down, make me visible
		node 1.1.1		FIXED _ITEM	null	null	if my parent is selected, make me visible
		node 1.1.2		FIXED _ITEM	null	null	if my parent is selected, make me visible
		node 1.1.3		FIXED _LIST_ ITEM	EDITA BLE	null	if my parent is selected, make me visible and show the button to enter free text
			node 1.1.3.1 - template	ITEM	TEMP LATE	CLASSI FICATI ON	do NOT show me, but instead use me as a template for creating children
			node 1.1.3.1 (free text)	ITEM	null	null	create me as a child of the parent, copy me from the template, create variables and grids from the template

Scenario 5 (type 1 customization with type 1 customization)

Name				Type	Custo mizati on	Dimens ion to change	Action
node 1				LIST	EDITA BLE	null	show the button to enter free text
	node 1.1 - template			ITEM	TEMP LATE	CATEG ORY	do NOT show me, but instead use me as a template for creating children

		node 1.1.1	FIXED_ITEM	TEMP LATE	null	when you add my parent, also create me
		node 1.1.2	FIXED_LIST_ITEM	TEMP LATE_EDITABLE	null	when you add my parent, also create me and show the button to enter free text
		node 1.1.2.1 - template	ITEM	TEMP LATE	CLASSIFICATION	do NOT show me, but instead use me as a template for creating children
	node 1.1 (free text)		ITEM	null	null	create me as a child of the parent, copy me from the template, create variables and grids from the template
		node 1.1.1	FIXED_ITEM	null	null	create me as a child of the parent, copy me from the template, create variables and grids from the template
		node 1.1.2	FIXED_LIST_ITEM	null	null	create me as a child of the parent, copy me from the template, create variables and grids from the template
		node 1.1.2.1 (free text)	ITEM	null	null	create me as a child of the parent, copy me from the template, create variables and grids from the template

7.2. Grid

The entity grid specifies the list of rows in the grid, and it has a tree structure (grouping) e.g., see the example of the node "1.A.1.a.i Electricity Generation"

Sectors/Totals > 1. Energy > 1.A. Fuel combustion activities (sectoral approach) > 1.A.1. Energy industries > 1.A.1.a. Public electricity and heat production > 1.A.1.a.i. Electricity generation

1.A.1.a.i. Electricity generation

Expand all Show/hide years Export

ID	Description	Unit	1990	1991	1992	1993	1994	1995	1996	1997
01	▼ Fuel consumption	TJ	NA	NA	NA	NA	NA	NA	NA	NA
02	└ Liquid fuels	TJ	NA	NA	NA	NA	NA	NA	NA	NA
03	└ Solid fuels	TJ								
04	└ Gaseous fuels	TJ								
05	└ Other fossil fuels	TJ								
06	└ Peat	TJ								
07	└ Biomass	TJ								
08	▼ Calorific value		GCV/NA	GCV/NA	GCV/NA	GCV/NA	GCV/NA	GCV/NA	GCV/NA	GCV/NA
09	└ Liquid fuels		NA	NA	NA	NA	NA	NA	NA	NA
10	└ Solid fuels		GCV	GCV	GCV	GCV	GCV	GCV	GCV	GCV
11	└ Gaseous fuels		GCV	GCV	GCV	GCV	GCV	GCV	GCV	GCV

The grid has the following attributes:

- node_id - reference to the node with internal identifier
- node_uid - reference to the node with a global unique identifier
- tooltip_uid - reference to dimension instance with specific dimension "TOOLTIP"
- footnotes - footnotes of the node (appears next to the header of a node) - array with references to the dependent entities with a list of the internal identifiers of dimension instance with specific dimension "FOOTNOTE" and order of appearance (used by the application)

- additional_columns - an array of additional column definitions, have the following attributes:
 - order_number - display order between columns with dependency from group, for example, if equal "-1" then the column appears before the group, if equal "1" then after the group
 - dimension_instance_id - reference to the dimension instance with an internal identifier with specific dimension "GRID_LINE_NAME" to define column header
 - dimension_instance_uid - reference to the dimension instance with a global unique identifier with specific dimension "GRID_LINE_NAME" to define column header
 - formula - expression to calculate the value
 - mappings - array of rows:
 - order_number - rows order which should be consistent with order of groups
 - variable_uid - reference to the variable with global unique identifier
 - dimension_instance_uid - reference to the dimension instance with global unique identifier with specific dimension "GRID_LINE_NAME"
- group - set of rows (lines)

