

A Standard for Accessing Heliophysics Time Series Data

By Robert S Weigel

1

A Standard for Accessing Heliophysics Time Series Data

2

3

4

5

6

7

Robert Weigel^{1,2}, Jon Vandegriff³, Jeremy Faden^{4,5}, Todd King⁶, D Aaron Roberts⁷, Bernard Harris⁷, Robert Candey⁷, Nand Lal⁷, Scott Boardsen⁷, Chris Lindholm^{8,9}, Doug Lindholm^{8,9}, Thomas Baltzer^{8,9}, Larry Brown, Eric Grimes⁶, Baptiste Cecconi¹⁰, Vincent Génot¹¹, Benjamin Renard¹², Arnaud Masson¹³, Beatriz Martinez¹⁴

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

¹George Mason University

²Space Weather Lab

³Johns Hopkins Applied Physics Lab

⁴Cottage Systems

⁵University of Iowa

⁶University of California Los Angeles,

⁷NASA Goddard Space Flight Center

⁸University of Colorado

⁹Laboratory for Atmospheric and Space Physics

¹⁰LESIA, Observatoire de Paris-PSL, CNRS, Meudon, France

¹¹Institut de Recherche en Astrophysique et Planétologie, CNRS, Univ. Toulouse, CNES, Toulouse, France

¹²AKKA Technologies

¹³Tpz UK for ESA, ESAC

¹⁴RHEA Group for ESA, ESAC

23

Key Points:

24

25

26

- There are many non-standard approaches to serving Heliophysics time series data.
- HAPI provides a standard specification that will simplify data access.
- Many software tools have been developed for servers and clients.

Corresponding author: R.S. Weigel, rweigel@gmu.edu

3

1

Abstract

Heliophysics data analysis often involves combining diverse science measurements, many of them captured as time series. Although there are now only a few commonly used data file formats, the diversity in mechanisms for automated access to and aggregation of such data holdings can make analysis that requires inter-comparison of data from multiple data providers difficult. The Heliophysics Application Programmer's Interface (HAPI) is a recently developed standard for accessing distributed time-series data to increase interoperability. The HAPI specification is based on the common elements of existing data services and it standardizes the two main parts of a data service: the request interface and the response data structures. The interface is based on the REpresentational State Transfer (REST) or RESTful architecture style, and the HAPI specification defines five required REST endpoints. Data are returned via a streaming format that hides file boundaries; the metadata is detailed enough for the content to be scientifically useful, e.g., plotted with appropriate axes layout, units, and labels. Multiple mature HAPI-related open-source projects offer server-side implementation tools and client-side libraries for reading HAPI data in multiple languages (IDL, Java, MATLAB, and Python). Multiple data providers in the US and Europe have added HAPI access alongside their existing interfaces, and based on this experience, data can be served via HAPI with little or no information loss compared to similar existing web interfaces. Finally, HAPI has been adopted as a recommended COSPAR standard for time series data delivery.

Plain Language Summary

The Heliophysics Application Programmer's Interface (HAPI) specification allows data providers to use a standard set of conventions for returning data in response to a URL-based request. This specification was developed because many data providers provide such data with incompatible conventions so that users wanting automated access to and aggregation of data must develop custom code for each data provider's interface. The benefit to the user is that clients for many common languages have been developed to read HAPI data; if a data provider serves data using the HAPI specification, the user does not need to write additional code to access the data. The starting point for using and learning more about HAPI is <http://hapi-server.org/>.

1 Introduction

Heliophysics (HP) researchers study the nature and dynamical interactions of the Sun, the heliosphere, and the plasma environments of the planets based on data from a fleet of spacecraft and ground-based platforms termed the "Heliophysics System Observatory" (HSO) (*Heliophysics Data Environment*, 2021). As we seek a more detailed understanding of the global and local dynamics of this system, we increasingly need easy access to data from many of the observatories. The best route to integrating these data is the adoption of standards for data formats, metadata terminology and syntax, and access APIs (Application Programming Interface). HP has nearly universally adopted the data file formats of CDF (*NASA Common Data Format*, 2021), FITS (Ponz et al., 1994), HDF (Folk et al., 1999), and netCDF (Rew & Davis, 1990); and SPASE (King et al. (2010); Roberts et al. (2018)) for metadata. As summarized in the following section, there are many different APIs for obtaining time series-like data stored in these data files. (In this work, we use time series-like to mean data structures that for each time value have an associated scalar, vector, tensor, spectrogram, or higher-dimensional construct.)

