Precise, Actionable Reference to Dynamic Data: Recommendations of the Working Group on Data Citation (WGDC), Version: 2025-04-04T08:30Z

Summary of Changes

- New Title
- New section on Background and Context (II)
- Revised the language of the recommendations to clarify requirements (III). Some recommendations were combined into new recommendations (Rec. 1 and Rec. 12) and one implicit recommendation was made explicit (Rec. 3). Added an explanatory rationale statement for each recommendation.
- New Section on not-endorsed approaches (V)
- Updated adoption and FAQ sections (VI and VII).
- General edits for clarity from Working Group participants.

DOI: TBD

Authors: Andreas Rauber, Mark A. Parsons, and the Working Group on Data Citation.

Group Co-Chairs: Andreas Rauber, Mark Parsons.

Published: TBD

Abstract: These recommendations enable researchers and data infrastructures to identify and reference the precise data used in experiments and studies. Instead of providing static data exports or textual descriptions of data subsets, the recommendations support a dynamic, query centric view of data sets. They enable unambiguous identification of the precise subset and version of data used, supporting reproducibility of processes using data and the sharing and reuse of data. This document is an updated version of the recommendations published in 2015.

Keywords: data citation; dynamic data.
Language: English

Citation and Download: TBD

Previous Version: Rauber, A., Asmi, A., van Uytvanck, D., & Proell, S. (2015). Data Citation of Evolving Data: Recommendations of the Working Group on Data Citation (WGDC). Zenodo. https://doi.org/10.15497/RDA00016

I. Purpose

These recommendations enable researchers and data infrastructures to identify and reference the precise data used in experiments and studies. Instead of providing static data exports or textual descriptions of data subsets, the recommendations support a dynamic, query centric view of data sets. They enable unambiguous identification of the precise subset and version of data used, supporting reproducibility of processes using data and the sharing and reuse of data. This document is an update to the 2015 RDA Recommendation “Data Citation of Evolving Data”[1].

[1] Rauber, Andreas, Ari Asmi, Dieter van Uytvanck, and Stefan Proell. “Data Citation of Evolving Data: Recommendations of the Working Group on Data Citation (WGDC)”. Zenodo, October 20, 2015. https://doi.org/10.15497/RDA00016.

Goals of this WG are to create identification mechanisms that:

- allow us to identify and reference arbitrary views of data, from a single record to an entire data set in a precise, machine-actionable manner.
- allow us to identify and retrieve that data as it existed at a certain point in time, whether the database is static or dynamic.
- are stable and consistent across different technologies and technological changes and are applicable to all data types.

Solution: The WG recommends solving this challenge by:

- ensuring that data is stored in a versioned and timestamped manner.
- identifying data (sub)sets by storing and assigning persistent identifiers (PIDs) to timestamped queries that can be re-executed against the timestamped data store.

II. Background and Context

This document is an update to the 2015 RDA Recommendation "Data Citation of Evolving Data"[2]. The revised document addresses questions that have arisen over the years through the implementation of the specific recommendations described within the overall Recommendation. The changes do not affect the core requirements of the original recommendations; rather they clarify those recommendations. Some recommendations were combined into new recommendations (Recs. 1, 10 and 12) and two implicit recommendations were made explicit (Rec. 3 and Rec. 11). While there are now fewer recommendations, none of the original recommendations have been superseded.

The main point of clarification is that these recommendations focus on precise identification and reference to dynamic data. They do not address how these references are credited and recognized in scholarly systems. These recommendations address most elements of the Joint Declaration of Data Citation Principles (Evidence, Unique Identification, Access, Persistence, Specificity and Verifiability ,Interoperability and Flexibility)[3], but, they do not fully address the Importance principle or the Credit and Attribution principle because these are very social and cultural considerations. We have correspondingly retitled the document "Precise, Actionable Reference to Dynamic Data" recognizing that it does not address all aspects of data citation and that it also has other applications such as facilitating and tracing automated workflows or tracking precise provenance. The detailed recommendations are fundamental requirements for data archiving and stewardship of dynamic data.

