



Interoperable CSV for Environmental Data Archival and Exchange: iCSV

RESEARCH PAPERS

PATRICK LEIBERSPERGER 

MATHIAS BAVAY 

IONUT IOSIFESCU ENESCU 

CHASE NÚÑEZ 

[*Author affiliations can be found in the back matter of this article](#)

 ubiquity press

ABSTRACT

Environmental research relies on seamless data exchange between institutions globally, but inadequate documentation and complex formats hinder collaboration. We introduce interoperable CSV (iCSV), a self-describing, human-readable format that combines the simplicity of Comma-Separated Values (CSV) with the metadata richness of NetCDF(Network Common Data Format)/CF (Climate and Forecast). iCSV ensures long-term interpretability, interoperability, and user accessibility, addressing key challenges in environmental data stewardship. By embedding structured metadata directly in a human-readable text file, iCSV enables automated validation, supports FAIR principles, and lowers the barrier to data sharing and reuse while ensuring data remains interpretable for future users and maintaining broad compatibility with existing software. This work motivates the need for a simple, self-describing tabular format for environmental time series, presents the iCSV specification, positions it within existing binary and human-readable format ecosystems through comparative analysis, and discusses current limitations with directions for future improvements.

CORRESPONDING AUTHOR:

Mathias Bavay

Institute for Snow and
Avalanche Research (SLF),
Davos, CH

bavay@slf.ch

KEYWORDS:

Earth Science; FAIR;
Interoperability; Metadata;
NetCDF

TO CITE THIS ARTICLE:

Leibersperger, P., Bavay, M.,
Enescu, I.I. and Núñez, C. 2026
Interoperable CSV for
Environmental Data Archival
and Exchange: iCSV. *Data
Science Journal* 25: 39,
pp. 1–19. DOI: [https://doi.org/
10.5334/dsj-2026-039](https://doi.org/10.5334/dsj-2026-039)

1 INTRODUCTION

Addressing the fundamental environmental challenges of our time—climate change (Gupta et al., 2023), biodiversity loss (Butchart et al., 2010; Rixen et al., 2014), and extreme weather—requires integrating vast and heterogeneous datasets collected across ecosystems and temporal scales (Kitchin, 2014; Zhang et al., 2023). These data are distributed across institutions, countries, and individual researchers (Plantin, Lagoze, Edwards, 2018), yet the collaborative analyses they demand are routinely impeded by inadequate documentation and unnecessarily complex file formats (Romano, 1997) (Figure 1). Seamless data sharing between institutions worldwide is therefore essential (Key et al., 2016), but achieving it depends on adopting standardized, self-describing formats with rich metadata.

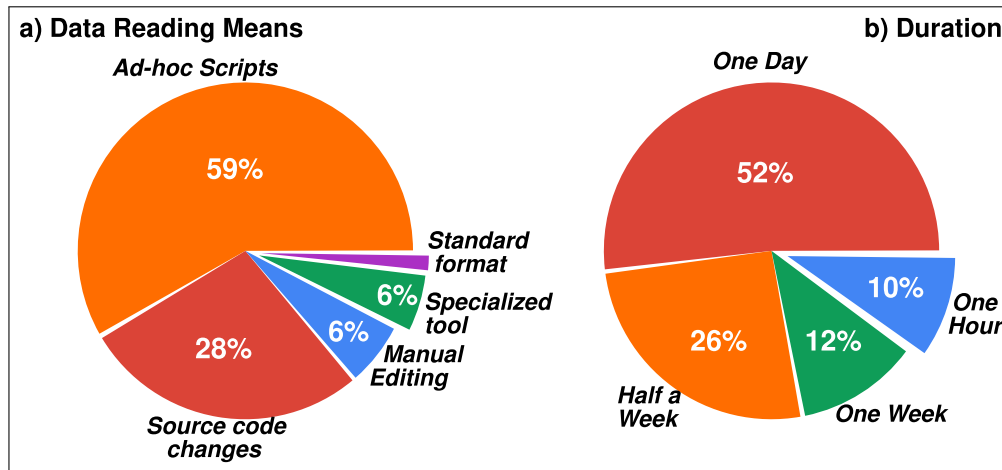


Figure 1 Data reading methods used by researchers to interpret third-party datasets, highlighting the prevalence of ad-hoc scripts and manual editing as well as the associated time investment. Fifty-four participants took part in this survey between fall 2019 and January 2020. Cited from (Bavay, Fiddes, Godøy, 2020).

Data incompatibility is not a theoretical challenge; it routinely impedes collaboration (Kirchhoff, Gottschalk, Engels, 2022) and will worsen as open science grows. As datasets are increasingly preserved in online repositories, researchers may gain access to the files long past when the original data creator is able to explain their structure and meaning. This makes the case for standardized, self-describing formats in environmental science especially compelling (Petti and Karev, 2022).

The urgency is even greater when considering long-term data stewardship (Tenopir et al., 2020). Environmental time series, which often span decades, are irreplaceable scientific assets that must remain interpretable despite rapid shifts in software, hardware, and research practices. As data volumes continue to grow, manual curation is no longer feasible. Durable archival instead requires formats that combine consistent organization with rich contextual metadata, ensuring not only reliable preservation but also meaningful reuse for future generations of researchers.

Findable, Accessible, Interoperable, and Reusable (FAIR, (Wilkinson et al., 2016)) are the guiding principles of data sharing today. The ideal environmental data archive format should seamlessly integrate into workflows adhering to these principles and possess the following key characteristics:

- **Self-describing:** Future readers should be able to understand the contents of the file without additional software or knowledge.
- **Interoperable:** Allow integration into different scientific workflows.
- **Accessible:** Data stored in this format should be readable even when software/hardware ecosystems change over time.
- **Flexible:** To prevent having specific formats for each scientific field, flexibility is needed to accommodate different needs in respective sectors.

In addition to these core characteristics, other factors shape the design of a sustainable environmental data format. A format that is technically ideal but difficult to use will never see broad adoption and, without adoption, cannot be maintained over the long term. At the same time, the growing role of generative artificial intelligence (AI) and automated workflows underscores the importance of rigorous data provenance. A robust format should capture the full evolution of a dataset, either through its native structure or via integration with

version-tracking systems such as Git ([Chacon and Straub, 2014](#)). Recording this history enables independent verification, supports reproducibility by allowing exact reconstruction of prior states, and facilitates auditing of data lineage. Without such features, even well-structured datasets risk becoming opaque, difficult to validate, and irreproducible over time.

An additional, often underappreciated, advantage of embedding rich, structured metadata directly in a dataset is that the metadata becomes machine-actionable ([Batista et al., 2022](#); [Zuiderwijk, Jeffery, Janssen, 2012](#)). It can be read, interpreted, and converted into executable checks by software without human intervention. When field names, types, delimiters, missing-value markers, coordinate systems, timestamp semantics, and other contextual attributes are declared in-band, tools can automatically construct a validation schema (e.g., a Frictionless table schema ([Fagundes and Ribeiro Junior, 2020](#); [Petti and Karev, 2022](#); [Petti, 2023](#))), run syntactic and semantic checks (correct delimiter, required sections, ISO8601 timestamps, EPSG codes), and generate deterministic, machine-readable reports. In this way, metadata is transformed from a passive description into an active control surface for data quality ([Bratt et al., 2017](#)): repositories and pipelines can reject malformed submissions at ingest, continuous-integration (CI) systems can gate merges on compliance, and archival services can run periodic audits to detect bit rot or format drift.

Automated validation substantially reduces the manual effort typically required to interpret third-party datasets and directly improves reproducibility and trust. Structural checks (e.g., required data fields are present), field-level consistency checks (matching list lengths), type and range checks (numeric bounds, enumerations, timestamp formats), and profile-specific rules (such as 2D time-series step markers) can all be executed automatically once metadata are in place. The resulting validation artifacts—human-readable summaries, machine-readable JSON reports, and deterministic exit codes—integrate smoothly into automated workflows, dashboards, and downstream processing (including machine learning pipelines), while providing immediate, actionable feedback to data producers.

This paper formalizes and extends our earlier efforts (NEAD and SMET) into a general cross-domain specification that balances ease of use with machine-level interoperability. It sets out clear requirements and recommendations while allowing flexibility to support broad adoption across the scientific community. We introduce iCSV, a self-describing (the data and metadata are sufficient to interpret the file without external documentation), human-readable format (a file can be interpreted by humans without specialized software) that combines the simplicity of Comma-Separated Values (CSV) with the metadata richness of NetCDF/CF. iCSV ensures long-term interpretability and interoperability, addressing key challenges in environmental data stewardship. In this paper, we demonstrate this by reviewing related data formats, describing the specification in detail, and presenting examples and use cases. Together, these illustrate how iCSV can serve as a valuable tool for environmental researchers—fostering collaboration, accelerating scientific progress, and enriching our understanding of the environment.