The primary design objective for HAPI is to provide the simplest API that allows access to time series data in a streaming form and allows a user not to need knowledge of file system boundaries, directory layouts, and file formats. With this approach, a scientist programming in, for example, IDL, MATLAB, or Python can use a line of code to bring data into a processing or analysis routine from a diverse set of data providers

77 and for a wide array of data – from uniform cadence ground magnetometer measurements
 78 to multi-dimensional distribution functions of ions from a spacecraft instrument. This
 79 paper provides an overview of the HAPI design and specification, a summary of its API,
 80 and a discussion of plans for future development. The API is mature, and the current
 81 effort involves broadening the adoption of the standard by data providers and increas-
 82 ing the use of the API and the HAPI software clients by scientists.

83 1.1 Related Works

84 As HP projects have primarily converged on CDF as the preferred data file format,
 85 the next barrier in interoperability is the diversity of APIs needed to access various HP
 86 data centers. Table 1 lists common access mechanisms and an accounting of the meth-
 87 ods used for several well-known data providers listed in Table 2.

Table 1. Common Heliophysics data access methods

Method	Description
A.	A FTP or HTTP directory of files (usually one day of data per file).
B.	A web page with link that indicates when processing is complete, often with the ability to display the data and download it.
C.	A web page with link to an archive of files (zip or tgz) that is available when the processing of a request is complete.
D.	A REST API that returns a data stream or file directly to a user’s application.

Table 2. Heliophysics data providers, links to their top-level web site, and the data access methods listed in Table 1 that they natively support. An asterisk “*” indicates that data are already available through a HAPI API (*HAPI Server List*, 2021). A plus “+” indicates that HAPI support is under development. Not all of the above data providers that use methods A, B, and C use the same file formats (when files provided); none of the data providers using method D use the same API. However, they all provide data that could be served without loss information through a HAPI server.

Provider	URL	Access Methods
AMDA* (Génot et al., 2021)	http://amda.irap.omp.eu/service/hapi	B, D
CAIO+	https://csa.esac.esa.int/csa/aio/	B, C, D
CARISMA	http://www.carisma.ca/	C
CCMC/iSWA*	https://iswa.gsfc.nasa.gov/	D ⁴
CDAWeb*	https://cdaweb.gsfc.nasa.gov/	A, B, C, D
Das2*	http://das2.org/	D
FTEC*	http://hapi.ftecs.com/hapi	A
IMAGE	http://space.fmi.fi/image	C
INTERMAGNET+	http://intermagnet.org/	A, C
ISAS/JAXA	https://darts.isas.jaxa.jp	A
LiSIRD*	http://lasp.colorado.edu/lisird/	A, D
OMNIWeb*	https://omniweb.gsfc.nasa.gov/	A, B, D
NGDC	https://ngdc.noaa.gov/stp/satellite/	A, C, D
PDS+	https://pds-ppi.igpp.ucla.edu/hapi	A, B
SSCWeb*	https://sscweb.gsfc.nasa.gov/	B, D
SuperMAG	http://supermag.jhuapl.edu/	D
SWPC	https://www.swpc.noaa.gov/	B
MADRIGAL	https://openmadrigal.org/	A, D

1.2 Design Constraints and Associated Motivations

The HAPI specification (*HAPI Project Overview Page*, 2021) was developed by Heliophysics researchers and data scientists as a grass-roots effort to improve interoperability in Heliophysics data. Experience with related efforts such as TSDS (Weigel et al., 2010), a service that mapped data available by methods A to D in Table 1 through a single API, led to the choice of REST (Representational state transfer; Richardson and Ruby (2007)) style services. The REST conventions allow for simple, stateless servers and the streaming of data that naturally hides any file boundaries between data files.