Several other related RDA Working Groups and Recommendations have arisen since the 2015 version was published that address additional aspects and applications of citation. Data citation continues to be an evolving issue. Related Working Groups and Recommendations include but are not limited to:

Related Working Groups:

- Artificial Intelligence and Data Visitation (AIDV) WG https://www.rd-alliance.org/groups/artificial-intelligence-and-data-visitation-aidv-wg/
- Complex Citations Working Group https://www.rd-alliance.org/groups/complex-citations-working-group/
- Data Granularity Working Group https://www.rd-alliance.org/groups/data-granularity-wg/

---

[2] Rauber, Andreas, Ari Asmi, Dieter van Uytvanck, and Stefan Proell. "Data Citation of Evolving Data: Recommendations of the Working Group on Data Citation (WGDC)". Zenodo, October 20, 2015. https://doi.org/10.15497/RDA00016.

[3] Data Citation Synthesis Group: Joint Declaration of Data Citation Principles. Martone M. (ed.) San Diego CA: FORCE11; 2014 https://doi.org/10.25490/a97f-egyk

- Data Versioning Interest Group (formerly Working Group)
  https://www.rd-alliance.org/groups/data-versioning-ig/

Related Recommendations:

- ICSU-WDS & RDA Publishing Data Services WG Interoperability Framework Recommendations. https://doi.org/10.15497/RDA00002
- RDA Research Data Collections WG Recommendations. https://doi.org/10.15497/RDA00022
- RDA Data Usage Metrics WG Recommendations. https://doi.org/10.15497/RDA00062
- RDA/TDWG Attribution Metadata Working Group: Final Recommendations. https://www.rd-alliance.org/group_output/rda-tdwg-attribution-metadata-working-group-final-recommendations/

III. Recommendations

To realise the goal of precisely identifying arbitrary subsets of data, from single values to entire data holdings in settings that range from static data to highly dynamic data streams for any type of data[4], the WG recommends the adoption of the following actions:

A. Preparing the Data and the Query Store

Prepare data sources and provide the required infrastructure, which is needed for implementing the query based approach as follows:

1: Versioning and Timestamping: IF data is evolving AND earlier states of the data should be reproducible , THEN the infrastructure MUST version changes to the data. It further MUST assign a timestamp of when a change becomes visible in the system.

Rationale: If data changes and earlier states of the data should be reproducible, then these earlier states must be retained. By storing with each update/change to the data set the time when this change becomes visible in the system, the state of the data can be referred to at any arbitrary point in time, i.e. each update (addition/deletion/correction) becomes immediately usable and reproducible. Timestamps allow to refer to the state of data as it existed in the system at any given point in time. Semantic labels (as used in semantic versioning) may be added as assertions on the state of data at any such point in time but are not a suitable versioning approach in itself.

2 – Query Store Facilities: The infrastructure MUST provide means for storing queries and the associated metadata, specifically an execution timestamp, to allow re-execution against the then-valid state of the data source .

Rationale: Retrieving a data set requires the re-execution of the time-stamped query against the time-stamped data source. As the data infrastructure may change over time, queries may need to be migrated to work with a new data representation. Hence, the queries must be stored with the data infrastructure and not externally e.g. with the user. Furthermore, security considerations may motivate a data infrastructure not to release the actual query string/code but simply store it internally.

B. Persistently Identify Specific Data Sets

When a user selects (a subset of) data, this is referred to as a query executed against the data store. IF such a subset should be identifiable persistently, a data infrastructure SHOULD apply the following steps:

---

[4] Of course, it is technically impossible to declare a solution for any type of data, but we have found no exceptions to date. Special challenges with very rapidly changing data can be addressed with bulk-writing updates, or overwriting updates without read-access in-between - or limiting read access to certain time intervals, etc.

3: Limit query functionality: A data infrastructure SHOULD limit the query to functions that can be deterministically re-computed across different technical infrastructure implementations.

Rationale: Data infrastructures and query execution environments will evolve over time. To ensure precise reproducibility any function used as part of a query needs to have a deterministic implementation. While classical select/project functions can be precisely re-computed, advanced functions such as computing similarity scores, averages etc. may depend on the precise implementation and numerical accuracy of the query execution environment and should thus be avoided and delegated to the subsequent data processing. A query may be any form of data access function such as SQL queries selecting rows and columns in a database, a bounding box specification or color profiles selecting an area in an image, a selection of files in a file system via filename properties, slice and dice operators in multidimensional data cubes, start and end points in data / video streams, sub-graph selections in a graph data structure, etc.

