Briefing:
File Formats in Climate and Energy System Modelling

(alphabetical order)
Lukas Emele, Hannah Förster, Martin Glauer, Christian Hofmann, Ludwig Hülk, Mirjam Stappel, Christian Winger

Version 0.0.1 – July 29, 2020

Contents

1 Introduction

With this briefing we provide definitions for relevant file types, formats, data bases and concepts commonly used in climate and energy system modelling.

We offer this document, because not all data is created equal. We have to deal with a multitude of data sources for input and output.

Some information comes in special formats and some information can be stored in different ways. Each way comes with specific advantages and drawbacks.

Sometimes, conversion is needed and may not be straight forward. These particularities are important with respect to reproducibility, comparability, reuse and documentation.

We lay out this reference to allow a productive approach in working with commonly used file formats and related issues in climate and energy system modelling.

We introduce binary and text based file formats, the difference between vector and raster formats, the classification of the structuredness in data, and the use of databases. Subsequently, we provide a description of specific, commonly used formats from all categories. Whenever possible we reference an external resource for further explanation. Finally, we conclude with a short section of best practices and recommendations.
This briefing may develop further and is open to suggested additions by anyone. Please post suggestions here:
`https://github.com/OpenEnergyPlatform/tutorial/projects/1`. Please attach the label *formats-briefing* to your issue.[1]

2 Binary and Non-Binary Data

At a generic level of description, there are two kinds of computer file formats: text files and binary files.

Binary data in computer science describes all data in a raw state of ones and zeros. For the data to be interpreted, some context must be provided. For example, the same binary sequence can be interpreted as a fixed number or an ASCII character set. The computer can identify the file/data format on the basis of meta information in the binary sequence of a file. The file format usually gives that contextual information. One way for computers and humans to find out what kind of data is presented, is to look at the file name extension and the specification of the corresponding file format.

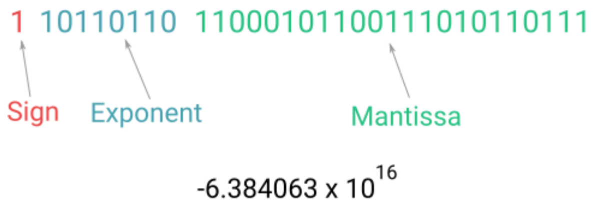


Figure 1: Binary representation of a single precision floating point.

The zeros and ones that make up binary data are called bits. Bits are processed by the computer in fixed groups. Eight bits constitute one byte and four bytes constitute one word. The position and state of the individual bit in a byte sequence is relevant for processing. Figure 1[2] illustrates the binary representation of a float number. The leading (left) single bit defines whether the number is positive or negative. Table 1 shows how geographic coordinates are stored as well known text (WKT) and well known binary (WKB) respectively.

[1] `https://github.com/OpenEnergyPlatform/tutorial/labels/formats-briefing`

[2] `https://ryanstutorials.net/binary-tutorial/binary-floating-point.php`

Dataformat	**Representation**
WKT	Point(10 10)
WKB	0101000000000000000000F03F000000000000F03F

Table 1: Representation of geographical data in human readable and binary format. Well known binaries are usually represented as hexadecimal strings to save space. That is why there are other symbols than one and zero in the representation.

While it is possible in theory, humans are not very good, or at least very slow at interpreting a given set of ones and zeros. Binary data is therefore considered to be only machine readable. Only the interpreted or translated information, for example an alphabetic text, is considered to be human readable. In contrast, if an information is considered not to be machine readable, it usually means that structure or context is missing, which prevents processing of that information with an algorithm. All information can be broken down to ones and zeros, but for a computer it is not straight forward to interpret the number 3 when it was drawn by hand and then stored as a digital photograph.

The term **non-binary** data is not defined. As mentioned before, all information can have a binary representation. In our context we define non-binary data as data with a textual representation. The main purpose of providing digital data as text is to make it human readable, which greatly improves understanding and facilitates maintenance of digital infrastructure. It comes with the drawback of taking up more space and being less efficient in processing.