The primary design considerations for the HAPI specification were:

1. The minimum requirements for HAPI compliance should be easy to implement for existing, well-structured data collections.

Because ease of implementation increases the likelihood of adoption, the HAPI requirements closely resemble the implementations of many existing time series data providers. They do not include any data operations – only delivering existing content; faced with a new specification, many data providers will likely only ever implement the minimum elements needed for compliance.

2. The required metadata is the minimum amount needed to make the data scientifically usable.

A common way to evaluate the science usability of structured data is whether enough information is present to automatically create a labeled, scientifically comprehensible plot of the data. To reduce the effort required to serve data through a HAPI API, the required HAPI metadata is limited to that needed to do this and allow common content access and usage scenarios rather than search and discovery use-cases. However, HAPI metadata can point to other richer metadata or explanatory information about a dataset, and metadata for search and discovery can point to HAPI endpoints and metadata. For example, in Heliophysics, the *Space Physics Archive, Search and Extract (SPASE)* metadata model (Roberts et al., 2018) has become a standard method for describing data products. (SPASE does not include a specification for the access of data.) In addition to pointing to unstructured external information, HAPI metadata can reference any existing SPASE record, which has structured information such as detailed documentation and caveats. Also, SPASE records are being updated to reference the HAPI-enabled servers indicated in Table 2 as a possible access mechanism for access to data products.

3. Data are accessible at a minimum in a simple CSV-formatted stream

Although HAPI servers may use high-performance streaming formats that are defined in the specification, all servers must support CSV for output. CSV data is easy to read in all scientific programming languages with only a few lines of code, making it easy for scientists to start working with the data immediately.

The HAPI streaming format requires servers to create a response stream that aggregates data if it is stored in multiple files. While this introduces complexity, in potential conflict with constraint (1), a simplifying aspect is that servers do not need to have a staging process for file aggregation nor do they need to operate asynchronously or perform user notifications when the result of a request is complete.

The requirement that HAPI requests and responses occur synchronously adds a practical limitation on the amount of data sent in a single response. The HAPI specification does not dictate any data transfer limits. However, servers may define data volume constraints and report via a HAPI error response that a request was too large to process.

The HAPI specification and server capabilities are related to OPeNDAP (Cornillon et al., 2003) and NeML (Nativi et al., 2005), but the HAPI specification was designed primarily for time-series data, whereas OPeNDAP is primarily used for spatial data, although it can be used for time series. Several of the developers had extensive experience in attempting to adopt OPeNDAP for providing HAPI-like services, but this effort was abandoned in favor of the HAPI specification due to the complications of mapping from OPeNDAP's data model, in which space is a primary index, to HAPI, in which time is the primary index.

1.3 Data and Metadata Formats

All HAPI metadata uses JSON (Javascript Object Notation), which has near-universal support in modern scientific programming languages.

A HAPI server may stream data in CSV and optionally JSON or binary. CSV is highly accessible for scientists and is easily read by all modern programming languages. JSON is also widely supported by many languages and web-based frameworks.

The binary format is a straightforward translation of the CSV output: newlines are removed and integer parameters are written as 64-bit signed integers. Floating-point parameters are written as 64-bit IEEE 754 little-endian floats (*IEEE 754 Standard*, 2021), and string parameters are written as a sequence of 8-bit ASCII characters. HAPI metadata contains the information required to interpret the binary file, including the number of parameters, the dimensionality of each parameter, and the formats of numerical parameters (or length for string parameters).

This custom, simple, and efficient binary format was selected because typical scientific data formats, such as CDF (*NASA Common Data Format*, 2021), HDF (Folk et al., 1999), and netCDF (Rew & Davis, 1990) do not support streaming of data. Experimental code for streaming HDF and netCDF can be found online, but not as integrated parts of the code base for those formats. Protocol buffers (*Protocol Buffers*, 2021) were considered, but library support for this format in the commonly used programming languages (IDL, MATLAB, Python) is poor.