2 MOTIVATION

To understand the need to create a new file format, we examine existing formats, their limitations, and advantages. We show the evolution from plain CSV to NEAD, the predecessor, and the changes that came with each file format ([Figure 2](#)). Although iCSV is American Standard Code for Information Interchange (ASCII)-based, there are several well-known binary formats with design goals and approaches that were incorporated and need to be reviewed.

2.1 GRIB AND BUFR

Gridded Binary or General Regularly-distributed Information in Binary form (GRIB) and Binary Universal Form for the Representation of meteorological data (BUFR) ([Manual on codes. International codes. volumes I.2 & 1.2, 2009](#); [Yang et al., 2019](#)) were developed to facilitate real-time meteorological data exchange within the World Meteorological Organization (WMO) network. The key design goals for these formats were efficient data storage and flexibility to allow adoption across national borders and scientific fields ([Karhila and Hernandez-Carrascal, 2010](#)). These goals were achieved by using a binary format that supports various compression algorithms, depending on the use case, and by including metadata with the data. Since the

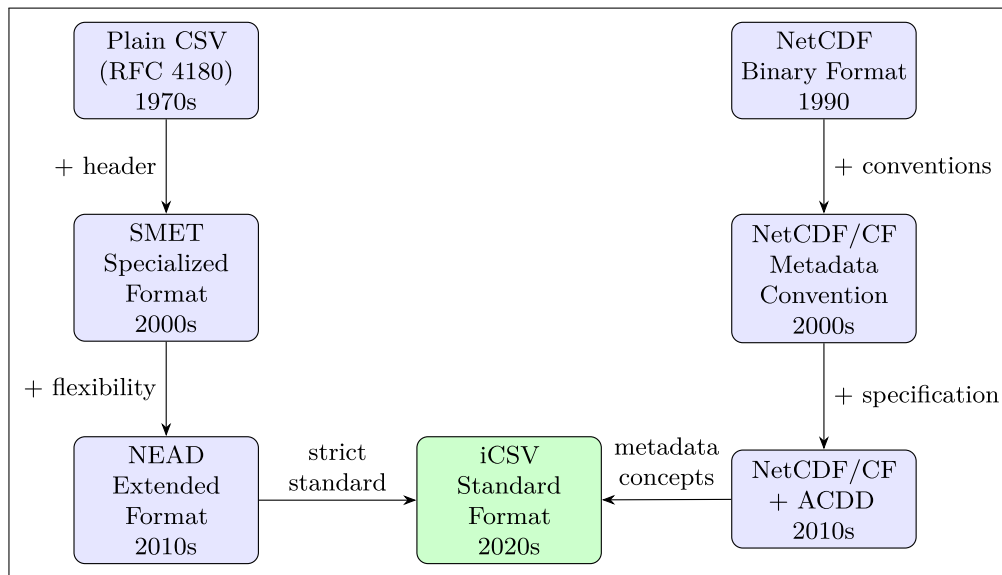


Figure 2 Evolution of environmental data formats leading to iCSV. The diagram shows how iCSV combines the human-readable aspects of CSV-based formats with the rich metadata framework inspired by NetCDF conventions. Each transition in the CSV formats adds key improvements addressing limitations of previous formats.

metadata is compressed together with the data, the sequences and format that were used need to be known to decode the files. This is achieved by providing tables that contain all the necessary information. These tables can be extended by anyone, as long as they provide the corresponding table. To read and write GRIB or BUFR files, extensive knowledge and specialized software are needed, making navigating, reading, and writing those files very cumbersome. Additionally, the flexibility to create custom tables leads to metadata that sometimes cannot be fully decoded anymore, in cases where creators do not make tables publicly accessible or the tables are simply deleted. In the end, the flexibility to change parts of the format is needed to allow wide adoption, but reliance on external data (not in the file itself) to understand a file is a significant problem (Bergès, 2002; Caron and Oxelson, 2013).

2.2 NetCDF

NetCDF (Network Common Data Format) offers a powerful approach for storing multidimensional scientific data (Rew and Davis, 1990). This self-contained, open-source format bundles the data itself alongside crucial metadata like variable names, units, and data types within a single file. The format was developed at Unidata, part of the University Corporation for Atmospheric Research (UCAR), and has become a standard in atmospheric and oceanic sciences.

NetCDF employs byte order information within the header to ensure readability across different computing architectures, addressing the endianness problem that plagued many earlier binary formats. The current versions (NetCDF-4) leverage the Hierarchical Data Format (HDF5) for enhanced features such as compression, chunking, and efficient data access (Folk et al., 2011). These features make NetCDF particularly well-suited for large-scale climate modeling, remote sensing, and geophysical applications where data volumes can reach terabyte scales.

The NetCDF format benefits from an extensive and mature software ecosystem, with robust support across programming languages (Michna and Woods, 2013), analysis tools, and visualization platforms. Libraries exist for nearly all major programming languages, and numerous scientific applications have built-in NetCDF support. This broad adoption has established NetCDF as a cornerstone format for scientific data exchange within many research communities (Hankin et al., 2009).

NetCDF offers a great amount of flexibility; it can contain any number of dimensions, images, and any desired metadata. It offers a robust data storage solution; however, additional conventions are needed to constrain its structure for specific scientific communities. To address this limitation, the scientific community developed complementary metadata conventions that work with NetCDF files. The Climate and Forecast (CF) convention, initially designed for gridded data from climate and forecast models, has become a widely adopted standard across various fields, including numerical modeling and geosciences (Hassell et al., 2017). The CF convention establishes a standardized metadata framework for comprehensive data

descriptions, including a standard name table to ensure consistent identification of physical quantities. It operates through an open forum, facilitating continuous updates and revisions based on the evolving needs of the scientific community. Although there are clear descriptions of the conventions, they are not necessarily easy to understand or quite exhaustive. This leads to misunderstandings and generally poor standard compliance, causing problems in sharing and accessing NetCDF data.

Complementing the CF convention, the Attribute Convention for Data Discovery (ACDD) ([Attribute Convention for Data Discovery 1-3, 2023](#)) provides additional metadata attributes specifically designed to enhance dataset discoverability in digital libraries and data catalogs. ACDD builds upon and is compatible with both the NetCDF User Guide recommendations and CF conventions, adopting all CF global attributes while adding discovery-oriented metadata such as creator information, geographic coverage, and temporal extent. This convention is particularly valuable for data repositories and discovery systems that need to extract metadata for cataloging and search purposes.

Together, CF and ACDD conventions transform NetCDF from a simple binary storage format into a comprehensive, self-describing data container that promotes both scientific interoperability and data discoverability. The CF convention ensures that the scientific content is properly described with standardized vocabulary and units, while ACDD provides the discovery metadata necessary for efficient data management and sharing.

However, NetCDF still has two primary limitations. First, as a binary format, it lacks human readability. Data cannot be examined directly without specialized software, making quick inspection and validation more challenging. Second, despite robust software support, there is a learning curve associated with using NetCDF effectively. Users must install dedicated libraries and become familiar with specific APIs to access the data, which can present a barrier to entry for new users or in scenarios requiring rapid ad hoc analysis.

The scientific community often expresses reservations regarding binary formats like NetCDF, particularly during initial validation, analysis, and processing steps where human-readable formats like CSV are favored. This preference is especially pronounced when sharing data with collaborators who may be unfamiliar with NetCDF or when performing quick exploratory analyses. This underscores the ongoing need for an interoperable data format that is human-readable while still maintaining rich metadata capabilities.

When referring to NetCDF metadata throughout this paper, we specifically mean NetCDF files that follow both CF and ACDD conventions (NetCDF/CF-ACDD), representing the current best practices for comprehensive scientific data description and discoverability.

2.3 CSV

CSV files offer a user-friendly, human-readable alternative to binary data formats, without almost any learning curve. This row-based format stores each data record on a separate line, with values separated by commas. Despite its ubiquity, CSV was only formally standardized in 2005 through RFC 4180 ([Shafranovich, 2005](#)), although various implementations often deviate from this standard. The specification is intentionally minimal, providing only basic guidelines:

- Each record appears on a separate line, terminated by a line break (CRLF).
- The last record may or may not have an ending line break.
- An optional header line appears in the same format as normal record lines.
- Within the header and each record, there may be one or more fields separated by commas.
- Fields containing line breaks, double quotes, or commas should be enclosed in double quotes.
- A double quote within a field must be represented by two double quotes.