The group has the following attributes:

- id - [mandatory] internal identifier of the group
- uid - [mandatory] universal/global unique identifier of the group (UUID/GUID)
- order_number - [mandatory] display order between direct children
- variable_uid - [mandatory] reference to the variable universal/global unique identifier (UUID/GUID)
- dimension_instance_id - [mandatory] reference to the dimension instance with an internal identifier, the combination of "name" of the dimension instance is used for the label of the row
- dimension_instance_uid - [mandatory] reference to the dimension instance with global unique identifier, the combination of "name" of the dimension instance is used for the label of the row
- line_type - [mandatory] type of the line e.g., read-only, editable, or calculated
- selectable_variables - array of universal/global unique identifier of the variables (obsolete)
- dropdown_id - [mandatory] reference to the drop-down identifier within the metadata file, used to show drop-down selection for the group (line)
- type - [mandatory] used for the customization of the data entry grid group (line)

Type	Visibility	UI actions	Combinations	
			drop down (type 2)	free text (type 1)
FIXED	always visible	no action	no combinations	no combinations
LIST	always visible	show button for customization	CUSTOMIZATION is set to null	CUSTOMIZATION is not null
LIST_ITEM	if the user selects me from the drop-down	show button for customization	CUSTOMIZATION is set to null	CUSTOMIZATION is not null
FIXED_LIST_ITEM	visible if the parent is visible	show button for customization	CUSTOMIZATION is set to null	CUSTOMIZATION is not null
ITEM	if the user selects me from the drop-down	no action	CUSTOMIZATION is set to null	CUSTOMIZATION is not null
FIXED_ITEM	visible if the parent is visible	no action	CUSTOMIZATION is set to null	CUSTOMIZATION is not null

- customization_type - [mandatory] used for customization of the data entry grid group (line)
 - CS_COMMON - country-specific group (line) that can be used across all countries and customizable based on the template. The only difference compared to the free text (type 1) customization is that variable and related dimension instances are pre-created in the metadata.
 - EDITABLE - group (line) that is editable when there is no country-specific child group (line) added;
 - TEMPLATE - group (line) used as a template for customization of free text **type 1 customization** only.
 - AUTO_GENERATED - auto generate group description with child nodes (Type 1 customization) using the codes (example of codes: "#.", "@.#"), used in NDC for auto generation (used by application)
 - SINGLE_SELECT - is used for groups with Type 2 customization allow to select only one item (used by application)
- template_group_uid - [mandatory] reference to the template group (line)
- tooltip_uid - reference to dimension instance with specific dimension "TOOLTIP"
- group_description_calc - is used to concatenate a group description from different sources, array of entities has the following attributes:
 - entity_type - type of the entity e.g., variable, node, dimension instance
 - entity_property - name of an attribute from entity
 - entity_uid - global unique identifier of an entity
 - order_number - order of concatenation
- footnotes - group footnotes (appears next to the header of group (grid line)) - array with references to the entities with list of internal identifier of dimension instance with specific dimension "FOOTNOTE" and order of appearance (used by the application)
- group - [mandatory] set of rows within the group, basically a reference to the group entity (itself); if a not nested group is defined, then it is set to *null*

Example: Grid section

```

1  "grid": [
2      {
3          "node_id": 225,
4          "node_uid": "711ab9da-13cd-44d8-b8f4-33a954171186",
5          "tooltip_uid": null,
6          "footnotes": [
7              {
8                  "order_number": 6,
9                  "dimension_instance_id": 37134
10             }
11         ],
12         "additional_columns": [],
13         "group": [
14             {
15                 "id": 93282,
16                 "uid": "a5e962f3-223e-4f59-8c3b-bee76b5907e3",
17                 "order_number": 1,
18                 "variable_uid": null,
19                 "dimension_instance_id": 36093,
20                 "dimension_instance_uid": "a24c28e8-d9d4-481c-978e-2b1e941c2aa2",
21                 "line_type": "READONLY",
22                 "selectable_variables": [],
23                 "dropdown_id": null,
24                 "type": "FIXED",
25                 "customization_type": null,
26                 "template_group_uid": null,
27                 "tooltip_uid": null,
28                 "group_description_calc": [],
29                 "footnotes": [],
30                 "group": [
31                     {