As described in the following section, few scientists should ever need to write parsers and readers for HAPI metadata or data.

1.4 Facilitating Adoption

The HAPI specification is designed to provide a standardized way to provide streaming services that many time-series data providers already offer. The use of the common aspects of existing features of HP data providers means that data providers can, with minimal effort, modify existing streaming servers to make them HAPI compliant.

In addition, several tools have been developed to facilitate adoption.

1. Although the HAPI streaming formats are simple enough that only 10-20 lines of code are needed to read a basic HAPI response, there are error conditions and other optimizations to consider (such as caching and parallelization of requests); a robust and comprehensive client implementation is beyond what should be expected of science users or even software developers who want to create tools that use HAPI data. Mature HAPI server client libraries are available from open-source HAPI projects (*HAPI GitHub Project Page*, 2021) for Java, IDL, MATLAB, and Python, so most users can begin with these libraries. These clients read a HAPI response into a data structure appropriate for the given language (e.g, in Python the response is read into a NumPy N-D array with timestamps converted to Python `datetime` objects). Work is underway to incorporate HAPI readers into existing popular anal-

183 ysis frameworks, such as SPEDAS in IDL and Python (*SPEDAS Space Physics*
 184 *Environment Data Analysis Software*, 2021), Autoplot in Java (Faden et al., 2010),
 185 SunPy in Python (Mumford et al., 2015), and SpacePy in Python (Morley et al.,
 186 2010).

- 187 2. To assist server developers with adoption, a cross-platform reference server has
 188 been developed. With this server, a data provider only needs to provide HAPI JSON
 189 metadata and a command-line program that emits HAPI CSV given inputs of a
 190 dataset identifier, and start/stop times (*HAPI Project Overview Page*, 2021). The
 191 server handles HAPI error responses, request validation, and logging.
- 192 3. A validator/verifier has been developed that runs many tests on a server (*HAPI*
 193 *Server Verifier*, 2021). The server executes a comprehensive suite of tests for com-
 194 pliance with the specification. By either downloading and running the test soft-
 195 ware locally, or by entering the URL of a HAPI server into a website, data providers
 196 can test most aspects of their HAPI implementation, with detailed results indi-
 197 cating warnings or errors. This has proven to be a very effective way to assist de-
 198 velopers, and it helps ensure that new HAPI servers are fully functional. The tests
 199 include checks of JSON responses against the HAPI metadata schema, verifying
 200 that the server handles different allowed start/stop time representations (e.g. year,
 201 day-of-year and year, month day with varying amounts of additional time preci-
 202 sion), error responses and message, timeouts, and testing that output is independ-
 203 ent of the output format. Also, the validator/verifier makes recommendations
 204 – for example, if a server does not include cross-origin request sharing headers or
 205 respond with compressed data when gzip is specified in the Accept-Encoding re-
 206 quest header, it emits a message noting their advantages and a link to informa-
 207 tion on how to enable or implement these features.

208 2 API

209 The five required endpoints that constitute the HAPI API are summarized in this
 210 section. In addition to defining the responses to requests to these URLs, the HAPI spec-
 211 ification includes requirements for the format and structure of the response. These ad-
 212 ditional requirements are summarized in the description of the relevant endpoint.