4 – Query PID: A data infrastructure MUST assign a new PID to the query if either the query is new or if the result set returned from an earlier identical query is different due to changes in the data. Otherwise, return the existing PID of the identical query.

Rationale: Subsets of data are identified via the queries used to retrieve them. Whether two subsets are identical is determined both by the query and the result set. Two queries may retrieve an identical subset but be semantically different. Thus, a new PID must be minted if two queries are semantically different or if two queries are identical but return different results because the underlying data has changed. Determining whether two queries are semantically equivalent may not be trivial when the same semantics may be expressed indifferent ways. Queries may be re-written to a normalized form in order to help with determining semantic identity. *In case of doubt, two query representations may be assigned different PIDs in spite of being semantically equivalent. Not every query processed needs to be persisted: queries can be temporarily collected in a staging area where users may decide which queries they want persisted and which ones can be discarded.*

5 – Query Timestamping: If the query received a new PID, the data infrastructure MUST assign a timestamp to the query based on the time the query was executed (or the last update to the data). This allows retrieving the data as it existed at the time a user issued a query.

Rationale: By identifying the (local) execution timestamp of the query, it can be re-executed against that specific state of the data store in the future. As the precise execution timestamp of a query may reveal privacy-sensitive information on when exactly a query was issued, a data infrastructure may decide to, instead, use the timestamp of the last global update to the data or the last update to the data set affected by the query to provide a timestamp at a coarser level of granularity.

6 – Consistent Sorting: The data infrastructure SHOULD ensure that the sorting of the elements in any data set returned is unambiguous and reproducible. IF consistent sorting is not guaranteed the data infrastructure MUST be explicit about that.

Rationale: While the results of any query execution are deterministic (Rec. 3), the individual elements are not necessarily sorted in an identical manner. As the result of subsequent processing steps may depend on the order of the data, a data infrastructure should ensure to apply a deterministic sorting to all records/elements of data returned before applying any user-defined sorting criteria to ensure that the result sets are always sorted in the same sequence upon re-executions. This is also needed to ensure fixity information can be computed and matched.

7 – Result Set Verification: The data infrastructure SHOULD compute fixity information (checksum) of the query result set to enable verification of the correctness of a result upon re- execution.

Rationale: Data sets may become irreproducible, e.g. when data needs to be physically and permanently deleted because of legal obligations or due to operational decisions of a data infrastructure. Fixity information allows infrastructures to detect such a situation when a query re-execution does not return the identical data as the original execution. (Requires Rec. 6 to be applied or inconsistent sorting to be considered in the fixity computation). In

exceptional circumstances when computing fixity information would be computationally prohibitive or incur prohibitive cost, infrastructures must identify a suitable surrogate such as computing fixity information on record/item/data instance metadata (e.g. row/column headers such as PIDs and attribute names, file-ID, modified timestamp and filesize or individual file fixity information in object stores/file systems, ...)

8 – Query Persistence: The data infrastructure MUST persistently store the query and associated metadata (e.g. PID, original and normalized query, fixity information (if available), timestamp, database PID, data set description, and other information) in the query store associated with the data queried and MUST assume responsibility for re-executing the query to reproduce the dataset when required to do so.

Rationale: The queries must be maintained with the data infrastructure so that queries can be migrated when a data infrastructure should decide to change its data representation.

9 – Citation Texts: The data infrastructure SHOULD generate citation texts in the format prevalent in the designated community to encourage scholarly citation of the data (e.g. Authors, Date, Dataset Title. PID).

Rationale: These recommendations aim predominantly at identifying (subsets of) data in a machine actionable form so that the identified data can be used directly (e.g. as an input parameter to a data processing pipeline). However, such citation texts also should encourage proper scholarly data citation. Providing recommended citation text snippets in different formattings prevalent in the community supports this.

C. Resolving PIDs and Retrieving the Data

10 – Landing Page: PIDs MUST resolve to a human readable and machine-actionable landing page that provides a mechanism for accessing the data (via PID query re-execution) and metadata. The landing page MAY reveal the actual query being executed, as well as the associated timestamps and other metadata depending on client needs, and security and privacy concerns.