3 Vector vs Raster Data

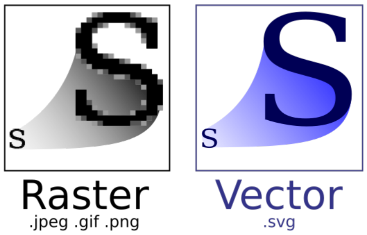


Figure 2: Close-Up of Raster and Vector Images

Digital representation of information can be achieved in fundamentally different ways. Raster data information is stored in a grid of columns and rows, where every cell contains a value. This is most commonly used in images where every cell is a picture cell or a pixel containing a specific colour information. But the concept also extends to different kinds of geographic data. Working with raster data tends to be fast. Display of raster data requires little computing power, because the grid of the image is easily translated to the pixel grid on the screen. Common raster formats are JPG, PNG, GIF and TIFF. Due to it's gridded nature, curves and gradients can only be approximated in raster files. The level of detail that can be covered hinges on its resolution. This characteristic is depicted in figure 2[3]. Vector data offers a solution for this shortcoming.

With vector data, objects are stored in so called *paths*. For instance a point is stored as a combination of its coordinates in a reference system and its attributes. A line is stored as a connection between a starting point A and an ending point B, each with coordinates. Attached to the line can be characteristics like its thickness, in what way it is curved, its colour and so on. Vector data is generally characterised as slower than raster, but also more "correct"[4]. A common file format for vector data in images is SVG. Common vector data formats for spatial information include GPKG, GeoJSON, SHP, GML, KML and OSM.

4 Data Model and Structuredness of Data

Below, we summarise the key elements and terms used in data management from a practitioners point of view. There are different fundamental principles how data is organised. These principles are called the data models. A good way to categorise data is to look at the structure. There are three main categories: unstructured, semi-structured and structured data:

- *Unstructured data* is information that has no pre-defined format or organisation. Examples of unstructured data are text files, videos and audio files. While there can be some sections or patterns, the information is mostly without structured context. By enriching data with tags or other markers to separate semantic elements and to enforce hierarchies of records and fields, it becomes semi-structured data.

- *Semi-structured data* is sometimes described as having a self-describing structure. A computer can process (parse) this information more easily than it can unstructured data. Typical semi-structured file formats are XML and JSON.

- *Structured data* is information that conforms to a well defined data model. This data model explicitly determines the structure of data, like the existing columns, their names and data types.

In energy system analysis, data mainly exists in the relational mode; it is organised in tables. The data is arranged in relations (tables) with defined attributes (columns) and tuples (rows). At this point, the same data (and its information) can exist in all three structures but its usability can vary significantly.

[3] Demonstration of differences between bitmapped and SVG images. CC-BY-SA 2.5 `https://commons.wikimedia.org/wiki/File:Bitmap_VS_SVG.svg`

[4] `https://www.esri.com/content/dam/esrisites/en-us/media/pdf/teach-with-gis/raster-faster.pdf`

5 Databases

Files are a good and easy solution for many small tasks. They can be created and moved quickly and easily. However, files come with drawbacks attached: They are often bound to the software program with which they were created. Saving data in files also comes with redundancies. Large amounts of data require either very large unhandy files or a distributed file system. Maintaining data in files is therefore high in maintenance and prone to errors. Furthermore it is difficult to analyse data contained in files.

An alternative way to store data is to use databases. A database consists of two components:

a. the actual data, and

b. a software to manage it, which is called database management system (DBMS).

The DBMS organises read and write access to the data, as well as a structured way to store it. Databases have several advantages over files. Redundant information can be avoided, as every information is only stored in the database once. Data from a database can be used in other programs. The data can be accessed by several programs and users at the same time. A database can handle very large amounts of information efficiently, track versions and allow for fast querying. Finally, the user interface is not dependent on a program. Instead all programs use the same application programming interface (API) to access the database.

There are some drawbacks to using a database, however. Creating, maintaining and adapting a database is neither an easy nor a quick task. Usually, a programmer is required to do this. Establishing access, also entails more work than opening a simple file. Finally, there are the problems that come with a centralised structure. Any errors in the data will transfuse to all applications. Writing access therefore needs to be managed. Reading access can be granted publicly, however.

A public database is therefore well suited in situations where many people and programs need to access large amounts of data in an efficient, structured, open and reproducible manner. In other words, it is the system of choice for open science.

5.1 PostgreSQL

PostgreSQL, also known as Postgres is a database management system, compliant with the Structured Query Language (SQL) standard. It is a free and open-source tool to handle data in a database.

5.2 Sqlite

For smaller projects, a single file database like sqlite may be sufficient. The data can still be queried and analysed similarly to a fully fledged DBMS. Distribution of the data is easier as it is contained in a single file. Concurrent writing access however is not practical with sqlite.