213 2.1 /about endpoint

214 The response to this endpoint is a JSON object containing the HAPI API version
 215 implemented, the status of the response, the `id` of the server, the server `title` (a brief
 216 description of the provided data), and contact information for the server administrator.
 217 An example response is

```
218 {
219   "HAPI": "3.0",
220   "status": {"code": 1200, "message": "OK"},
221   "id": "SSCWeb",
222   "title": "SSCWeb Data and Metadata",
223   "contact": "example@example.org"
224 }
```

225 This endpoint has no request parameters.

226 2.2 /capabilities endpoint

227 The response to this endpoint is a JSON object containing the HAPI API version
 228 implemented, the status of the response, and output formats supported. A client can use

229 this endpoint to determine if the HAPI server provides additional data output formats
230 beyond the required CSV. An example response is

```
231 {  
232   "HAPI": "3.0",  
233   "status": {"code": 1200, "message": "OK"},  
234   "outputFormats": ["csv", "binary", "json"]  
235 }
```

236 Note that the minimum requirement is that `outputFormats=["csv"]`. This end-
237 point has no request parameters.

238 **2.3 /catalog endpoint**

239 The response to a request to this endpoint is a list of the available datasets from
240 the HAPI server. An example is

```
241 {  
242   "HAPI": "3.0",  
243   "status": {"code": 1200, "message": "OK"},  
244   "catalog":  
245     [  
246       {"id": "dataset1", title: "dataset 1 title"},  
247       {"id": "dataset2"}  
248     ]  
249 }
```

250 The minimal requirement is that the objects in the `catalog` array contain an `id`;
251 the `title` is optional. This endpoint has no request parameters.

252 **2.4 /info endpoint**

253 This endpoint has one required request parameter, `dataset`, which corresponds to
254 an identifier of the dataset that appears in the `/catalog` response. The response to this
255 endpoint contains information about the parameters in a dataset. It contains the meta-
256 data needed to satisfy requirement (2) in section 1.2 that HAPI metadata should be the
257 minimum required for a program reading the data to produce a plot with labels needed
258 for scientific interpretation.

259 For example, a response to `/info?dataset=ACE_MAG` could be

```
260 {  
261   "HAPI": "3.0",  
262   "status": {"code": 1200, "message": "OK"},  
263   "startDate": "1998-001Z",  
264   "stopDate" : "2017-100Z",  
265   "parameters":  
266     [{  
267       "name": "Time",  
268       "type": "isotime",  
269       "units": "UTC",  
270       "fill": null,  
271       "length": 24  
272     }],  
273     ...
```

```

274 {
275     "name": "mag_GSE",
276     "type": "double",
277     "units": "nT",
278     "fill": "-1e31",
279     "size" : [3],
280     "description": "hourly ave magnetic field in GSE",
281     "label": "B field in GSE"
282 }
283 }

```

284 In this example, the size element of the `mag_GSE` parameter indicates the dimen-
 285 sionality of the parameter. `size=[3]` means that this parameter will occupy three columns
 286 in the CSV data response. Higher-dimensional parameters are allowed. For example, a
 287 dataset parameter consisting of the energy of protons in one of 6 energy ranges striking
 288 a sensor at 12 different pitch angles would have `size=[12, 6]` and would occupy 72 columns
 289 in the CSV response.

290 2.5 /data endpoint

291 This endpoint has required request parameters of a dataset identifier and time range,
 292 and the response is a CSV stream of all parameters in the dataset. Optionally, the re-
 293 quest may include a comma-separated subset of parameters to return. A sample response
 294 for the request

```

295 /data?dataset=ACE_MAG&parameters=mag_GSE&time.min=2016-01-01Z&time.max=2016-01-02Z

```

296 is

```

297 2016-01-01T00:00:00.000Z,0.05,0.08,-50.98
298 ...
299 2016-01-01T23:59:59.000Z,0.09,0.03,-30.0

```

300 The HAPI binary representation of this file is 86400 blocks, each containing 24 ASCII-
 301 encoded bytes (representing a timestamp) followed by 3 IEEE 754 little-endian 64-bit
 302 floats.

303 3 Summary and Future Development

304 The HAPI specification captures a diverse range of time series representations used
 305 in Heliophysics, including scalars, vectors, spectra, and multi-dimensional time series com-
 306 monly made by energetic particle instruments. The specification also captures many of
 307 the features of existing APIs, simplifying the transition from a non-standards-based API
 308 to a HAPI API.