```

```

32         "id": 93275,
33         "uid": "33375a20-bcda-46e4-90fa-b8fead9fdd1a",
34         "order_number": 2,
35         "variable_uid": "52ab74bd-0a97-4a23-a2fc-b1c63e884b7d",
36         "dimension_instance_id": 36099,
37         "dimension_instance_uid": "fae75146-c49f-4b63-a386-861744a224c2",
38         "line_type": "CALCULATED",
39         "selectable_variables": [],
40         "dropdown_id": null,
41         "type": "FIXED",
42         "customization_type": null,
43         "template_group_uid": null,
44         "tooltip_uid": null,
45         "group_description_calc": [],
46         "footnotes": [],
47         "group": []
48     },
49     {
50         "id": 93290,
51         "uid": "78931dbf-8171-45be-b414-8139ebcc046a",
52         "order_number": 3,
53         "variable_uid": "f05d8ef0-cd04-4769-81be-303183271230",
54         "dimension_instance_id": 36081,
55         "dimension_instance_uid": "2f0a40e7-383f-42e1-aa4c-969dd86c3f74",
56         "line_type": "CALCULATED",
57         "selectable_variables": [],
58         "dropdown_id": null,
59         "type": "FIXED",
60         "customization_type": null,
61         "template_group_uid": null,
62         "tooltip_uid": null,
63         "group_description_calc": [],
64         "footnotes": [],
65         "group": []
66     },
67     ...
68 ]

```

7.2.1. Customization of grids

Currently, there are two types of customization defined within the data entry grid:

- Type 1: When the user has the possibility to add a new line (country-specific). In this case, the "type" attribute has "TEMPLATE" and variable_uid has a reference to the template variable. The new variable and line of the grid should be saved in the country-specific section of the data exchange JSON;
- Type 2: When the user selects a new group (line) from the pre-defined list of available items, the same approach as for the type 2 node customization is applicable.

7.2.2. Visual appearance

Color codes are driven by the type of the line: an empty cell is gray, read-only (cross-referenced from another data entry grid) is blue, the calculated cell is green, and the editable cell is white; other types are not visible to the user.

8. Reporting tables

Metadata includes instructions to generate reporting tables. Reporting tables are Excel-based files with the set of tables used for presenting data in the format agreed by Parties. As the reporting tables have pre-defined styles and order of the tables in the file, metadata assumes that the Excel template file is used as a basis, and metadata includes instructions on how to populate the empty template with the actual data.

The structure of the reporting tables has a tree structure of "reporting table" entity with the following attributes:

- id - [mandatory] internal unique identifier with the integer type
- uid - [mandatory] universal (global) unique identifier (UUID or GUID)
- name - [mandatory] name of the table (worksheet name)
- order_number - [mandatory] order of the appearance
- abs_row_start - [optional] absolute index of the start row in the worksheet
- abs_col_start - [optional] absolute index of the start column in the worksheet
- abs_row_end - [optional] absolute index of the last row in the worksheet
- abs_col_end - [optional] absolute index of the last column in the worksheet
- ranges - [optional] array of the range in the worksheet to be populated with the data and has the following attributes:
 - name - [mandatory] human-readable name of the range is unique within the reposting table ranges
 - row_start - [mandatory] index of the start row in the range
 - col_start - [mandatory] index of the start column in the range
 - row_end - [mandatory] index of the last row in the range
 - col_end - [mandatory] index of the last column in the range
 - order - [mandatory] processing order of the range
 - type - [mandatory] type of the range can have the following values:
 - MAIN - main range of the table in the worksheet
 - COUNTRY - static range of the table to display country name
 - SUBMISSION - static range of the table to display submission name
 - YEAR - static range of the table to display inventory year
 - HISTORICAL_YEAR - static range of the table to display historical year (used for NDC)
 - DOC_BOX - documentation box range. All variables in the documentation box need to be concatenated and populated within the range.
 - DYNAMIC - dynamic range, range can be changed due to the country-specific information
 - CUSTOM_FOOTNOTES - dynamic range is used to display custom footnotes
 - KCA_KEY_CATEGORY - dynamic range for CRT Table 7
- dyn_type - [mandatory] used only if the type of the range is DYNAMIC. It specifies the operation to be performed, and it can have the following settings:
 - REMOVE_ROW - remove row
 - REMOVE_ROW_MULTI and REMOVE_ROW_ITEM- remove row from REMOVE_ROW_MULTI range if all child ranges with dyn_type REMOVE_ROW_ITEM are removed (used only by application)
 - REMOVE_COLUMN - remove column
 - ADD_ROW - add row
 - ADD_EXTENDED_TIMESERIES_ROW - add rows for each year in the timeseries with additional years
 - ADD_ROW_YEAR - add row for each available year in the defined node entity
 - ADD_COLUMN - add column
 - ADD_TIMESERIES_COLUMN - add columns for each year in the timeseries
 - FOOTNOTE_1 - set footnote 1 to the cell (used for FTC)
 - FOOTNOTE_2 - add footnote 2 of the defined node entity (used for FTC)