Rationale: The landing page must support a standardized access protocol used by the designated community to allow machines to access the data directly (thus allowing the PID to be used as input parameter in subsequent data processing pipelines) as well as any metadata required to make informed use of the data, including license information, access permission requirements, etc.

11 - Data Access: Data access MUST include the possibility to re-run the timestamped query against the accordant state of the data store. It SHOULD further provide the option to re-run the query against the current state of the data store and retrieve the semantically equivalent dataset but benefiting from all additions and corrections made (leading to a new PID being assigned if the result set differs). It MAY further offer the opportunity of retrieving a difference set of the changes that were made between the original execution timestamp and the current state, or re-executing the query against any arbitrary state(i.e. timestamp) of the data. The data infrastructure SHOULD verify the correctness of the re-execution using the fixity information stored with the original query execution and inform the client accordingly.

Rationale: The queries provide the means to re-create the accordant data set. Whether this re-creation process is made transparent by the data infrastructure or simply hidden behind a "download" functionality is up to the data infrastructure to decide, as is the integration of any access regulations/controls.

D. Upon Modifications to the Data Infrastructure

12 – Technology Migration: When data is migrated to a new representation (e.g. new database system, a new schema or a completely different technology), the data infrastructure MUST migrate also the queries and associated fixity information and verify the correctness of the migration.

Rationale: We may expect any data representation to change at some point in time in the future. When migrating all data to a new representation, potentially requiring a different query mechanism, then the data infrastructure is

responsible for ensuring that queries that need to be persistently kept can be re-executed. This will usually require the new data representation and access logic to have at least the same representational power and data granularity. IF this is not possible or desired, then the conscious decision may, of course, be taken, that past subsets are no longer reproducible or that different mechanisms may be foreseen for previous data representations, similar to any decommissioning / re-appraisal effort in archival settings and following the infrastructure's policies with respect to data deletion.

IV. Benefits

The proposed solution has several benefits compared to current approaches relying on individual data exports for each data set or ambiguous natural language descriptions of data set characteristics.

- It allows identifying, retrieving and citing the precise data set with minimal storage overhead by only storing the versioned data and the queries used for creating the data set. In many environments data versioning is considered a best practice. Data subsets can be re- created on demand.
- It allows retrieving the data both as it existed at a given point in time as well as the current view on it, by re-executing the same query with the stored or current timestamp, thus benefiting from all corrections made since the query was originally issued. This allows tracing changes of data sets over the time and comparing the effects on the result set.
- The query stored as a basis for identifying the data set provides valuable provenance information on the way the specific data set was constructed, thus being semantically more explicit than a mere data export.
- The query store offers a valuable, central basis for analyzing data usage.
- Metadata such as checksums support the verification of correctness and authenticity of data sets retrieved.
- The recommendations are applicable across different types of data representation and data characteristics (big or small data; static or highly dynamic; identifying single values or the entire data set).
- If data is migrated to new representations, the queries can also be migrated, ensuring stability across changing technologies.
- Distributed data sources can rely on local timestamps at each node, avoiding the need for expensive synchronization in loosely coupled systems.

V. Not Endorsed Approaches

- A data infrastructure MUST NOT store redundant data exports in lieu of the present recommendations. These MAY still be used as an additional precaution, for continuing current practices. (*Rationale: Storing redundant data exports is both cumbersome from a data management perspective and wastes valuable storage space. It also delays any reproducible use of current data until the next snap-shot has been exported. "Recommended" or "stable snapshots" etc. may still be identified via the query-based mechanisms by storing an according query executed at given time intervals and applying an according assertion as semantic label.)*

- A data infrastructure SHOULD NOT apply semantic versioning to persistently store major/minor revisions etc. Any semantic qualification SHOULD be phrased as semantic annotations or assertions against the data source. (*Rationale: For data, the difference between major and minor changes commonly used in semantic versioning does not exist. The semantics of a change to the data is entirely dependent on the intended use of the data set. Something that might be considered a "minor change" for one type of use may have a major impact for other types of uses. Correcting a single typo in a record, or removing a single incorrect value or pixel may render work that has been produced downstream irreproducible. Hence, all changes need to be treated equally, with the impact of any such change depending on the subsequent and unforeseeable use of data. Semantic labels MAY still be assigned to specific states of the data as assertions in the metadata, and may be used for citation - but SHOULD NOT be used for identification with respect to the semantics they carry.)*

