Data Cube Manual

Version 3.1.0

1. Data Cube Structure

In order for the system to properly work, the data needs to be provided following specific conventions. Not following the conventions or providing broken information (for example an invalid JSON file) will lead to system malfunctions and unpredictable results. It's up to the ISIpedia team to guarantee the correctness and the fulfilment of the conventions of the Data Cube.

For Data Cube (or briefly "Cube" or "cube"), we refer to a directory (a "directory" can be called also "folder") containing all the subdirectories and files produced by the ISIpedia team and their contributors.

1.1. Directory Structure

A. On the first level of the cube there is one directory called `country_data`, one (optional) json file called `plot-config.json`, and several **Indicator** directories.

B. Inside the `country_data` directory there are several **Country** directories and one directory for the **World**. Only the countries defined here are processed for each Indicator. If a country directory exists inside an Indicator but does not exists inside the `country_data` directory, it will be ignored.

C. Inside each country directory under `country_data` there is one json file containing the country configuration and statistical info displayed in the country report, and there could be several **Cross-indicator Climate Reports** directories.

D. Inside each **Indicator** directory there is a configuration file and one or more **Study Type** directories.

E. Inside each **Study Type** directory there are several **Country** directories.

F. Inside each **Country** directory there are several files: one markdown document, json files used to display line plots and / or calculate rankings, images and other documents you may find useful for the users to download.

G. The **world** directory contains a subdirectory `maps` containing global data in CSV files.

H. The required / allowed content of each **Cross-Indicator Climate Report** directory is the same of the **Country** directories inside the Indicators, with the difference that no ranking is calculated on the json files eventually available inside Cross-Indicator Climate Report directories, so the ranking-related tags will not work if used here.

1.2. Directories and Files convention

1.2.1. General names convention

Every directory name and every file name inside the cube can contain only these chars:

- English alphabet letters (`a-z, A-Z`)
- Digits (`0-9`)
- Underscores (`_`) and dashes (`-`)

Spaces are not allowed. The file extensions (the part of the name after the dot) have to be lower case.

Indicators directories and Cross-Indicator Report directories names need to be lower case.

1.2.2. Supported image formats

You can use **JPG**, **PNG** and **SVG** images. For raster images (JPG and PNG) the largest size (width or height) should be maximum 2500 pixels. Smaller versions for devices with small screens are generated automatically by the system.

1.2.3. Files used to calculate rankings

Json files used to calculate rankings need to follow a specific pattern:

```
<whatever-you-want>_<country-name>.json
```

The `<whatever-you-want>` part needs to be specified inside the **Indicators configuration**.

Any other plot file not used to calculate rankings can be arbitrary named as long as it follows the **General names convention**.

1.2.4. Country config file names

Config files for each country inside directories under the country_data folder follow this pattern:

```
<country-dir-name>_general.json.
```

Examples of valid names: AFG_general.json, FRA_general.json.

2. Indicators configuration

Inside each Indicator directory there should be a configuration file, named exactly `config.json`. It contains information about the study types, the files used to calculate rankings and the Topics to which the Indicator belongs.

Here is an example of a config.json file:

```
{
    “name”: “River flood”,
    “topics”: [“Extreme events”, “Water”, “Agriculture”],
    “study-types”: [
        {
            “directory”: “ISIMIP-projections”,
            “name”: “Future Projections”,
            “description”: “Short description of the study type”,
            “ranking-files”: {
                “land-area-affected-by-river-flood-absolute-changes”: {
                    “direction”: “desc”,
                    “min-max”: [-20.0, 150.5]
                },
                “land-area-affected-by-river-flood-relative-changes”: {
                    “direction”: “desc”,
                    “min-max”: [0.0, 1058.96]
                },
                “population-exposed-to-river-flood-absolute-changes”: {
                    “direction”: “desc”
                },
                “population-exposed-to-river-flood-relative-changes”: {
                    “direction”: “desc”,
                    “min-max”: [-100.0, 32.53]
                },
            }
        }
    ],

}
```

The `name` field is optional, if not provided the Indicator Name will be generated by replacing every dash in the directory name with a space and capitalizing the first letter.