This minimalist approach has contributed to CSV's widespread adoption across diverse computing environments ([Boehm, 2025](#)). CSV files can be easily created, edited with simple text editors, and processed by virtually any programming language or data analysis tool. Their text-based nature also makes them resilient to technological changes—CSV files created

decades ago remain readable today without specialized software (Hart et al., 2016). However, the very aspects that contribute to CSV's ease of use also introduce significant limitations for long-term data archiving and interoperability (Mitlöhner et al., 2016). The lack of mandatory metadata, as well as naming schemes for the column names, means that crucial information about units, coordinate systems, measurement methods, and quality flags must be stored separately or communicated through external documentation. Even the delimiter itself is inconsistently implemented across different applications, with some using semicolons, tabs, or other characters instead of commas (Hameed et al., 2024). This minimal standardization can lead to what might be called a “chaotic data environment” where plain CSV files become challenging to interpret without prior knowledge of their contents (van den Burg, Názábal, Sutton, 2019). Data exchange effectively becomes dependent on knowing what is stored beforehand or having access to supplementary documentation, which may become separated from the data over time. These limitations become particularly problematic in scientific contexts where precise interpretation of data is essential.

2.4 CSV-BASED STANDARDS

To mitigate the inherent ambiguities of plain CSV, numerous use-case-specific format specifications have emerged. Some notable examples are the IAGA-2002 Data Exchange Format (IAGA 2002 Data Exchange Format, 2026), the BADC-CSV format (Parton and Pepler, 2009), and the SMET format (Bavay, 2021). The BADC-CSV and SMET formats are highlighted in more detail, the latter being a precursor to iCSV and the former being a closely related approach to combining tabular ASCII data with structured metadata.

2.4.1 BADC-CSV

The BADC-CSV Format for Data Exchange (Parton and Pepler, 2009) was developed at the British Atmospheric Data Centre (BADC) as an ASCII-based submission format that retained the accessibility of CSV while adding metadata needed for archiving and reuse. Its design reflects the familiarity of many users with spreadsheet software and common analysis tools, while drawing on established metadata practices from formats and conventions such as NASA-Ames and CF.

BADC-CSV is conceptually close to iCSV in its structure. A file begins with a format identifier and a metadata header, followed by a tabular data section delimited by explicit data markers. Metadata entries are associated either with the complete file or with individual data columns, allowing dataset-level and variable-level metadata to coexist in the same text file.

BADC-CSV therefore demonstrates that self-describing CSV-like formats can improve the long-term interpretability of environmental datasets without abandoning human readability. Its limitations are mainly structural and practical rather than conceptual. The format is intended for simple one-dimensional datasets, which restricts its suitability for more complex data structures. In addition, because metadata and data markers are part of the same un-commented text stream as the tabular data, BADC-CSV files are not directly readable by standard CSV parsers without preprocessing. The shared metadata header can also make authoring and validation more demanding as files become more complex. Finally, there is no formal extension mechanism for BADC-CSV, which limits its extensibility. These limitations motivate iCSV's stricter separation between metadata and data, its explicit field model, and its focus on reusable tooling.

2.4.2 SMET

The Station METeorological (SMET) format offers a structured approach for storing meteorological station observation data (Bavay, 2021). Developed by the WSL Institute for Snow and Avalanche Research SLF, SMET was designed specifically to address the need for consistent data formatting, and metadata that make a file self-describing, in environmental science applications, particularly for meteorological and snow measurement data.

According to the official SMET specifications (version 1.2), the format was created with specific requirements in mind: it should contain both data and metadata in a self-describing manner, support an arbitrary number of meteorological parameters for any number of time steps, accommodate variable timesteps without a fixed sampling rate, be easily parsed with common

programming languages and tools, allow preview with standard analysis software without preprocessing, maintain user-friendly column alignment, be extensible, and serve a wide range of applications.

The structure of a SMET file consists of several key components:

- **Signature line:** The first line of the file begins with “SMET” followed by a version number and the encoding (ASCII).
- **Header section:** Marked by “[HEADER]” and contains key-value pairs defining metadata such as station location (latitude, longitude, altitude), timezone, and measurement parameters.
- **Fields definition:** Includes mandatory “fields” key listing column names and optional “units_offset” and “units_multiplier” for value transformations.
- **Data section:** Marked by “[DATA]” and contains the actual measurement data in a tabular format.

SMET utilizes a whitespace-delimited tabular data section (spaces and/or tabs), promoting clear column separation and enhanced readability for humans. This design choice was deliberately made to facilitate easy viewing and manipulation with common tools like R, gnuplot, Excel, and Matlab, as well as manual editing in text editors. The format also supports compressed (gzipped) files and even offers a binary version for specialized applications, which has not seen much use in practice and is of secondary concern.

The format specification enforces certain constraints, such as requiring timestamp as the first field (in ISO format), and defines standard parameter names like TA (air temperature in degrees Celsius), RH (relative humidity in percent), VW (wind velocity in m/s), and ISWR (incoming shortwave radiation in W/m²). This standardization is particularly valuable in the snow and avalanche research community, where consistent data formatting is essential for reliable model inputs.

While SMET was an important advancement over plain CSV by introducing structured metadata, its specific requirements tailored to meteorological station data limit its applicability to other environmental science domains.

2.5 NEAD

The Non-binary Environmental Archive Data (NEAD) (2024) format represents an evolutionary step forward from SMET, building upon its structured approach while offering greater versatility for environmental science data archiving. Developed to address some of the limitations of SMET while maintaining its advantages, NEAD aims to provide a more flexible framework suitable for diverse environmental data types beyond meteorological station observations.

Similar to SMET, NEAD employs a dedicated header structure that is clearly divided into two main sections:

- **Metadata section:** Stores information about the entire dataset, including spatial reference, temporal coverage, and provenance information.
- **Fields section:** Details the variables contained within the file, including their names, units, and interpretative information.

To facilitate clear distinction between the header and data sections, each header line is prefixed with a “#” character. This consistent marking makes it straightforward for both humans and software to identify metadata elements. A significant improvement over SMET is that the data section in NEAD uses a configurable delimiter specified in the metadata (typically a comma, semicolon, or pipe character), rather than being restricted to tabs and whitespace. This approach combines the flexibility of CSV with the structured metadata of more complex formats.

Unlike the more restrictive SMET format, which was designed specifically for meteorological data, NEAD enforces fewer domain-specific requirements, promoting flexibility and broader applicability. This adaptability allows NEAD to be effectively used across a wide range

of environmental science disciplines, from hydrology and soil science to ecology and atmospheric measurements.

NEAD laid the foundation for iCSV by combining human-readable tabular data with structured metadata in a single file. To encourage broader adoption, the specification was formalized, field-specific requirements and ACDD metadata were added, supporting software tools were developed, and the format was renamed iCSV to emphasize its closeness to CSV.

3 THE iCSV FORMAT

While each of the formats described above has strengths, and there are some existing formats that partially address these needs, none fully address the need for a backward-compatible, human-readable, self-describing, and interoperable format that balances simplicity with metadata richness and extensibility (Table 1). iCSV fills this gap by combining the best aspects of CSV (human-readable) and NetCDF/CF (rich and structured metadata in the same file as the data) into a single, standardized solution. The NEAD format served as an important stepping stone toward the development of iCSV, but it lacked a formal specification. The iCSV format addresses these limitations through standardization and community engagement. Community engagement is supported through issue tracking at <https://code.wsl.ch/EnviDat/iCSV>, while the evolving specification is documented here and updated regularly online.

FORMAT	HUMAN-READABLE	SELF-DOCUMENTING	STANDARD METADATA	FILE SIZE	SOFTWARE REQUIREMENTS
CSV (plain)	✓	×	×	×	✓
NetCDF	×	✓	×	✓	✓
NetCDF/CF	×	✓	✓	✓	✓
GRIB/BUFR	×	×	✓	✓	(✓)
SMET	✓	✓	✓	×	×
NEAD	✓	✓	✓	×	(✓)
iCSV	✓	✓	✓	×	✓

Table 1 Comparison of environmental data formats. x = poor/none, ✓ = great/yes, () = partly. Human-readability refers to the ability to interpret the data without specialized software. Self-documenting indicates whether only the file itself is enough to understand the content of the data. Standard metadata refers to support for established metadata conventions including units (e.g., meters, degrees Celsius, kg/m³). File size evaluates storage efficiency. Software requirements considers the ease of implementing readers/writers for the format, and existing supported software.

Terminology. To support consistency across the format, we adopt the following terminology: the *metadata* section contains file- and dataset-level attributes; the *fields* section describes the tabular columns (names and attributes); the *data* section contains the tabular records themselves. An *application profile* constrains and/or extends the base iCSV rules for a specific domain use case. We use *self-describing* and *human-readable* consistently to emphasize interpretability without specialized software.

3.1 FORMAT SPECIFICATION

While creating a file may require knowledge of the specification, reading it should remain possible in the future without consulting external documentation. Generally speaking, an iCSV file consists of three distinct sections: the metadata section, the fields section, and the data section. Each section serves a specific purpose in ensuring data is well-documented and interoperable.