- A data infrastructure SHOULD NOT apply a change log approach to implement the versioning to ensure queries can be re-executed efficiently. (*Rationale: applying a log-based approach will require a data*

infrastructure to be halted, applying a roll-back to re-create an earlier state of a data store, making the re-execution of a query against an earlier time stamp cost-intensive, blocking continuous operation. Such approaches SHOULD be avoided after careful consideration of existing options.)

VI. Current Adoption

The International Organization for Standardization (ISO) recommends the use of these Recommendations in ISO690: "Guidelines for bibliographic references and citations to information resources".[5],

The Earth Science Information Partners (ESIP) encourages the application of these recommendations in their "Data Citation Guidelines for Earth Science Data".[6]

Multiple repositories have implemented some or all of the recommendations. Descriptions and lessons learned are documented in a review paper[7] and in the notes and slides of the Working Group.

VII. Frequently Asked Questions

- May data be deleted? Yes, given appropriate policies. Queries may then not produce the same result set when re-executed. Landing pages should persist.
- Does the system need to store every query? No. Only data sets that should be persisted for citation / later re- use need to be stored. Persisting queries can be decided individually or policy-based in an automated fashion.
- Can I obtain only the most recent data set? Queries can, in principle, be re-executed with the original timestamp or with the current timestamp or any other timestamp desired. This allows retrieving the semantically identical data set but incorporating all changes, corrections or updates applied before the given timestamp.
- Which PID system should be used? Any PID system can be applied according to the institutional policy.
- How are the queries created? Queries can either be created manually via an interface/workbench or applications create the proper queries automatically. Any form of data access ultimately leads to a query, ranging from a single file access (retrieving the file's segments from a storage device and re-assembling them), to bounding boxes or slice/dice operations in multidimensional data cubes. All require the adaption of the query by adding metadata and timestamps.
- How can I share parts of my database? The query centric view allows selecting any particular view or data subset of the data from the complete data set, such as a time-delay embargo providing a view of the data exposing all changes made prior to a certain point in time.
- How does this support giving credit and attribution? Attribution and giving credit is supported via a provenance chain from a subset/view of data to the data set it was derived from and the data infrastructure hosting it, allowing to document intellectual contributions on the way. Analysis and recommendations on how to aggregate bibliometrics and credits is not addressed in the context of this WG.

A more extensive, peer-reviewed FAQ is included in the review paper.[7]

---

[5] ISO 690:2021, Information and documentation| Guidelines for bibliographic references and citations to information resources (International Organization for Standardization (ISO), 2021), Sec. 8.13.3.8, p131

[6] ESIP Data Preservation and Stewardship Committee. 2019. Data Citation Guidelines forEarth Science Data. Ver. 2. Earth Science Information Partners. https://doi.org/10.6084/m9.figshare.8441816

[7] Rauber, A., Gößwein, B., Zwölf, C. M., Schubert, C., Wörister, F., Duncan, J., Flicker, K., Zettsu, K., Meixner, K., McIntosh, L. D., Pröll, S., Miksa, T., & Parsons, M. A. (2021). Precisely and Persistently Identifying and Citing Arbitrary Subsets of Dynamic Data. *Harvard Data Science Review*, *3*(4). https://doi.org/10.1162/99608f92.be565013

VIII. Maintenance Plan

The Working Group on Data Citation will continue regular updates and presentation of adoption stories at future RDA meetings. The group will also inform relevant standards bodies (see above) of the new version.

IX. Next Steps

The WG continues ongoing evaluation in a series of operational implementations in different domains. We encourage interested community members to participate and provide improvements, comments, suggestions, and general feedback via the working space of the WG[7]. We are very interested in further real world use cases to act as demonstrations.

X. Get Involved

You can find additional information on the RDA Working Group Page. Please register on the mailing list to stay informed. The community feedback is collected in the group web page[8].

[8] https://www.rd-alliance.org/groups/data-citation-wg/