Each Indicator can belong to multiple `topics` and must belong at least to one. The topic names are like “categories”, so they should be different from the indicator names. We provide a proper icon for each of these topic names: “Extreme events”, “Agriculture”, “Permafrost”, “Water”, “Energy”, “Fisheries”, “Biodiversity”, “Lakes”, “Coastal infrastructure”, “Health” and “Forests”. Any other name will not have a matching icon.

The `study-types` field is required to specify which study types are implemented for the given indicator. It is an array of objects, each object defines the `directory` name, the `name`, a `description` of the Study Type and the ranking files. We provide a proper icon for each of

these Study Types: “Future Projections”, “Model Evaluation” and “Detection and attribution”. Any other name will not have a matching icon.

The `ranking-files` field defines the list of files used to calculate rankings. For each entry, the key is the file name prefix and it contains at least the field “`direction`”, defined as:

- `asc`: Ascending order, smaller values come first
- `desc`: Descending order, bigger values come first.

It can contain also the field “`min-max`” (which is optional), which is an array of two float numbers. If provided, it could be used to define the color edges in the global ranking maps and in the country maps.

The corresponding json files follow the convention defined in **Files used to calculate rankings**. So for example for the entry “`land-area-affected-by-river-flood-absolute-changes`” there will be a file called:

```
land-area-affected-by-river-flood-absolute-changes_AFG.json
```

where AFG is the directory name for the Afghanistan country.

It is important to note that any json file following this name convention but not being defined inside the `ranking-files` field of the config, could be used to generate a line-plot but will not be used to calculate any ranking.

3. Countries configuration

Inside each country directory inside the country_data folder, there will be a configuration file, named after the convention **Country config file names**.

Here is an example of a `AFG_general.json` config file:

```
{
     “name”: “Afghanistan”,
     “type”: “country”,
     “sub-countries”: [],
     “stats”: [
          {
               "type": "POP_TOTL",
               "label": "Total population",
               "unit": "million people",
               "value": "31.29",
               "rank": "40.0"
          },
          {
               "type": "POP_DNST",
               "label": "Population density",
               "unit": "people/sq. km",
               "value": "47.93",
               "rank": "114.0"
          }
     ],
     “reports”: {
          “report-subdir-1”: “Cross-indicator Climate Report about Afghanistan”,
          “report-subdir-2”: “Another Cross-indicator Climate Report Title here”
     }
}
```

The `name` field defines the full name of the country displayed inside the Configurator and the reports.

The `type` field can be one of: “**country**” or “**global**”. “global” must be used only once, for the world directory.

In case you are grouping some small areas into a single country, you can fill the “`sub-countries`” field with an array of names of the included areas. In this way it will be possible to search for them in the Configurator finding the aggregating country as a result. An example could be:

```
{
     “name”: “Italy”,
     “type”: “country”,
     “sub-countries”: [“San Marino”, “Vatican City”],
     …
}
```

In this scenario, searching for “Vatican” will suggest Italy as a result.

The `stats` field is an array of objects, each object defines an information related to the country that will be displayed in the “Country Details” section at the beginning of each Country Report. In the example above there are only two examples, but the number of entries and their content is totally up to you. If a value is not available it has to be indicated with `“nan”`. If the unit is not necessary it has to be indicated as an empty string: `“”`. Values and units will be displayed “as is”, without applying any additional formatting.

In order to associate a country map to a line-plot, there should be also a file `bounds.json` and an image named `country.svg`.

This is an example of `bounds.json`:

```
{"code": "AFG", "bounds": {"left": 60.0, "right": 75.0, "bottom": 29.0, "top": 39.0},
"indices": {"left": 480, "right": 509, "bottom": 121, "top": 102}}
```

All the values are calculated by you, and allow to extract the right values from the World Data CSV files. When the country territories wraps around -180° or 180° meridians (like USA or Russia), there is an additional field in the json object:

```
"splitted": true
```

Currently, only `indices` and the optional `splitted` field are taken into account by the code. The other fields (code and bounds) could be removed safely.

In the `reports` field you can define any **Cross-indicator Climate Report** or any other kind of Report you would like to include, related to the specific Country. For each entry, the key indicates the directory name (lower case is mandatory) and the value indicates the title of the report that will be displayed inside the Report Cover as the main Page Title. These defined reports directories are stored in the same level of the configuration file, so following our previous example the directory structure would look like this:

```
- country_data/
-- AFG/
--- AFG_general.json
--- bounds.json
--- country.svg
--- report-subdir-1/
--- report-subdir-2/
```

4. Plot Configuration

You can control most of the plot and map features from a configuration file inside the cube directory, called plot-config.json.