6 Data Formats

6.1 Geographic File Formats

There is a plethora of geographic file formats in use. Many of them are geared towards very specific use cases. The ones listed below are either more general purpose formats or ones that are in widespread use. Geographic data always comes with some degree of structure.

File Type	**Binary / Text**	**Vector / Raster**	**Structuredness**
GeoJSON	text	vector	semi-structured
GeoTIFF	binary	raster	semi-structured
GML	text	vector	semi-structured
GPKG	binary	both	structured
GPX	text	vector	semi-structured
KML	text	vector	semi-structured
OSM	text	vector	semi-structured
SHP	binary	vector	semi-structured

Table 2: Overview of Common Geographic File Formats.

6.1.1 GeoJSON

GEOJSON is designed for representing simple geographical features, along with their attributes. These include points, line strings, polygons and multi-part collections of these types. GeoJSON is an open standard format based on JSON.

6.1.2 GeoTIFF

GeoTIFF is a public domain metadata standard which allows georeferencing information to be embedded within a TIFF file. This potential additional information includes map projection, coordinate systems, ellipsoids, datums, and everything else necessary to establish an exact spatial reference for the file.

6.1.3 GML

The Geography Markup Language (GML) is used to express geographical features. It uses XML grammar. GML serves as a modelling language for geographic systems as well as an open interchange format for geographic transactions on the Internet. GML is able to integrate all forms of geographic information, including not only conventional ”vector” or discrete objects, but also coverages and sensor data.

6.1.4 GPKG

GeoPackage (file extension .gpkg) is an open, platform-independent, standards-based data format for geographic information. Files are technically a SQLite database container. It is fast, small, well structured and well supported. When at all in doubt which file format you should use when working with geographic information, use this file format.

6.1.5 GPX

GPX, stands for GPS Exchange Format. It is an XML schema designed as a GPS data format for software applications and can be used to describe waypoints, tracks, and routes.

6.1.6 KML

Keyhole Markup Language, abbreviated KML is an XML notation for expressing geographic annotation and visualization within two-dimensional maps and three-dimensional Earth browsers. It was developed for use with Google Earth.

6.1.7 OSM

OSM is a file format for geodata that originated in the OpenStreetMap project. The format builds on the xml standard. To save space .osm files are often compressed into a binary .pbf-file.

6.1.8 SHP

A shapefile is a multi-file vector format for geographic information systems. It was developed and is maintained and regulated by Esri and has been the de-facto standard for GIS software products since the early 1990s. Due to it's age there are a number of challenges to using shapefiles, like limited data types, limited number of characters for attributes, 2 GB size limitation, and the fact that it is a multi-file format. There is at least one dedicated website that discourages the continued use of shapefiles.[5] We also highly recommend using newer standards. GeoPackage comes without the limitations of shapefiles, adds some speed and is one of the best candidates to become the successor of the old standard. A human readable, text-based format with similar functionalities as shapefiles is GeoJSON.

6.2 Office File Formats

Office File Formats are a category with somewhat fuzzy boundaries. Broadly speaking this category includes all file formats that are commonly created, edited or exported with the use of common office software suites such as LibreOffice or Microsoft Office.

[5] http://switchfromshapefile.org/

File Type	Binary / Text	Vector / Raster	Structuredness
CSV	text	vector	semi-structured
ODF	binary	vector	unstructured
PDF	text	vector	unstructured
XLS	binary	vector	semi-structured
XLSX	text	vector	semi-structured

Table 3: Overview of Common Office File Formats.

6.2.1 CSV

CSV stands for comma-separated values. A CSV file consists of delimited text using a comma to separate values. The format is not fully standardised, so sometimes instead of a comma, other delimiters are used. Common other separators include semicolons and tabulators. Each line of a CSV file is one data record. Each record consists of one or more fields. A CSV file typically stores tabular data.

6.2.2 ODT

The Open Document Format for Office Applications is an xml-based file format for spreadsheets, charts, presentations and word processing documents. The text files are zip compressed.

6.2.3 PDF

The portable Document Format was developed to store electronic text documents that look the same independent of hardware or operating systems. A pdf file in itself is a vector description of page layout that allows free scaling. It can incorporate, among other things, vector data and raster images.

6.2.4 XLS

XLS stands for Excel Binary File Format and is a proprietary, binary file format used by Excel as its primary format. It is used for tabular data. Since 2007 Excel uses the Office Open XML (XLSX), but still supports the old standard.