An iCSV file begins with a dedicated header line (Listing 1) that identifies the file format, its version number, and the character encoding used. Currently, version “1.0” is the only available version, and UTF-8 encoding is required for all files. Optionally, an application profile name can be specified in all capital letters to indicate adherence to a specific domain profile. The first line follows this structure:

```
1 # iCSV 1.0 UTF-8 {APPLICATION PROFILE}
```

Listing 1 iCSV Format Identification Line.

The definition of metadata in the header uses a key-value pair structure (Listing 2), with each pair occupying a distinct line. Header lines begin with a ‘#’ symbol followed by the key, assigned to its value using an ‘=’ symbol. This format results in header lines as:

```
1 # key = value
```

Listing 2 iCSV Format Identification Line.

3.1.1 Metadata Section

The first header section following the identification line is the metadata section, marked by the header line “# [METADATA].” This section contains information about the dataset’s origin, location, and other important context.

Required keys. This section must contain, at minimum, the following key-value pairs:

- **field_delimiter:** Specifies the delimiter used to separate columns and values within the data section, as well as within variable descriptions in the fields section. Valid delimiters include comma (,), semicolon (;), colon (:), forward slash (/), backslash (\), and pipe (|).
- **srid:** Identifies the EPSG code of the reference coordinate system used to geolocate the data’s validity. It should be formatted as “EPSG:xxxx” (where “xxxx” represents the valid EPSG code). For example: “EPSG:4326.”
- **geometry:** Specifies the spatial reference or origin of the data. Valid options include:
 - Well-Known Text (WKT) string ([Herring, 2011](#))
 - A location specified in the format “POINT(x:float y:float)”
 - “POINTZ(x:float y:float z:float)” for locations including altitude data
 - The name of the column containing this information (for moving sensors)

Recommended keys. In addition to the required metadata keys, we strongly recommend incorporating any applicable CF metadata. Several metadata keys are particularly useful for various applications, and recommended for use in an iCSV:

- **station_id:** Stores an alphanumeric string identifying the station, satellite, or other platform from which the data was observed.
- **nodata:** Defines a value representing missing data points within the dataset. It can be either an integer or a floating-point number.
- **timezone:** Specifies the time zone applicable to timestamps used in the data. It can be represented as an integer, a floating-point number, or a time zone string.
- **doi:** A Digital Object Identifier (DOI) to uniquely identify the dataset.
- **timestamp_meaning:** Clarifies the meaning of timestamps present in the data. Valid options include “beginning,” “end,” “middle,” “instantaneous,” “other,” “average,” or “undefined.”
- **ACDD:** Allows incorporating any relevant metadata adhering to the Attribute Convention for Data Discovery (ACDD) standards.

By including these and other applicable CF metadata parameters, researchers can ensure the completeness, interoperability, and usability of their datasets across long time periods, national borders, and scientific domains. Any additional information that needs to be shared, such as file history or detailed description, can be included in the metadata section with an appropriate keyword.

To illustrate the metadata section format ([Listing 3](#)), consider a station located at the Weissfluhjoch Measurement Site in Davos. This station has the following attributes:

```
1 # iCSV 1.0 UTF-8
2 # [METADATA]
3 # field_delimiter = ,
4 # geometry = POINTZ(781957.771521 185571.662370 1556.700000)
5 # srid = EPSG:21781
6 # station_id = Wfj
7 # nodata = -999
8 # timezone = +1
9 # creator = John Doe
```

Listing 3 Example of a metadata section for a weather station.

3.1.2 Fields section

This subsection defines how to declare the tabular variables (columns) and their attributes.

The fields section follows the metadata section and defines the variables stored within the data. It starts with the header line: “# [FIELDS].” Each attribute must be a list with values separated by the delimiter defined in the metadata section. The length of the list must match the number of columns in the data section.

The mandatory key in this section is **fields** (Listing 4). It contains a list of column names present in the data section. For example, a meteorological station might have columns for timestamp, total precipitation (PSUM), snow height (HS), and air temperature at 2 meters (TA).

Listing 4 Example of a Fields Section with the Mandatory fields Key.

```
1 # [FIELDS]
2 # fields = timestamp,PSUM,HS,TA
```

While the mandatory information provides a basic understanding of the data, including additional explanatory keys can significantly enhance clarity and interoperability. We recommend incorporating the following keys:

- **units:** The measurement units for each column (e.g., meters for snow height, degrees Celsius for air temperature).
- **standard_name:** Defined by the CF convention, this key ensures consistent identification of physical quantities across different datasets.
- **long_name:** Provides a comprehensive and user-friendly description of each variable, eliminating the need for further reference (e.g., “Air temperature measured 2 meters above ground”).
- **units_offset:** If a constant value has been added or subtracted from the data values, this key specifies the offset required to obtain the original values. Applied after multiplier.
- **units_multiplier:** Indicates a multiplier necessary to convert the stored values to the actual units. Applied before offset.
- **timestamp_meaning:** As before, used if the timestamp format varies between variables.

Here is an expanded example (Listing 5) including some of the recommended fields information:

Listing 5 Example of a Fields Section with Recommended Keys.

```
1 # [FIELDS]
2 # fields = timestamp,PSUM,HS,TA
3 # units = ISO8601,mm,cm,degC
4 # standard_name = timestamp_iso,total_precipitation,snow_height,air_temperature
5 # long_name = Time of measurement,Total precipitation,Snow height,
   Air temperature at 2m
```

3.1.3 Data section

This subsection defines how to represent tabular records with the agreed delimiter and field order.

The header section concludes with a line denoting the start of the data section: “# [DATA].” This line is followed by the actual data, organized in rows with columns separated by the predefined field delimiter (Listing 6). After this marker, lines beginning with “#” are not permitted, except in well-defined cases such as 2D Time Series structures.

The data values in each row must match the number of fields defined in the fields section. Whitespace is allowed between values and delimiters to improve readability in text editors.

Listing 6 Example of a Fields Section with Recommended Keys.

```
1 # [DATA]
2 2005-08-23T15:30:00,0.0,50,8.94
3 2005-08-23T16:30:00,0.0,47,9.05
4 2005-08-23T17:30:00,0.2,42,8.76
```

An example of a complete iCSV file is shown in Listing 7.

3.2 APPLICATION PROFILES

Since the format specification defines relatively few strict rules, extensions can be introduced for specific use cases that require additional constraints. Such *application profiles* may specify stricter metadata requirements or alternative data organizations—for example, datasets stored column-major, i.e., variables in rows and entries in columns, indexed by time, or indexed by location. Profiles allow standards for these specialized cases without cluttering the base format. Every application profile MUST be available long term, and should preferably be included in the iCSV repository. The format is designed in a way, such that even missing specifications on an application profile do not inhibit readability or interpretability. Rather, application profiles are intended as a guideline on how to write standardized metadata in a particular domain to promote uniformity.

Application profiles are therefore primarily intended for use cases that require conventions beyond those provided by the base format. Many datasets can already be represented using its inherent flexibility. For example, when the set of recorded variables changes over time, a fixed set of columns can be retained, with nodata values indicating periods when a variable is not recorded. Similarly, records from different locations can be distinguished using a location column, with nodata values for variables that do not apply at a given location. Similarly, long data formats can be represented as well. Variables that are computed from a varying amount of measurement locations currently cannot be represented, and thus would need an application profile.

The complete format specification, including updates, application profiles, and implementation examples, can be found at the official iCSV website: <https://git-pages.wsl.ch/envidat/icsv/>. The specification is maintained as an open resource and continues to evolve through community feedback (via GitLab) and implementation experience across different environmental science domains.

Listing 7 Structure of an iCSV file showing the three main sections: METADATA (top), FIELDS (middle), and DATA (bottom). This example shows a meteorological time series with various measurements.

```

1 # iCSV 1.0 UTF-8
2 # [METADATA]
3 # field_delimiter = |
4 # geometry = POINTZ(781957.771521 185571.662370 1556.700000)
5 # nodata = -999.000000
6 # srid = EPSG:21781
7 # station_id = ANU564
8 # timezone = 1.000000
9
10 # [FIELDS]
11 # fields = timestamp|DW|ISWR|P|PSUM|RH|TA|VW_U|VW_V|VW_MAX
12 # units_offset = 0|0|0|0|0|0|0|0|0|0|0
13 # standard_name = timestamp_iso|wind_direction|short_wave_incoming_radiation|...
14
15 # [DATA]
16 2005-08-23T15:30:00|-999|50|83700|0|0.889|281.94|1.5|-999|2.8
17 2005-08-23T16:30:00|-999|47|83700|0|0.913|282.05|1.21|-999|3.2
18 2005-08-23T17:30:00|-999|42|83700|0|0.897|281.76|1.39|-999|3.5

```

4 USE CASES

One application profile that is already in use and was developed for the need of an easy-to-read format for the output of numerical snow cover models is the 2D Time Series profile.