If you do not put that file in the cube, the system will fallback to a default version available inside `/assets/json/plot-config.json`.

Here is the current version of the file (with some omissions for brevity, please refer to the default one as a working example):

```
{
    "linePlot": {
        "individualSimulation": {
            "label": "Individual model simulation",
            "lineSize": 1,
            "notSelectedOpacity": 0.1,
            "dotsNotSelectedOpacity": 0.5,
            "legendColor": {"rgba": [0, 0, 0, 0.4]}
        },
        "gcmMedian": {
            "label": "GCM median",
            "lineSize": 5,
            "notSelectedOpacity": 0.1,
            "dotsNotSelectedOpacity": 0.5,
            "legendColor": {"rgba": [0, 0, 0, 0.4]}
        },
        "multiModelMedian": {
            "label": "Multi-model median",
            "lineSize": 5,
            "notSelectedOpacity": 0.6,
            "dotsNotSelectedOpacity": 0.5,
            "legendColor": {"rgb": [0, 0, 0]}
        },
        "interAnnualVariability": {
            "label": "Inter annual variability",
            "color": "pattern",
            "patternOpacity": 0.5,
            "patternBackgroundOpacity": 0.1
        },
        "interModelOpacity": 0.1,
        "interModelLegendOpacity": 0.4,

        "scenario": {
            "all": {
                "color": {"rgb": [242, 115, 90]},
                "interModelLabel": "Inter model spread"
            },
            "historical": {
                "label": "Historical Period",
                "color": {"rgb": [0, 0, 0]},
                "legendColor": {"rgba": [0, 0, 0, 0.4]}
            },
```

```
                "rcp26": {
                      "label": "2°C Compatible Scenario (RCP2.6)",
                      "color": {"rgb": [0, 52, 102]},
                      "interModelLabel": "Inter model spread RCP2.6"
                },
                "rcp60": {
                      "label": "Business-As-Usual Scenario (RCP6.0)",
                      "color": {"rgb": [196, 121, 0]},
                      "interModelLabel": "Inter model spread RCP6.0"
                }
          },

          "climateModel": {
                "GFDL-ESM2M": {},
                "IPSL-CM5A-LR": {},
                "MIROC5": {}
          },

          "impactModel": {
                "CLM45": {},
                "GEPIC": {},
                "..."
          }
    },

    "maps": {
          "colors": {
                "positive": {
                      "hueRange": [22, 1],
                      "saturationRange": [0.86, 0.61],
                      "lightnessRange": [0.89, 0.35]
                },
                "negative": {
                      "hueRange": [201, 212],
                      "saturationRange": [0.51, 0.9],
                      "lightnessRange": [0.88, 0.2]
                },
                "zero": {"rgb": [255, 255, 255]},
                "noData": {"rgb": [241, 244, 244]}
          },
          "rankingMapColors": {
                "noData": {"rgb": [187, 195, 195]}
          },
          "countryMapColors": {
          },
          "countryMapCountryBasedMax": true,
          "rankingMapSelectorBasedMax": true
    }
}
```

Most of the content is self explanatory, just setting line colors, line sizes, opacity values and labels. For each climate model and impact model you can add a “`label`” field inside the respective object, in order to use a nicer name in the ranking info.

In the map section there are values related to the global ranking maps and country maps. The generic values inside “`colors`” can be overriden for the specific type of maps, like the `noData` color in the example.

There are two boolean flags that need to be explained:

“countryMapCountryBasedMax": this flag controls the range of values used to build the color scale for the country maps. When it is true, the system will extract the maximum ranking values for the specific variables belonging only to that specific country. This implies that the color scale will be almost totally used in the visual representation, leading to nicer more colored maps, but at the same time implies that maps of different countries will use different color scales. Setting the flag to false will generate color scale based on the “`min-max`” value indicated in the config of the specific variables or, if this value are not declared, to the maximum ranking values belonging to any available country. This will lead to “flatter” maps, having the same scale for each country.