- DOC_BOX - set documentation box value to the cell from the defined node entity with "documentation" set to TRUE
- NK_EXPLANATION_IE - set explanation text of IE notation key (CRT Table 9) for defined DIMENSION_INSTANCE_{id} entity
- NK_EXPLANATION_NE - set explanation text of NE notation key (CRT Table 9) for defined DIMENSION_INSTANCE_{id} entity
- FILL_FROM_NODE - set variable value from defined node entity
- DELETE_IF_NOT_YES - remove row if the value for defined variable entity is not "Yes"
- DELETE_IF_NOT_CB - remove row if the value for defined variable entity is not "Capacity-building"
- DELETE_IF_NOT_TT - remove row if the value for defined variable entity is not "Technology development and transfer"
- DELETE_IF_HIDDEN - remove range if defined grid_line entity is hidden
- DELETE_IF_NO_CHILD - set NA to the cell if the defined node entity has no children
- entity - [mandatory] used only if the type of the range is DYNAMIC. It specifies the entity used for the customization, and it could be the following entities:
 - NODE - country-specific node
 - GRID_LINE - country-specific grid line
 - VARIABLE - variable value
 - SUBMISSION_SETTING - value from version settings variable
 - PROJECTION_YEAR - value for projection year variable
 - DIMENSION_INSTANCE_{id} - dimension instance with internal unique identifier
- entity_uid - [mandatory] used only if the type of the range is DYNAMIC. It specifies the unique identifier of the entity attribute.
- cell_mappings - [mandatory] an array of cell mapping entity. It specifies the mappings of variables to cells, and it has the following attributes:
 - variable_uid - [mandatory] universal (global) unique identifier of the entity which is defined in the "type" attribute
 - row - [mandatory] cell index of the row
 - col - [mandatory] cell index of the column
 - formula - [optional] formula used to populate the value from the variable
 - is_template - [optional] specifies if the variable is used as a template
 - type - [optional] defines the type of the cell has the following values:
 - link
 - variable
 - group
 - submission_settings
 - DIMENSION
 - node
 - FORMULA_TEXT
 - COMMENT
- child_ranges - [optional] child range entity

Example: Reporting tables

```

1 {
2     "reporting_tables_tree": [
3         {
4             "id": 2649,
5             "uid": "36fc856c-562a-4688-a18c-348f1db46f19",
6             "name": "Energy",
7             "order_number": 1000000000000000000,
8             "reporting_table": [
9                 {
10                    "id": 2650,
11                    "uid": "6ea4056f-a35b-4faa-8057-cd1f84aa0b07",