309 In addition to developing an API, as discussed in 1.4, significant effort was made
 310 to facilitate adoption. For example, for server developers, a generic server is available
 311 – a data provider needs only to provide `/catalog` and `/info` metadata JSON files and
 312 executable that takes inputs of the dataset identifier and start/stop times and returns
 313 HAPI CSV. Usually, such programs are fewer 50 lines. For users, software has been de-
 314 veloped for commonly used programming languages in Heliophysics, including IDL, Java,
 315 MATLAB, and Python.

316 The type of data described by HAPI metadata and served according to its spec-
 317 ification is easily extended to other science domains such as the Earth Sciences and is
 318 an area of active research.

319 References

- 320 Cornillon, P., Gallagher, J., & Sgouros, T. (2003). OPeNDAP: Accessing data in a
 321 distributed, heterogeneous environment. *Data Science Journal*, *2*, 164–174.
- 322 Faden, J. B., Weigel, R. S., Merka, J., & Friedel, R. H. (2010). Autoplot: a browser
 323 for scientific data on the web. *Earth Science Informatics*, *3*(1), 41–49.
- 324 Folk, M., McGrath, R., & Yeager, N. (1999). HDF: an update and future directions.
 325 In *International Geoscience and Remote Sensing Symposium*. IEEE. doi: 10
 326 .1109/igarss.1999.773469
- 327 Génot, V., Budnik, E., Jacquey, C., Bouchemit, M., Renard, B., Dufourg, N., ...
 328 Cabrolie, F. (2021, July). Automated multi-dataset analysis (AMDA): An
 329 on-line database and analysis tool for heliospheric and planetary plasma data.
 330 *Planetary and Space Science*, *201*, 105214. doi: 10.1016/j.pss.2021.105214
- 331 *HAPI Github Project Page*. (2021). Retrieved from [https://github.com/hapi-
 332 -server/](https://github.com/hapi-server/)
- 333 *HAPI Project Overview Page*. (2021). Retrieved from [http://hapi-server.org/
 334 HAPI Server List](http://hapi-server.org/). (2021). Retrieved from [http://hapi-server/servers/
 335 HAPI Server Verifier](http://hapi-server/servers/). (2021). Retrieved from <http://hapi-server.org/verify>
- 336 *Heliophysics Data Environment*. (2021). Retrieved from [https://hpde.gsfc.nasa
 337 .gov/](https://hpde.gsfc.nasa.gov/)
- 338 *IEEE 754 Standard*. (2021). Retrieved from [https://standards.ieee.org/
 339 standard/754-2019.html](https://standards.ieee.org/standard/754-2019.html)
- 340 King, T., Thieman, J., & Roberts, D. A. (2010, May). SPASE 2.0: a standard data
 341 model for Space physics. *Earth Science Informatics*, *3*(1-2), 67–73. doi: 10
 342 .1007/s12145-010-0053-4
- 343 Morley, S. K., Welling, D. T., Koller, J., Larsen, B. A., & Henderson, M. G. (2010).
 344 *Spacepy—a python-based library of tools for the space sciences* (Tech. Rep.). Los
 345 Alamos National Lab.(LANL), Los Alamos, NM (United States).
- 346 Mumford, S. J., Christe, S., Pérez-Suárez, D., Ireland, J., Shih, A. Y., Inglis, A. R.,
 347 ... others (2015). Sunpy—python for solar physics. *Computational Science &
 348 Discovery*, *8*(1), 014009.
- 349 *NASA Common Data Format*. (2021). Retrieved from [https://cdf.gsfc.nasa
 350 .gov/](https://cdf.gsfc.nasa.gov/)
- 351 Nativi, S., Caron, J., Davis, E., & Domenico, B. (2005, November). Design and
 352 implementation of netCDF markup language (NcML) and its GML-based
 353 extension (NcML-GML). *Computers & Geosciences*, *31*(9), 1104–1118.
 354 Retrieved from <https://doi.org/10.1016/j.cageo.2004.12.006> doi:
 355 10.1016/j.cageo.2004.12.006
- 356 Ponz, J., Thompson, R., & Munoz, J. (1994). The FITS image extension. *Astron-
 357 omy and Astrophysics Supplement Series*, *105*, 53–55.
- 358 *Protocol Buffers*. (2021). Retrieved from [https://developers.google.com/
 359 protocol-buffers](https://developers.google.com/protocol-buffers)
- 360 Rew, R., & Davis, G. (1990, July). NetCDF: an interface for scientific data ac-
 361 cess. *IEEE Computer Graphics and Applications*, *10*(4), 76–82. doi: 10.1109/
 362 38.56302
- 363 Richardson, L., & Ruby, S. (2007). *RESTful web services*. O'Reilly.
- 364 Roberts, D. A., Thieman, J., Génot, V., King, T., Gangloff, M., Perry, C., ...
 365 Hess, S. (2018, December). The SPASE Data Model: A metadata stan-
 366 dard for registering, finding, accessing, and using Heliophysics data obtained
 367 from observations and modeling. *Space Weather*, *16*(12), 1899–1911. doi:
 368 10.1029/2018sw002038