"rankingMapSelectorBasedMax”: this flag controls how the color scale is generated for the global ranking maps. If the flag is true, the legend will use the maximum value limited to the specific temperature (or timeslice) indicated in the ranking map tag. If it is false, instead, it will use the maximum value belonging to the whole variable (eventually using the `min-max` from the config, if it was specified) leading to “flatter” maps, because the maximum color could be mapped to a value belonging to a different temperature (or timeslice), so not visible in the current map visualization.

5. Global Data

Inside the `world` country directory there is a subdirectory `maps`.

Inside maps there is a folder for each ranking variable defined in the indicator configuration.

Inside each of this directory there are several csv files, one for each combination of scenarios, climate models, impact models and "selectors" (temperature or timeslice values). The file names are made by the names of these four components, separated by underscores.

When there is no scenario (like for temperature-based values) the value used is "all". For medians, the value should be "median". Here are examples of the file names:

```
all_median_median_0.csv (overall median values for 0°C)
all_median_median_1.5.csv (overall median values for 1.5°C)
all_IPSL-CM5A-LR_median_2.csv (median values for IPSL-CM5A-LR climate model at 2°C)
all_MIROC5_CLM45_1.csv (values for MIROC5 climate model and CLM45 impact model at 1°C)
piControl_median_median_1861-1880.csv (overall median for piControl scenario in the
timeslice 1861-1880)
piControl_GFDL-ESM2M_median_1961-1980.csv (median for piControl scenario using GFDL-
ESM2M climate model in the timeslice 1961-1980)
rcp26_median_median_2041-2060.csv (overall median for rcp26 scenario in the timeslice
2041-2060)
rcp26_MIROC5_CLM45_2061-2080.csv (values for rcp26 scenario using MIROC5 climate model
and CLM45 impact model in the timeslice 2061-2080)
```

Inside each csv file, there are 360 lines. Each line contains 720 comma-separated floating point number. When the value is not available, its place has to be left empty (the number of elements per row has to be always 720).

6. Update procedure

The Data Cube directory needs to be uploaded by the ISIpedia team on the `/cube` path on the document root of the web server hosting the ISIpedia website. Please be aware that – once the website is on production and publicly accessible – this update procedure needs to be executed as a "sync" (add the new files, overwrite the changed files and remove the files not needed any more), you cannot totally remove all the previous content and then after that upload the new one, because it will leave the website in a broken state for a long period of time.

Whenever anything in the cube changes, after the directory sync an additional procedure needs to be executed in order to process the information inside the cube and to keep the website in perfect sync with it. Please be aware that this procedure, due to the intrinsic interconnection of the data inside the cube, needs to process all the data every time, so it could last several minutes, based on the amount of data available inside the cube. More indicators means more time, more countries means more time, more rankings to be calculated means more time, and so on.

The additional procedure to be run after you upload the files is located at https://isipedia.org/data-sync, it requires you to be logged-in in the CMS Panel, otherwise it will not work. You have to click on "Start Data Sync" and wait with the page opened until it finishes. If you have connection issues during the process, it is safe to reload the page, it will continue from the last point till the end. There is a progress bar to give you a hint of the required time.

The Data Sync automatically invalidates the PDF Cache, anyway you may need to do this invalidation on its own, without running again a full Data Sync. For this reason there is a separate button for that. This action is instant.

7. Report Format

7.1. Requirements

There should be only one markdown file (with .md extension) inside each report directory in the Data Cube. Being only one, the name is not important for us, so you can use any name as long as it is respecting the **General names convention**. Having more .md files will lead to unpredictable results (you cannot know which one will be used).
A directory without any .md file inside will be totally discarded, even if it contains other files.

7.2. Markdown

Please refer to this page as the official Markdown formatting spec.:
https://daringfireball.net/projects/markdown/syntax

7.2.1. Formats with full support

These are the formatting elements that are fully supported as described in the Markdown spec:

- Paragraphs and Line Breaks
- Blockquotes
- Lists
- Horizontal Rules
- Emphasis

7.2.2. Formats with limited support

Headers

Only two levels of headers (titles and subtitles) are supported in the impact report format, with a specific syntax:

```
### This is a first level title
…

#### This is second level title
…
```

Please be aware that:
- first level titles (those with this prefix: "###") will be automatically linked in the Page Menu;
- there is no need to insert the Indicator Name as a title, because it will be automatically added at the beginning of the report when the web page is generated.