```



```

12     "name": "Sectoral summary tables",
13     "order_number": 101000000000000000,
14     "reporting_table": [
15         {
16             "id": 2651,
17             "uid": "f26a3ce7-084b-4105-b978-5ff07cf738e7",
18             "name": "Table1",
19             "order_number": 101010000000000000,
20             "abs_row_start": 8,
21             "abs_col_start": 2,
22             "abs_row_end": 59,
23             "abs_col_end": 10,
24             "ranges": [
25                 {
26                     "name": "Table1_MAIN",
27                     "row_start": 8,
28                     "col_start": 2,
29                     "row_end": 59,
30                     "col_end": 10,
31                     "order": 1,
32                     "type": "MAIN",
33                     "dyn_type": "",
34                     "entity": "",
35                     "entity_uid": "",
36                     "cell_mappings": [
37                         {
38                             "variable_uid": "8EF3E9A8-8A8A-48A9-AC25-C1929B7C0983",
39                             "row": 10,
40                             "col": 3,
41                             "formula": "",
42                             "is_template": false,
43                             "type": ""
44                         },
45                         {
46                             "variable_uid": "27E7168E-9B90-4EC9-BDA9-7B02224EABD5",
47                             "row": 10,
48                             "col": 4,
49                             "formula": "",
50                             "is_template": false,
51                             "type": ""
52                         },
53                         {
54                             "variable_uid": "F58F59A4-B902-4FD6-A8E6-448EAC9A8B62",
55                             "row": 10,
56                             "col": 5,
57                             "formula": "",
58                             "is_template": false,
59                             "type": ""
60                         },
61                         {
62                             "variable_uid": "6556B45A-14C8-43BA-BE76-EF78CC7818D3",
63                             "row": 10,
64                             "col": 6,
65                             "formula": "",
66                             "is_template": false,
67                             "type": ""
68                         },
69                     ]

```

```

70         ],
71         "child_ranges": null
72     }
73     ],
74     "reporting_table": null
75 }
76 ]
77 },
78 ....
79 ]
80 }
81 ]
82 }

```

Important

- In the case of child ranges, cell_mappings for the child ranges are not included in the parent, so they do not cross with each other
- Processing order is important when processing the ranges and when adding or removing rows/columns/cells. It doesn't impact already processed ranges.

9. Version settings

The section "settings" represents instructions for the version settings. Version settings are used for simplifying operations for the end user. Basically, it automates filling in the data within the data entry grids for the version. Currently, the following functions are defined:

- visibility condition - when the user hides/unhides parts of the data
- create country-specific definitions and data
- flexibility conditions - when the user defines pre-fill data with the "FX" notation key, disables aggregation, and provides aggregate value

There is a set of settings for the application. Each value in the settings is stored in the related variable and referenced by a unique identifier (UID). Some values trigger specific actions or operations in the application. Definitions in the metadata can give instructions to the application on what are the values in the settings and what these values can trigger in terms of the operations. The following structure is defined to describe this process.

We have an array of settings, and each setting has the following attributes:

- uid - [mandatory] unique identifier of the setting
- name - [mandatory] human-readable name of the setting
- order_number - [optional] order of the setting
- variable_uid - [mandatory] unique identifier of the variable, which will be used to store the value
- triggered_value - [optional] an array of values that may trigger an operation
- operation - [optional] an array of operation entity, which is triggered by a specific value. See attributes below:
 - operation - [mandatory] a type of operation or function which is called by the application, that could be visibility, flexibility
 - value - [mandatory] one of the values or array of values from the triggered_value attribute of the setting. The value is used as a condition to trigger an operation.
 - affected_entity - [mandatory] an array of the following structure:
 - uid - [mandatory] array of UIDs of the entity that are affected by the operation
 - type - [mandatory] type of the entity, could be node, variable, or grid line
 - value_to_set - [optional] value to be set to entity
- roles - [mandatory] is an array attribute to specify list of roles has ability to change the version setting. Used by application only

If the settings do not trigger any operation and only store the value entered by the user, then triggered_value and operation are optional in this case. An example could be the text entered by the user should appear in the reporting table, so in this case, stored value without any

operation will fulfill the requirement.

In more complicated cases, we may have different values in the same setting that trigger different operations (see the example below). It could also be a situation when the value is a Boolean and triggers the same operation but, in one case, makes it visible and, in another case, makes it invisible.