4.1 2D TIME SERIES

Generally, the base iCSV format would already allow 3D, or even XD, data to be stored by having a column for time and a column for geometry/position, so that there can be multiple positions for the same time point and vice versa. Using the WKT representation (Herring, 2011) for geometry, this is not limited to geographic positions but can also include complex geometries, with using location as a variable, or specifying metadata in a column. However, this type of storage can be unreadable when opened and inspected with a text editor. It will also be harder to correctly assemble a time series of vertical profiles, for example, requiring additional effort by the user. This is why the 2D Time Series profile was developed; it aims at vertical (or horizontal)

profiles over time. Specifically, this is achieved by including additional header lines to mark different time steps in the data section.

Rules specific to 2D Time Series:

- The profile token in the identification line must be `2DTIMESERIES`.
- Inside the data section, time steps are marked by header lines of the form `# [DATE=YYYY-MM-DD Thh:mm:ss]` (ISO 8601).
- Comment lines beginning with `#` are permitted inside the data section only to indicate time-step markers for this profile.
- The delimiter, field order, and the number of values per row follow the base rules; each data row must match the number of fields declared in the fields section.

This is demonstrated in the example below (Listing 8):

```

1  # iCSV 1.0 UTF-8 2DTIMESERIES
2  # [METADATA]
3  # field_delimiter = |
4  # geometry = POINTZ(781957.771521 185571.662370 1556.700000)
5  # nodata = -999.000000
6  # srid = EPSG:21781
7  # source = ExampleModel v1.0.0
8
9  # [FIELDS]
10 # fields = layer_index|T|SSA|RHO
11 # units = cm|degC|kg m-3|mm
12 # units_offset = 0|0|0|0
13
14 # [DATA]
15 # [DATE=2005-08-23T15:30:00]
16 6|273.15|-999|-999
17 5|-999|35|250
18 4|-999|36|280
19 3|-999|37|300
20 2|-999|37|300
21 1|-999|37|300
22 # [DATE=2005-08-23T16:00:00]
23 6|274.15|-999|-999
24 5|-999|35.5|280
25 4|-999|37|290
26 3|-999|37|300
27 2|-999|37|300
28 1|-999|37|300

```

Listing 8 Example of the 2D time series extension of the iCSV format. This variant allows for storing vertical profiles over time, such as snow profile evolution, soil layer properties, or atmospheric soundings. Note how the DATA section includes time markers (`# [DATE=...]`) that separate different timesteps.

4.2 REAL WORLD EXAMPLES

In some cases, iCSV, including the 2D Time Series profile, is already in use. For meteorological weather stations, iCSV is effective in bringing order to logger output and CSV-based data. An example of this is (Listing 9) from the station at Weissfluhjoch, Davos, Switzerland:

```

1  # iCSV 1.0 UTF-8
2  # [METADATA]
3  # field_delimiter = ,
4  # geometry = POINTZ(780862.827841 189223.271981 2535.000000)
5  # nodata = -999.000000
6  # srid = EPSG:21781
7  # station_id = WFJV
8  # timezone = 1.000000
9  # [FIELDS]
10 # fields = timestamp,DW,RH_ROT,RH_THY,TA_ROT,TA_THY,TSS,VW,VW_MAX
11 # [DATA]
12 2025-12-09T10:00:00,129,0.248,0.268,4.038,3.78,-5.62,1.4,4.2
13 2025-12-09T10:10:00,128,0.254,0.294,4.173,3.61,-5.18,1.9,3.6

```

Listing 9 Example of iCSV file from Weissfluhjoch, Switzerland. ACDD headers have been omitted to keep the example short.

Another example is for the Output of SnowModels (Listing 10), which are notoriously hard to read. Output from the well-used SNOWPACK model can be easily converted for archiving and post-processing. An example of such a file is shown below:

Listing 10 Example of iCSV file from SNOWPACK model.

```

1 # iCSV 1.0 UTF-8 2TIMESERIES
2 # [METADATA]
3 # nodata = NaN
4 # model = Helmut v0.1.0
5 # field_delimiter = |
6 # geometry = POINT(46.831 9.81)
7 # srid = EPSG:4326
8 # [FIELDS]
9 # units = -|m|W m-2|s|K|m2 kg-1|kg m-2|Pa s|kg m-2|1|W m-2|W m-1 K-1|kg m-2|K m-1|m|
  J|1|m3 m-3|m3 m-3|kg m-3|kg m-3|1
10 # standard_name = -|thickness|absorbed_shortwave_radiation|layer_deposition_time|
  temperature|specific_surface_area|mass_per_unit_area|viscosity|percolating_mass|
  layer_type|surface_melt|thermal_conductivity|liquid_mass|temperature_gradient|
  grain_size|enthalpy|percolating_water_fraction|volumetric_ice_content|
  volumetric_water_content|density|dry_density|sphericity
11 # fields = layer_index|Z|absorbed_shortwave_radiation|layer_deposition_time|
  temperature|SSA|M|viscosity|percolating_mass|layer_type|surface_melt|
  thermal_conductivity|liquid_mass|temperature_gradient|grain_size|enthalpy|
  percolating_water_fraction|volumetric_ice_content|volumetric_water_content|density|
  dry_density|SPH
12 # [DATA]
13 # [DATE=1995-11-01T21:30:00]
14 1|NaN|NaN|NaN|273.05000000000007|NaN|NaN|NaN|NaN|skin|0.0|NaN|0.0008187406296851576|
  NaN|NaN|273.05000000000007|NaN|NaN|NaN|NaN|NaN|NaN
15 2|NaN|NaN|NaN|273.15099999999995|NaN|NaN|NaN|NaN|skin|0.0|NaN|0.0008190434782608695|
  NaN|NaN|273.15099999999995|NaN|NaN|NaN|NaN|NaN|NaN
16 # [DATE=1995-11-02T00:30:00]
17 1|NaN|NaN|NaN|273.05|NaN|NaN|NaN|NaN|skin|0.0|NaN|0.0008187406296851574|NaN|NaN|
  273.05|NaN|NaN|NaN|NaN|NaN|NaN
18 2|0.02808775301864211|NaN|8.152659e8|271.00822769882694|64.99903458943011|
  3.539991370913109|7.400691532747515e7|0.0|snow|0.0|0.048208870856237804|0.0|
  63.342769603368325|0.0|-15967.387608475383|0.0|0.1374408619328104|0.0|
  126.03327039238714|126.03327039238714|0.4973432382173404
19 3|0.0039650454454901165|NaN|8.152722e8|268.9779146705702|65.06510376074726|
  0.47900137144965926|7.827944997073232e7|0.0|snow|0.0|0.04562609717379791|0.0|
  63.34276960336896|0.0|-4208.703256231531|0.0|0.13174048314263798|0.0|
  120.80602304179902|120.80602304179902|0.49775210749458565
20 4|NaN|NaN|(*NaN|268.657301023589|NaN|NaN|NaN|NaN|skin|0.0|NaN|0.0008055691185115113|
  NaN|NaN|268.657301023589|NaN|NaN|NaN|NaN|NaN|NaN

```

4.3 AVAILABLE SOFTWARE

In addition to the file format being useful in itself, there are already a few software solutions to handle iCSV files. First, since the data is stored as plain CSV, any software that can parse tabular data will also be able to parse iCSV files, such as pandas, R, and Excel. Sometimes the field delimiter and the comment character will have to be adjusted. Additionally, there is specialized software that simplifies reading and writing iCSV files. We showcase some examples that are the start of a growing ecosystem of tools. MeteoIO (Bavay and Egger, 2014) is a numerical pre-processing library in C++. The iCSV plugin provides fast read and write support (optionally with transparent compression and decompression), and allows iCSV as input to numerical models. Snowpat (Leibersperger, 2024) is a Python library using Pandas (Pandas, 2020) to handle various forms of data. Since SMET was already supported before the addition of iCSV, conversion between the two formats is easily possible. niViz is an interactive plotting tool, available as a web service or local JavaScript-based application. It can be used to visualize snow profiles and a variety of plots from either measured or simulated snow profiles including auxiliary data (such as meteorological time series) (Fierz et al., 2016).

DEVO (Data Enrichment and Validation Orchestrator) (Núñez, 2026) is a Python package, available on PyPI as `py-devo`, that converts a plain CSV file into iCSV. It infers column types, ranges, and missing-value counts directly from the data, writes a corresponding Frictionless Table Schema (Fagundes and Ribeiro Junior, 2020; Petti and Karev, 2022; Petti, 2023), and

produces a human-readable validation report. It is available both as a command-line tool (`devorun|enrich|validate`) and as a Python library, and is intended to support data producers preparing files for archival.