6.2.5 XLSX

Office Open XML (file extension xlsx) is an XML-based format for tabular data and the successor of xls as the standard file format in Microsoft Excel.

6.3 Image File Formats

Different image file formats can mostly be converted between each other. Usually this entails a drawback. Conversion to a different format may come at the cost of increasing storage size, losing image quality, interactivity or a different quality of the original file format. The file formats listed below are well adapted to special use cases. We highly recommend to align your use-case with the qualities of the format you plan to use.

File Type	Binary / Text	Vector / Raster	Structuredness
JPEG	binary	raster	unstructured
JPG	binary	raster	unstructured
PNG	binary	raster	unstructured
SVG	text	vector	semi-structured
TIFF	binary	raster	semi-structured

Table 4: Overview of Common Image File Formats.

6.3.1 JPG

See JPEG.

6.3.2 JPEG

JPEG is a commonly used method of lossy compression for digital images. The degree of compression can be adjusted, allowing a selectable trade-off between storage size and image quality. Common file extensions are .jpg and .jpeg. JPEG is an acronym for the Joint Photographic Experts Group, which developed the standard. We recommend to use JPEG files for photographic images when maintaining the original quality isn't important, such as in web applications or written publications. It is also possible to store photos as JPEG without a lossy compression. So it is also possible to use JPEG for example in large prints. If you want to make sure however, that the values of each individual pixel from the original photograph stays the same, which is important e.g. in aerial photos, we recommend using TIFF instead.

6.3.3 PNG

PNG stands for Portable Network Graphic. It is a binary raster graphics file format, supporting lossless data compression. It was developed for transferring images on the internet and therefore only supports rgb colourspace. We recommend using this format for non-photographic raster images in text documents or web resources.

6.3.4 SVG

Scalable Vector Graphics is an open xml file format standard used for two-dimensional graphics with support for interactivity and animation. We recommend using this format for graphs, diagrams, workflow charts and similar figures in documents or web resources.

6.3.5 TIFF

Tagged Image File Format, abbreviated TIFF or TIF is used for raster graphics images. It can store image data in a lossless format. It is flexible and adaptable and can handle images and data within a single file, by including header tags. A TIFF file, for example, can be a container holding JPEG (lossy) and PackBits (lossless) compressed images. File extensions are either .tif or .tiff.

6.4 Other File Formats

Listed below is a selection of file formats with a wide range of use cases that are difficult to place into a broader category. It includes generically applicable standards, as well as formats with a specific and narrow use case. We included all formats, that are somewhat frequently encountered in energy system modelling.

File Type	Binary / Text	Vector / Raster	Structuredness
DAT	any	any	any
JSON	text	vector	semi-structured
PBF	binary	vector	structured
PostgreSQL	binary	both	structured
RDF	text	vector	semi-structured
TXT	text	other	unstructured
WAV	binary	other	unstructured
XML	text	vector	semi-structured

Table 5: Overview of Other Common File Formats.

6.4.1 DAT

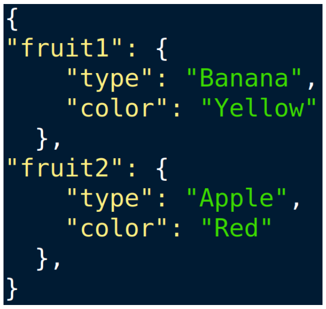
A file with the .dat file extension is a generic data file that stores specific information proprietary to a program that created it. A file may contain any kind of information, text-based or binary. There simply is no common standard.

6.4.2 JSON

JavaScript Object Notation (JSON) is an open standard file- and data interchange format. It uses human-readable text to store and transmit data objects consisting of key–value pairs and array data types. Figure 3 illustrates an example JSON-file. JSON is a very common data format especially for web based applications and serves among other things as a replacement for XML.

6.4.3 PBF

Protocol Buffers is a method of serialising structured data. The concept of structuredness is explained in 4. Serialisation means to bring data into a sequential form so that the computer can read it in a consecutive manner. This process is



```
{
"fruit1": {
    "type": "Banana",
    "color": "Yellow"
  },
"fruit2": {
    "type": "Apple",
    "color": "Red"
  },
}
```

Figure 3: Example JSON structure. The file contains two fruit objects which have two keys each, stored in the key-value pattern. ”type” and ”colour” constitute keys, while ”Banana”, and ”Yellow” make up values. As they are enclosed by the curly brackets, both of these key-value pairs in turn, make up the value to the ”fruit1”-key.

not straight forward with structured data. Protocol Buffers were developed by Google with simplicity and performance in mind. Protocol Buffers are similar to xml, but binary, smaller and faster.