Headers in cross-indicator climate reports

Inside these reports an additional header level is available:

```
## This is a first level title

### This is a second level title

#### This is a third level title
```

In this case both first and second levels (## and ###) will be automatically linked in the Page Menu.

Links

Only “inline” type of links are supported. The “reference” type cannot be used.
There is also an additional syntax to be used for internal links in order to allow the system to correctly resolve the URLs. Please refer to the section **Internal Links**.

7.2.3. Formats not supported

Original Markdown format for including images is not supported.
Please refer to the section **Images** in this document for the supported way of including images in the reports.

7.2.4. Comments

You can insert comments in the report code with this syntax:

```
<!-- this is a comment -->
```

Comments are not visually rendered in the web page, but please be aware that they will not be removed from the code, so any user could potentially read them looking into the page source in the browser.

7.3. Special Tags

7.3.1. Internal Links

Links to editorial pages

By “editorial pages” we mean pages created with the CMS, like Glossary terms or Stories.

```
(link: relative-path text: Link text here)

<!-- example: -->
(link: glossary/climate-change text: Climate Change)
```

The relative path is the part of the page URL coming after the site domain. Example: https://isipedia.org/**glossary/climate-change**. The obvious implication is that you cannot insert links to pages that are not existing yet.

Link to a Cross-indicator Climate Report

To link a Climate Report from a Climate Impact Report:

```
(link: climate-report/<lower-case-country-dir>/<report-dir> text: Link text here)

<!-- example: -->
(link: climate-report/afg/climate-in-afghanistan text: Climate Analysis in Afghanistan)
```

Link to files inside the Report directory

```
(download: file-name.ext text: Link text here)

<!-- example: -->
Download the (download: climate-change.pdf: text: Climate Change Document).
```

7.3.2. Anchor Links

If you want to add an anchor link in the page menu referring to a position of the document in which there is not a title, you can use this tag:

```
(menu-anchor: Intro)
<!-- will generate a link in the Menu with a label “Intro” -->
```

7.3.3. Rankings and Global Rankings Map

There are several tags related to rankings. Some of them can be used inside the country reports, the others can be used both inside global and country reports.

The file name specified in the tags for the country reports includes the country suffix (underscore followed by the country folder, like “_AFG”), while the name specified in the tags for the global report is without country suffix.

For each ranking tag you can specify these parameters:

- **`time`**: it is an interval expressed as two years separated by a dash, no spaces. It has to match exactly with one of the intervals defined in the field `timeslices_list` of the json file. It is not meaningful for files “against temperature”.
- **`temperature`**: it is one of the number indicating the Celsius degrees defined in the field `temperature_list`. It is not meaningful for files “against timeslices”. It is alternative to **`time`**, it does not make any sense to use both in the same tag.
- **`scenario`**: it is one of the value defined in the field `climate_scenario_list` of the json files. It is not meaningful for files “against temperature”.
- **`climate-model`**: it is one of the value defined in the field `climate_model_list` of the json files.

- **`impact-model`**: it is one of the value defined in the field `impact_model_list` of the json files.

If you do not specify `scenario`, `climate-model` or `impact-model`, the corresponding medians will be used. Be aware that for ranking "against timeslices" a `scenario` is always required, and one parameter between `time` and `temperature` is always required as well.

Ranking-value

This tag is meant to be used *inline,* to include single values inside a text paragraph in a country report:

```
(ranking-value: file-name-with-country-suffix value: position temperature: 2 climate-
model: IPSL-CM5A-LR impact-model: GEPIC)
<!--
the value parameter can be one of:
number           (1, 45, 33)
position         (1st, 45th, 33rd)
total            (amount of countries in the dataset, like 182)
total-positions  (amount of different positions, like 123)
-->
```

Ranking-area

This tag can be used *inline* inside the global report, to include information about a specific entry in the ranking list.

```
(ranking-area: file-name-without-country-suffix order: 3 value: name temperature: 2
climate-model: IPSL-CM5A-LR impact-model: GEPIC)
<!--
value parameter can be one of:
- name           (France, Germany, Afghanistan)
- number         (1, 45, 33)
- position       (1st, 45th, 33rd)
```

Please take in mind that several countries can have the same exact value, so they will share the position (three countries could be in 3rd position, for example). For this reason you have to specify the order (always growing number) that not necessarily matches with the position.