4.4 AUTOMATED QA/QC WITH THE FRICTIONLESS FRAMEWORK

One of the principal operational benefits of embedding rich, structured metadata directly in iCSV files is the ability to derive machine-actionable validation schemas and run deterministic QA/QC checks automatically. The Frictionless framework (Fagundes and Ribeiro Junior, 2020; Petti and Karev, 2022; Petti, 2023) provides lightweight, well-documented primitives for tabular schema representation, type validation, and rule-based checks. When an iCSV file contains machine-oriented keys such as per-field `types`, the declared `nodata` sentinel, and per-field constraints, a validator can (a) construct a Frictionless JSON schema deterministically, (b) validate the extracted `[DATA]` block, and (c) emit both human-readable and machine-readable reports that can be used in CI pipelines, ingestion gates, and archival audits.

Overview of the validator workflow

A practical validator follows three ordered steps:

- 1. Parse metadata and fields.** Read the identification line and the `# [METADATA]` and `# [FIELDS]` blocks. Extract required items such as `field_delimiter`, the declared `nodata` sentinel, per-field `types`, and per-field constraints.
- 2. Resolve or build a schema.** If an external schema is supplied (for example, as a sidecar file alongside the iCSV), resolve and use it. Otherwise, build a Frictionless table schema directly from the in-band metadata. Include the declared `nodata` value in the schema's `missingValues` list and field-level constraints where available.
- 3. Extract and validate data.** Extract the `[DATA]` block (honoring the declared delimiter and any profile-specific markers, e.g., 2D Time Series date tokens), write a clean CSV, and validate it using the Frictionless `Resource(...).validate()` API. Produce (a) a human-readable text summary, (b) a machine-readable JSON report, and (c) a deterministic exit code for CI integration.

Given the iCSV 2D Time Series above, an equivalent Frictionless schema (JSON) produced by the validator would be (Listing 11):

```

1 {
2   "fields": [
3     {"name": "layer_index", "type": "integer"},
4     {"name": "T", "type": "number", "constraints": {"minimum": -150,
5       "maximum": 60}},
6     {"name": "SSA", "type": "number"},
7     {"name": "RH0", "type": "number", "constraints": {"minimum": 0,
8       "maximum": 1000}}
9   ],
10  "missingValues": ["-999", "-999.000000"]
11 }
```

Listing 11 Example Frictionless schema generated from iCSV 2D Time Series metadata.

This schema can be fed directly into Frictionless for validation. By deriving field types and constraints deterministically from iCSV metadata, the validator avoids heuristics that rely on sampling data, ensuring consistent QA/QC across files and workflows.

DEVO (Núñez, 2026), introduced above, offers a concrete, publicly available implementation of this pattern: given a CSV or iCSV file, it derives a Frictionless schema directly from the file's fields and data, rather than resolving an externally referenced schema, validates the data against that schema, and emits both the JSON schema and a human-readable report, closely mirroring steps (b) and (c) above.

4.5 LIMITATIONS

Although the iCSV file format prioritizes user-friendliness, accessibility, and long-term interpretability through its human-readable structure, it necessarily entails certain trade-offs

that should be acknowledged. The most significant limitation is its large size, which has to be mitigated by compressing it. In order to preserve user-friendliness and long-term interpretability as much as possible, the compression format must be ubiquitous across platforms and software so that there are good reasons to assume it will remain supported in decades to come. A good candidate is the *deflate* algorithm (Deutsch, 1996; Ziv and Lempel, 1978) implemented in *gzip*. As seen in Table 2, for the long-term time series of meteorological data at one single station, it delivers competitive performance even compared to *NetCDF*. Higher performance could be achieved with more modern compression algorithms, as shown in Table 3. The *zstd* algorithm (Collet and Kucherawy, 2021) would be a very attractive choice, achieving good compression ratio with low computing overhead; it is available in a very large number of programming languages, but it is unfortunately not widely available in end-user tools. The *bz2* algorithm (Seward, 1996) also has some merits (best compression ratio of the benchmark with lower runtime than the current *gzip*), but although it has been widely used for several decades, it is not as widespread as *gzip*.

OUTPUT FORMAT	RUN TIME	FILE SIZE
iCSV, uncompressed	2.6679 (x1)	36663150 (÷1)
iCSV compressed	5.7218 (x2.1)	8807964 (÷4.2)
SMET	2.2000 (x0.8)	51560764 (÷0.7)
NetCDF	3.9806 (x1.5)	23407360 (÷1.6)

ALGORITHM	RUN TIME	FILE SIZE
no compression	0.3546 (x1)	36663150 (÷1)
gz (MeteoIO internal)	5.7218 (x16.1)	8807964 (÷4.2)
gz (-9, CLI)	4.3148 (x12.2)	9039555 (÷4.1)
bz2 (-9)	1.5897 (x4.5)	7024546 (÷5.2)
lzma (-9)	16.1591 (x45.6)	7131319 (÷5.1)
xz (-9)	16.1462 (x45.5)	7132476 (÷5.1)
brotli (-best)	51.3221 (x144.7)	7247582 (÷5.1)
zstd (-9)	0.8500 (x2.4)	8896637 (÷4.1)
lz4 (-9)	1.7564 (x5.0)	11532276 (÷3.2)
lzop (-9)	8.0297 (x22.6)	11482999 (÷3.2)

One inherent limitation of the iCSV format that can not be mitigated is that it does not offer any specific streaming capabilities nor advanced data extraction capabilities such as those available in *NetCDF*. However, this trade-off is often acceptable for environmental datasets which (except for large-scale gridded data) typically remain under a few gigabytes, even with extensive metadata, therefore well within practical limits for text-based storage. The dataset from Stillberg (Frei et al., 2023), for example, contains observations from fifty years with an hourly and later ten-minute resolution, and is well below a gigabyte. Modern computing infrastructure, including cloud storage solutions and distributed processing frameworks, further mitigates these size limitations in practical applications.

Notably, iCSV does not preclude storing data in a column-major format, which can substantially improve lookup performance when combined with streaming compression. Furthermore, since most compression algorithms are standardized and widely implemented, the risk of losing data access due to evolving computing environments is minimal. Nonetheless, we acknowledge that text-based formats like iCSV may become a bottleneck for data-intensive workflows. However, iCSV is not designed for such use cases and instead targets applications where interpretability and accessibility are prioritized over raw performance.

Another potential issue is the flexible nature of the metadata, which can lead to insufficient documentation by data producers. To address this, we propose implementing and supporting

Table 2 Benchmark of two iCSV format variants (uncompressed and *gzip*-compressed) with half-hourly meteorological and snowpack measurements from Weissfluhjoch, Davos, Switzerland. This dataset (Wever, 2017) has been read with *MeteoIO* and written out in diverse formats on a 10th generation Intel i9-10900 computer with 32 GB RAM. For each benchmark, three warm-up runs have been conducted, followed by ten timed runs. The median of such runs is shown in the table. The SMET output is optimized for readability with added white-space padding, leading to a much larger file size.

Table 3 Benchmark of compression algorithms for iCSV files. The Weissfluhjoch dataset (Wever, 2017) is used and compressed by command line tools on a 10th-generation Intel i9-10900 computer with 32 GB RAM. For each benchmark, three warm-up runs have been conducted, followed by ten timed runs. The median of such runs is shown in the table. The baseline *no compression* case reads the file content to the standard output and redirects it to a file in order to represent the data transfer speed. The *MeteoIO* internal compression with *gzip* forces the *Z_FILTERED* mode that is specially tailored to repetitive patterns in data.

“application profiles”. These profiles predefine the necessary metadata for specific applications, such as meteorological stations or satellites, based on established standards like CF conventions and WMO standards. This approach ensures consistent and comprehensive data documentation across different domains within environmental science.

For example, an application profile for meteorological stations might require additional metadata fields such as station elevation, instrument types, and maintenance history. Similarly, a hydrological profile might specify requirements for watershed information, flow measurement techniques, and water quality parameters. These application profiles would be indicated in the first line of the file after the format and encoding information. Since iCSV is fully self-describing, even when application profiles might be lost in the future, the data can still be read and processed easily.

5 CONCLUSION AND OUTLOOK

The scientific community faces a pressing challenge: if environmental science is to meet the urgency of climate change, biodiversity loss, and extreme weather, it must overcome persistent barriers to effective data sharing. Data that are difficult to interpret, poorly documented, or locked in fragile formats slow down discovery and undermine collaboration. Addressing these obstacles requires not only robust policies and infrastructures but also practical tools that make everyday data work easier, more reliable, and more reproducible.