Let's take an example of the GHG Inventory reporting tool for selecting calorific value (GCV/NCV) in sub-sector 1.A for the energy sector. In this case, the setting has its own **UID** for tracking "c1f094b7-a73d-498a-aca9-14fe42cabd6d", **name** as "NCV_GCV choice", **order_number** as 5, and **variable_uid** where the result of the user selection needs to be saved "e0db25da-5544-4d08-ae7c-988bb3fd97a0". Saving the user choice allows them to change the setting later on by the user even after starting submission.

setting uid	name	order_number	variable_uid
c1f094b7-a73d-498a-aca9-14fe42cabd6d	NCV_GCV choice	5	e0db25da-5544-4d08-ae7c-988bb3fd97a0

The selection choice triggers different operations. In this case, there are only two values that can be selected by the user, the **setting UID** is "c1f094b7-a73d-498a-aca9-14fe42cabd6d" and there are two values trigger the change "NCV" and "GCV":

setting UID	value
c1f094b7-a73d-498a-aca9-14fe42cabd6d	NCV
c1f094b7-a73d-498a-aca9-14fe42cabd6d	GCV

When the user choice is **NCV** then update values in the drop-down with the value **NCV**, and when the choice is **GCV**, update values in the drop-down with the GCV respectfully.

Based on the requirement of being able to change settings at any moment, we also need to define when the user changes the setting to the opposite one, and we need to revert back to previous operations and apply a new set of operations. So, in this case, we have the following operations if the user selects a different value. Based on this logic, let's define operations for it:

setting UID	type	value	affected_entity_uid	affected_entity_type
c1f094b7-a73d-498a-aca9-14fe42cabd6d	setValue	NCV	ec0b0658-3bb2-4372-8be3-00c69c8cda73, a703924d-021a-4551-b484-011a0c9627d4, 22d03dfb-9cca-4a42-a749-0124dd2cb19d	variable
c1f094b7-a73d-498a-aca9-14fe42cabd6d	setValue	GCV	ec0b0658-3bb2-4372-8be3-00c69c8cda73, a703924d-021a-4551-b484-011a0c9627d4, 22d03dfb-9cca-4a42-a749-0124dd2cb19d	variable

where **setting UID** is a UID "c1f094b7-a73d-498a-aca9-14fe42cabd6d", **type** is a function to call *setValue*, **value** is specifying what the user selected, **affected_entity_UID** is the uid (also could be an array of uids) that needs to be passed as arguments to the function, and **affected_entity_type** is specifying what the entity affected by the function is. For example, the function to set a value (setValue) can be triggered if the value is set to NCV and only for the **variable** (affected_entity_type) with the **uid** "ec0b0658-3bb2-4372-8be3-00c69c8cda73", "a703924d-021a-4551-b484-011a0c9627d4", "22d03dfb-9cca-4a42-a749-0124dd2cb19d" (affected_entity_uid) in the navigation tree. The JSON setting will look the following:

Example: Version setting sample structure

```
1      {
2          "uid": "c1f094b7-a73d-498a-aca9-14fe42cabd6d",
3          "name": "NCV_GCV choice",
4          "order_number": 5,
5          "variable_uid": "e0db25da-5544-4d08-ae7c-988bb3fd97a0",
6          "roles": [
7              "Expert_LULUCF",
8              "Inventory_FP",
9              "Inventory_FP_alternate"
10         ],
11         "triggered_value": [
12             "NCV",
13             "GCV"
14         ],
15         "operation": [
16             {
17                 "operation": "setValue",
18                 "value": [
19                     "NCV"
20                 ],
21                 "affected_entity": [
22                     {
23                         "uid": [
24                             "ec0b0658-3bb2-4372-8be3-00c69c8cda73",
25                             "a703924d-021a-4551-b484-011a0c9627d4",
26                             "22d03dfb-9cca-4a42-a749-0124dd2cb19d"
27                         ],
28                         ...
29                     },
30                     "type": "VARIABLE"
31                 ]
32             },
33             {
34                 "operation": "setValue",
35                 "value": [
36                     "GCV"
37                 ],
38                 "affected_entity": [
39                     {
40                         "uid": [
41                             "ec0b0658-3bb2-4372-8be3-00c69c8cda73",
42                             "a703924d-021a-4551-b484-011a0c9627d4",
43                             "22d03dfb-9cca-4a42-a749-0124dd2cb19d"
44                         ],
45                         ...
46                     },
47                     "type": "VARIABLE"
48                 ]
49             }
50         ]
51     },
```