This is an invented example of the rankings with the corresponding values:

Order	Number	Position	Name
1	1	1st	Afghanistan
2	2	2nd	France
3	2	2nd	China
4	4	4th	Italy
5	4	4th	Brazil
6	6	6th	Spain

Based on the data of this example, this peace of code

```
(ranking-area: file-name-without-country-suffix order: 3 value: name temperature: 2
climate-model: IPSL-CM5A-LR impact-model: GEPIC) ((ranking-area: file-name-without-
country-suffix order: 3 value: position temperature: 2 climate-model: IPSL-CM5A-LR
impact-model: GEPIC))
```

will generate the text: **China (2nd)**.

Ranking-map

This tag generates a colour-coded ranking global map in the global report.

```
(ranking-map: file-name-without-country-suffix temperature: 2 climate-model: IPSL-CM5A-
LR impact-model: GEPIC)
```

Ranking-download

This tag generate a link to download the rankings in a text file.

```
(ranking-download: file-name-without-country-suffix temperature: 2 climate-model: IPSL-
CM5A-LR impact-model: GEPIC text: Link text here)
```

7.3.4. Line Plots

To include a Line Plot section in your report you need to insert the line-plot tag. A line-plot section can display data from a single json file (against temperature or against timeslices) or from two json files (both against temperature and against timeslices). In the second case it will have a button to switch between the two datasets.

In case of two files, their names need to be separated with a comma, without any space.

```
<!-- single file version: -->
(line-plot: single-file-name first-temperature: 2)

<!-- two files version: -->
(line-plot: first-file-name,second-file-name first-temperature: 2 second-scenario:
rcp60 second-time: 2041-2060)
```

The files will be used in the given order.

In a similar way to the ranking tags, you have to specify parameters to pre-select a default point for all the files indicated. The available parameter are:

`first-scenario`, `first-climate-model`, `first-impact-model`, `first-temperature`, `first-time`, `second-scenario`, `second-climate-model`, `second-impact-model`, `second-temperature`, `second-time`.

Parameters starting with first- will refer to the first file, while those starting with second- will refer to the second file.

If the file name specified for the line-plot is one of those configured to be used to calculate rankings, the line-plot section will display also a ranking graph that will automatically be updated accordingly to user – plot interactions.

Line plots can be downloaded in hi-resolution PNG format (suitable for printable documents) with the specific button.

7.3.5. Images

To include an image with full width (accordingly with the design):

```
(image: file-name.ext alt: Image description here caption: Image caption here)
```

To include an image displayed at – more or less – half of the full width:

```
(image: file-name.ext alt: Image description here caption: Image caption here class:
medium)
```

To include an image inline inside a paragraph, keeping the image's original size:

```
(image: file-name.ext alt: Image description here class: inline)
```

Inline images do not have a caption.

7.3.6. Math Expressions

To include mathematical expressions in your report you have to wrap the formula in special tags, like this:

```
(math-formula)your math expression here(/math-formula)
```

The system uses KaTeX to render the expressions in the browser, so please refer to this page for the supported functions: https://katex.org/docs/supported.html

8. How to update the SVG World Map

The World Map used in the Configurator and in the Ranking Maps is currently stored in `/assets/img/map.svg`.
If you like to modify it (in order to change country borders, add or remove a country, and so on) you can duplicate it and put it in the root of the data cube in `/cube/map.svg`.

As you can see in the file, each country is made by one or more paths (`<path>`), eventually grouped inside a group (`<g>`), having the `id` attribute set to the corresponding three-letters country code (lowercase). Moreover, the country (so the group or the single path) requires the `class="area"` attribute.

There are some countries' territories that are spread around a big portion of the world, like USA or France. Since selecting a country make the configurator map to zoom into view its territories, there is the possibility to indicate a "main" territory in the map, in order to force the map to take into consideration only that smaller surface for the zoom, giving the user a better experience. In order to achieve that, you have to put `class="area main"` in the specific sub group / path inside the country group (please refer to France as an example).