The interoperable CSV (iCSV) file format is designed as one such tool: a lightweight but powerful step toward more efficient collaboration and stronger scientific impact. This work introduces iCSV, a standardized approach to three critical challenges in environmental data management: cross-institutional exchange, long-term interpretability, and user accessibility. The development of iCSV responds directly to the growing recognition that data management practices must evolve to support increasingly collaborative and long-term research initiatives that span traditional disciplinary and institutional boundaries. It directly supports the FAIR principles by embedding metadata for findability, using open standards for interoperability, and ensuring long-term accessibility through human-readable text.

By embedding structured metadata directly within the familiar CSV framework, iCSV offers a solution that balances machine readability with human interpretability. This design ensures that data remains fully contextualized and interpretable for future users, even as software environments and institutional knowledge change over time. Its text-based nature promotes broad compatibility with existing data processing tools and workflows, while also guaranteeing accessibility without specialized software—a critical requirement for preservation across decades or even centuries. The flexible structure allows for extension and adaptation to new use cases and domains, while remaining simple to use even without detailed application profiles.

The iCSV format combines the most effective aspects of existing approaches: the comprehensive metadata framework of NetCDF/CF conventions, the practical structure of SMET files, and the universal readability of CSV. This unique integration satisfies all requirements we set out to address, namely:

- **Self-describing:** Through its structured metadata and fields sections, each iCSV file carries essential context that makes the data understandable far into the future.
- **Interoperable:** By adhering to established standards like CF conventions and EPSG coordinate systems, iCSV files can be seamlessly integrated into diverse scientific workflows.
- **Accessible:** The human-readable format ensures that data remains accessible even as software and systems evolve over time.
- **Flexible:** The format accommodates a wide variety of environmental data types, from simple time series to complex spatio-temporal datasets, while maintaining its core simplicity.

Crucially, iCSV adds value not only for future users but also for today’s researchers. By making metadata both explicit and machine-actionable, iCSV enables automated quality checks that validate data structure, types, ranges, and constraints. This capability helps researchers identify

issues early in the workflow, improving data quality at the source, while also supporting reproducibility and transparency downstream. In this way, iCSV bridges the needs of current data producers with those of long-term data consumers.

We encourage the scientific community to replace its use of CSV files with iCSV files, as it has the potential to significantly improve data accessibility, foster robust sharing practices, and raise data quality standards across the environmental science community. To support uptake, we provide companion software tools that simplify handling metadata, performing conversions to and from iCSV, and running validation checks, thereby lowering the barrier to adoption.

We acknowledge that some limitations remain and that the iCSV ecosystem is still evolving. Future work will therefore focus on broadening the surrounding tooling and profiles, including Python plotting support; improved conversion utilities for integration with existing workflows; support for compression and other approaches for handling larger datasets efficiently; online and local validation services; and the continued development of application profiles for different disciplines.

We encourage the scientific community to contribute to the evolution of iCSV by developing new application profiles, tools, and validation services. By working together, we can ensure that iCSV meets the needs of diverse environmental science disciplines and becomes a standard for data exchange and archival. As environmental data continues to grow in both volume and importance, formats like iCSV that prioritize not only long-term accessibility and interoperability but also present-day data quality will play a crucial role in advancing scientific understanding and environmental stewardship. By offering a balance between human readability, machine-actionable metadata, and built-in validation capabilities, iCSV provides a practical foundation for stronger collaboration and higher-quality science. While no format alone can solve the environmental crises we face, the adoption of simple, interoperable, and robust data standards is a necessary step in ensuring that science can respond effectively. iCSV is designed to be such a step—an immediately usable improvement that can make the daily work of data sharing easier today, while safeguarding the interpretability and usefulness of our collective data for generations to come.

ACKNOWLEDGEMENTS

We acknowledge the contributions of all researchers and institutions who provided feedback on the iCSV format specification. Special thanks to the developers of MeteoIO and Snowpat libraries for implementing support for this format.

AI tools were used for rephrasing, grammar checking, and style polishing of this manuscript. No AI tools were used for content outline, logical structure, or scientific reasoning of the paper.

COMPETING INTERESTS

The authors have no competing interests to declare.

AUTHOR AFFILIATIONS

Patrick Leibersperger  orcid.org/0009-0006-3390-2977
Institute for Snow and Avalanche Research (SLF), Davos, CH

Mathias Bavay  orcid.org/0000-0002-5039-1578
Institute for Snow and Avalanche Research (SLF), Davos, CH

Ionut Iosifescu Enescu  orcid.org/0000-0002-1770-7833
Swiss Federal Institute for Forest, Snow and Landscape Research (WSL), Birmensdorf, CH

Chase Núñez  orcid.org/0000-0002-3273-6966
Lib4RI, Dübendorf, CH

REFERENCES

- Attribute Convention for Data Discovery 1-3** (2023) Earth Science Information Partners (ESIP). Available at: https://wiki.esipfed.org/Attribute_Convention_for_Data_Discovery_1-3 (visited on 05/22/2025).
- Batista, D. et al.** (2022) 'Machine actionable metadata models', *Scientific Data*, 9(1). Available at: <https://doi.org/10.1038/s41597-022-01707-6>

- Bavay, M.** (2021) 'Format specification for the SMET Weather Station Meteorological Data Format version 1.2'. <https://meteoio.slf.ch/doc-dev/html/smetio.html>
- Bavay, M. and Egger, T.** (2014) 'MeteoIO 2.4.2: a preprocessing library for meteorological data', *Geoscientific Model Development*, 7(6): pp. 3135–3151. Available at: <https://doi.org/10.5194/gmd-7-3135-2014>.
- Bavay, M., Fiddes, J. and Godøy, Ø.** (2020) 'Automatic data standardization for the global cryosphere watch data portal', *Data Science Journal*, 19(1): pp. 1–8. Available at: <https://doi.org/10.5334/dsj-2020-006>
- Bergès, J.C.** (2002) 'Support of WMO binary format (BUFR and GRIB)'. *Proceedings of the Open source GIS-GRASS user conference, Trento, Italy*. Vol. 11, p. 13.
- Boehm, M.** (2025) 'Technical perspective: TASHEEH: repairing row-structure in raw CSV files', *ACM SIGMOD Record*, 54(1): p. 89. Available at: <https://doi.org/10.1145/3733620.3733638>
- Bratt, S. et al.** (2017) 'Big data, big metadata and quantitative study of science: A workflow model for big scientometrics', *Proceedings of the Association for Information Science and Technology*, 54(1): pp. 36–45. Available at: <https://doi.org/10.1002/pr2.2017.14505401005>
- Butchart, S.H.M. et al.** (2010) 'Global biodiversity: indicators of recent declines', *Science*, 328(5982): pp. 1164–1168. Available at: <https://doi.org/10.1126/science.1187512>.
- Caron, J. and Oxelson, J.** (2013) 'BUFR and GRIB file formats considered harmful for data archiving'. In: *EGU general assembly conference abstracts, EGU2013–1961*. Göttingen, Germany: Copernicus Publications. <https://ui.adsabs.harvard.edu/abs/2013EGUGA.15.1961C/abstract>
- Chacon, S. and Straub, B.** (2014) *Pro git*. Berkeley, CA: Apress. Available at: <https://doi.org/10.1007/978-1-4842-0076-6>
- Collet, Y. and Kucherauw, M.** (2021) *Zstandard Compression and the 'application/zstd' Media Type*. RFC 8878. Available at: <https://doi.org/10.17487/RFC8878>
- Deutsch, P.** (1996) *DEFLATE Compressed Data Format Specification version 1.3*. RFC 1951. Available at: <https://doi.org/10.17487/RFC1951>
- Fagundes, M.F. and Ribeiro Junior, D.I.** (2020) 'Frictionless Data-based model applied to open government data', *RDBCI: Revista Digital de Biblioteconomia e Ciência da Informação*, 18: p. e020034. Available at: <https://doi.org/10.20396/rdbci.v18i00.8661528>
- Fierz, C. et al.** (2016) 'SnopViz, an interactive visualization tool for both snow-cover model output and observed snow profiles'. *Proceedings of the International Snow Science Workshop, CO*, pp. 3–7. Available at: https://arc.lib.montana.edu/snow-science/objects/ISSW16_P1.29.pdf.
- Folk, M. et al.** (2011) 'An overview of the HDF5 technology suite and its applications'. *Proceedings of the EDBT/ICDT 2011 Workshop on Array Databases*. AD '11. New York, NY, USA: Association for Computing Machinery, pp. 36–47. Available at: <https://doi.org/10.1145/1966895.1966900>
- Frei, E.R. et al.** (2023) *Long-term meteorological station Stillberg, Davos, Switzerland at 2090 m a.s.l.* Available at: <https://doi.org/10.16904/envidat.396>. url: <https://www.envidat.ch/dataset/meteo-stillberg>.
- Gupta, J. et al.** (2023) 'Earth system justice needed to identify and live within Earth system boundaries', *Nature Sustainability*, 6(6): pp. 630–638. Available at: <https://doi.org/10.1038/s41893-023-01064-1>
- Hameed, M., Vitagliano, G., Panse, F. and Naumann, F.** (2024) 'TASHEEH: Repairing Row-Structure in Raw CSV Files.', *International Conference on Extending Database Technology*. <https://api.semanticscholar.org/CorpusID:268554663>
- Hankin, S. et al.** (2009) 'NetCDF-CF-OPeNDAP: Standards for ocean data interoperability and object lessons for community data standards processes'. *Proceedings of OceanObs'09: Sustained Ocean Observations and Information for Society*. Available at: <http://www.oceanobs09.net/proceedings/cwp/Hankin-OceanObs09.cwp.41.pdf>
- Hart, E. et al.** (2016) 'Ten simple rules for digital data storage', *PeerJ Preprints*, 4: p. e1448v2. Available at: <https://doi.org/10.7287/peerj.preprints.1448v2>
- Hassell, D. et al.** (2017) 'A data model of the Climate and Forecast metadata conventions (CF-1.6) with a software implementation (cf-python v2.1)', *Geoscientific Model Development*, 10(12): pp. 4619–4646. Available at: <https://doi.org/10.5194/gmd-10-4619-2017>
- Herring, J.R.** (2011) *OpenGIS Implementation Standard for Geographic information - Simple feature access - Part 1: Common architecture*. Version 1.2.1. Available at: <https://www.opengis.net/doc/is/sfa/1.2.1> (visited on 06/29/2026).
- IAGA 2002 Data Exchange Format** (2026) Version June-2015. Available at: <https://www.ncei.noaa.gov/services/world-data-system/v-dat-working-group/iaga-2002-data-exchange-format> (visited on 06/29/2026).
- Karhila, V. and Hernandez-Carrascal, A.** (2010) 'BUFR: a meteorological code for the 21st century'. *2010 EUMETSAT Meteorological Satellite Conference, Cordoba, Spain*, Vol. 5.
- Key, J. et al.** (2016) *Global Cryosphere Watch (GCW) Implementation Plan*. Tech. rep. World Meteorological Organization. Available at: <https://library.wmo.int/idurl/4/55518>.
- Kirchhoff, J., Gottschalk, S. and Engels, G.** (2022) 'Detecting Data Incompatibilities in Process-Driven Decision Support Systems', in: *Business modeling and software design*. New York: Springer International Publishing, pp. 89–103. Available at: https://doi.org/10.1007/978-3-031-11510-3_6