6.4.4 SQLITE3

Files with the suffix .sqlite3 or .db3 are usually single file sqlite3 databases. They contain structural information about the tables as well as the table data. A single database can contain multiple tables.

6.4.5 RDF

Resource Description Framwework is a general method for conceptual description or modelling of information. It is used in web resources and knowledge management applications.. It can use a variety of syntax notations including xml, json and turtle.

6.4.6 TXT

Plain text. On Microsoft Windows operating systems, a file is regarded as a text file if it has the suffix .txt. However, other suffixes are used for text files with specific purposes. For example, source code of computer programs is usually kept in text files that have file name suffixes indicating the programming language in which the source is written.

6.4.7 WAV

WAV is a container format that stores sound data losslessly in tagged chunks.

6.4.8 XML

XML stands for extensible markup language. It was developed for encoding documents in a format that is both human- and machine-readable. It supports arbitrary data structures. Many file formats using xml syntax have been developed, including docx, osm, svg and xlsx

7 Recommendations and Best Practices

As discussed in 5 using a public database is the best option to publish data in open science. We recommend using PostgreSQL, with the PostGIS extension for geographic information, as an open source and well documented database software. If a database is not available to you for storing or publishing your data, or setting up a database requires too many resources for your use-case, we provide short recommendations of file types to use based on a range circumstances at hand.

7.1 Geographic Data

If you are using geographic data locally, we recommend using GPKG. It is fast, small, well structured and supports both, vector and raster data. It is the standard file format in QGIS. If you need to be able to look at the raw text of vector data or use it in web applications, we recommend GeoJSON. We recommend to avoid shapefiles for the reasons described in 6.1.8. Since all relevant GIS software supports the alternative formats, the main reason to use shapefiles is legacy. If an existing system is built on shapefiles and no switch is planned or allowed, only then should you keep on using them.

7.2 Tabular data

The simplest and most widely supported format for tabular data is CSV. It is human- and machine readable, can be opened with all common spreadsheet software, is easy to convert and to load into a data base. If you are working with temporal data, we recommend to use columns for attributes and rows for time steps. The resulting file will be heavy on rows, which is good. When reading this data with a program or loading it into a database, handling many rows and adding new ones tends to be easier than doing so with new columns. Also, reading is easier when scrolling down as opposed to having to scroll sideways. The following lines represent an example excerpt of weather series in CSV format. The first line makes up the column labels.

```
date,max_temperature_celsius,precipitation_mm,precipitation_type
2020-01-01,3.5,0,NULL
2020-01-02,-2.1,4,"snow"
2020-01-03,-6.0,0,NULL
```

7.3 Temporal data

There is an ISO standard for time and date related data - ISO 8601. It is well defined, unambiguous across cultures, human- and machine readable. For example, the 31st of December 2019 is written as 2019-12-31. Adding the time two seconds before midnight, the term becomes 2019-12-31T23:59:58. Time zones in ISO 8601 are represented as local time, as Coordinated Universal Time (UTC), or as an offset from UTC. If no UTC relation information is given with a time representation, the time is assumed to be in local time. If the time is in UTC, a Z directly after the time without a space may indicate this[6]. For other time zones the delay is given as +/-hh:mm.[7]

7.4 Metadata

For data describing the structure or content of other data (metadata), we recommend the use of the json file type. It is both machine and human readable. Unlike simple tabular formats, it allows for nested structures.

Acknowledgements

This work was supported by grant 03ET4057A–D (SzenarienDB) from the Federal Ministry for Economic Affairs and Energy of Germany (BMWi).

License

Please cite as: ”Briefing: File Formats in Climate and Energy System Modelling”. Fraunhofer IEE, Öko-Institut, Otto von Guericke Universität Magdeburg, Reiner Lemoine Institut. CC BY 4.0.
doi: https://doi.org/10.5281/zenodo.3957445

[6] Z stands for an offset of zero

[7] `https://www.loc.gov/standards/datetime/iso-tc154-wg5_n0038_iso_wd_8601-1_2016-02-16.pdf`