- 369 *SPEEDAS Space Physics Environment Data Analysis Software.* (2021). Retrieved
370 from <https://spedas.org/>
371 Weigelt, R. S., Lindholm, D. M., Wilson, A., & Faden, J. (2010, June). TSDS: high-
372 performance merge, subset, and filter software for time series-like data. *Earth*
373 *Science Informatics*, 3(1-2), 29–40. doi: 10.1007/s12145-010-0059-y

A Standard for Accessing Heliophysics Time Series Data

ORIGINALITY REPORT

4%

SIMILARITY INDEX

PRIMARY SOURCES

- 1 eprints.lancs.ac.uk 35 words — 1%
Internet
- 2 V. Génot, E. Budnik, C. Jacquy, M. Bouchemit et al. "Automated Multi-Dataset Analysis (AMDA): An on-line database and analysis tool for heliospheric and planetary plasma data", Planetary and Space Science, 2021 28 words — 1%
Crossref
- 3 T. M. Shaun Johnston, Shuguang Wang, Chia - Ying Lee, James N. Moum, Daniel L. Rudnick, Adam Sobel. "Near - inertial wave propagation in the wake of Super Typhoon Mangkhut: Measurements from a profiling float array", Journal of Geophysical Research: Oceans, 2020 14 words — < 1%
Crossref
- 4 "Headway in Spatial Data Handling", Springer Science and Business Media LLC, 2008 12 words — < 1%
Crossref
- 5 dblp.dagstuhl.de 12 words — < 1%
Internet
- 6 ntrs.nasa.gov 11 words — < 1%
Internet
- 7 D. Aaron Roberts, James Thieman, Vincent Génot, Todd King et al. "The SPASE Data Model: A 10 words — < 1%

Metadata Standard for Registering, Finding, Accessing, and Using Heliophysics Data Obtained From Observations and Modeling", Space Weather, 2018

Crossref

8 mafiadoc.com

Internet

8 words — < 1%

9 I. Gingell, S. J. Schwartz, J. P. Eastwood, J. E. Stawarz et al. "Statistics of Reconnecting Current Sheets in the Transition Region of Earth's Bow Shock", Journal of Geophysical Research: Space Physics, 2020

Crossref

7 words — < 1%

10 Takeshi Nogi, Satoko Nakamura, Yoshiharu Omura. "Full particle simulation of whistler - mode triggered falling - tone emissions in the magnetosphere", Journal of Geophysical Research: Space Physics, 2020

Crossref

7 words — < 1%

11 C. K. Louis, S. L. G. Hess, B. Cecconi, P. Zarka, L. Lamy, S. Aicardi, A. Loh. "ExPRES: an Exoplanetary and Planetary Radio Emissions Simulator", Astronomy & Astrophysics, 2019

Crossref

6 words — < 1%

EXCLUDE QUOTES ON

EXCLUDE MATCHES OFF

EXCLUDE BIBLIOGRAPHY ON