- Kitchin, R. (2014) 'Big Data, new epistemologies and paradigm shifts', *Big Data & Society*, 1(1). Available at: <https://doi.org/10.1177/2053951714528481>
- Leibersperger, P. (2024) SNOWPAT. Available at: <https://snowpat.slf.ch/> (visited on 05/22/2025).
- Manual on codes. *International codes. volumes I.2 & 1.2* (2009) 2009 ed. OCLC: 607526193. Geneva, Switzerland: Secretariat of the World Meteorological Organization.
- Michna, P. and Woods, M. (2013) 'RNetCDF - A Package for Reading and Writing NetCDF Datasets', *The R Journal*, 5(2): p. 29. Available at: <https://doi.org/10.32614/rj-2013-023>
- Mitlöhner, J. et al. (2016) 'Characteristics of Open Data CSV Files', *2nd International Conference on Open and Big Data (OBD)*. IEEE. Available at: <https://doi.org/10.1109/obd.2016.18>.
- Non-Binary Environmental Archive Data (NEAD) format - EnviDat (2024) Available at: <https://www.envidat.ch/dataset/nead> (visited on 05/10/2024).
- Núñez, C. (2026) DEVO: Data Enrichment and Validation Orchestrator. Version 0.2.2. Available at: <https://github.com/chasenunez/devo> (visited on 07/03/2026).
- Pandas, Development Team (Feb. 2020) *pandas-dev/pandas: Pandas*. Version latest. Available at: <https://doi.org/10.5281/zenodo.3509134>
- Parton, G.A. and Pepler, S. (2009) *The BADC-CSV Format: Meeting user and metadata requirements*. Available at: <https://doi.org/10.5281/zenodo.7355840>
- Petti, S. (2023) 'A better governance for the Frictionless Standards'. In: *Submitted to Open Infrastructure Fund / Fondo de Infraestructura Abierta*. Available at: <https://openreview.net/forum?id=02iXOVKoPo>.
- Petti, S. and Karev, E. (2022) 'Frictionless Data Standards'. In: *Commonplace*. Available at: <https://doi.org/10.21428/6ffd8432.1619ad93>
- Plantin, J.-C., Lagoze, C. and Edwards, P.N. (2018) 'Re-integrating scholarly infrastructure: the ambiguous role of data sharing platforms', *Big Data & Society*, 5(1). Available at: <https://doi.org/10.1177/2053951718756683>
- Rew, R. and Davis, G. (1990) 'NetCDF: an interface for scientific data access', *IEEE Computer Graphics and Applications*, 10(4): pp. 76–82. Available at: <https://doi.org/10.1109/38.56302>
- Rixen, C. et al. (2014) 'Faster, higher, more? Past, present and future dynamics of alpine and arctic flora under climate change', *Alpine Botany*, 124(2): pp. 77–79. Available at: <https://doi.org/10.1007/s00035-014-0141-z>
- Romano, D. (1997) 'Data mining leading edge: Insurance & banking', In: *Proceedings of Knowledge Discovery and Data Mining, Unicom, Brunel University*.
- Seward, J. (1996) 'bzip2 and libbzip2'. pp. 8–18. Available at: <http://www.bzip.org>.
- Shafranovich, Y. (2005) *Common Format and MIME Type for Comma-Separated Values (CSV) Files*. Request for Comments RFC 4180. Num Pages: 8. Internet Engineering Task Force. Available at: <https://doi.org/10.17487/RFC4180>. url: <https://datatracker.ietf.org/doc/rfc4180> (visited on 05/10/2024).
- Tenopir, C. et al. (2020) 'Data sharing, management, use, and reuse: Practices and perceptions of scientists worldwide', *PLOS ONE*, 15(3): p. e0229003. Available at: <https://doi.org/10.1371/journal.pone.0229003>
- van den Burg, G.J.J., Názábal, A. and Sutton, C. (2019) 'Wrangling messy CSV files by detecting row and type patterns', *Data Mining and Knowledge Discovery*, 33(6): pp. 1799–1820. Available at: <https://doi.org/10.1007/s10618-019-00646-y>
- Wever, N. (2017) *WFJ_MOD: Meteorological and snowpack measurements from Weissfluhjoch, Davos, Switzerland*. Available at: <https://doi.org/10.16904/1>. url: <https://www.envidat.ch/dataset/10-16904-1>.
- Wilkinson, M.D. et al. (2016) 'The FAIR Guiding Principles for scientific data management and stewardship', *Scientific Data*, 3(1). Available at: <https://doi.org/10.1038/sdata.2016.18>
- Yang, M. et al. (2019) 'An efficient storage and service method for multi-source merging meteorological big data in cloud environment', *EURASIP Journal on Wireless Communications and Networking*, 2019(1). Available at: <https://doi.org/10.1186/s13638-019-1576-0>
- Zhang, Y. et al. (2023) 'Multi-sensor integration management in the earth observation sensor web: State-of-the-art and research challenges', *International Journal of Applied Earth Observation and Geoinformation*, 125: p. 103601. Available at: <https://doi.org/10.1016/j.jag.2023.103601>
- Ziv, J. and Lempel, A. (1978) 'Compression of individual sequences via variable-rate coding', *IEEE transactions on Information Theory*, 24(5): pp. 530–536. Available at: <https://doi.org/10.1109/TIT.1978.1055934>
- Zuiderwijk, A., Jeffery, K. and Janssen, M. (2012) 'The Potential of Metadata for Linked Open Data and its Value for Users and Publishers', *JeDEM – eJournal of eDemocracy and Open Government*, 4(2): pp. 222–244. Available at: <https://doi.org/10.29379/jedem.v4i2.138>

TO CITE THIS ARTICLE:

Leibersperger, P., Bavay, M., Enescu, I.I. and Núñez, C. 2026 Interoperable CSV for Environmental Data Archival and Exchange: iCSV. *Data Science Journal* 25: 39, pp. 1–19. DOI: <https://doi.org/10.5334/dsj-2026-039>

Submitted: 12 February 2026

Accepted: 05 October 2026

Published: 22 October 2026

COPYRIGHT:

© 2026 The Author(s). This is an open-access article distributed under the terms of the Creative Commons Attribution 4.0 International License (CC-BY 4.0), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited. See <https://creativecommons.org/licenses/by/4.0/>.

Data Science Journal is a peer-reviewed open access journal published by Ubiquity Press.