Simple, Standards-Based Archiving in Dataverse

Sebastian Karcher, Jim Myers, Sebastian Ostrowski, Nic Weber
Qualitative Data Repository (https://qdr.syr.edu)

QUALITATIVE DATA REPOSITORY

Introduction

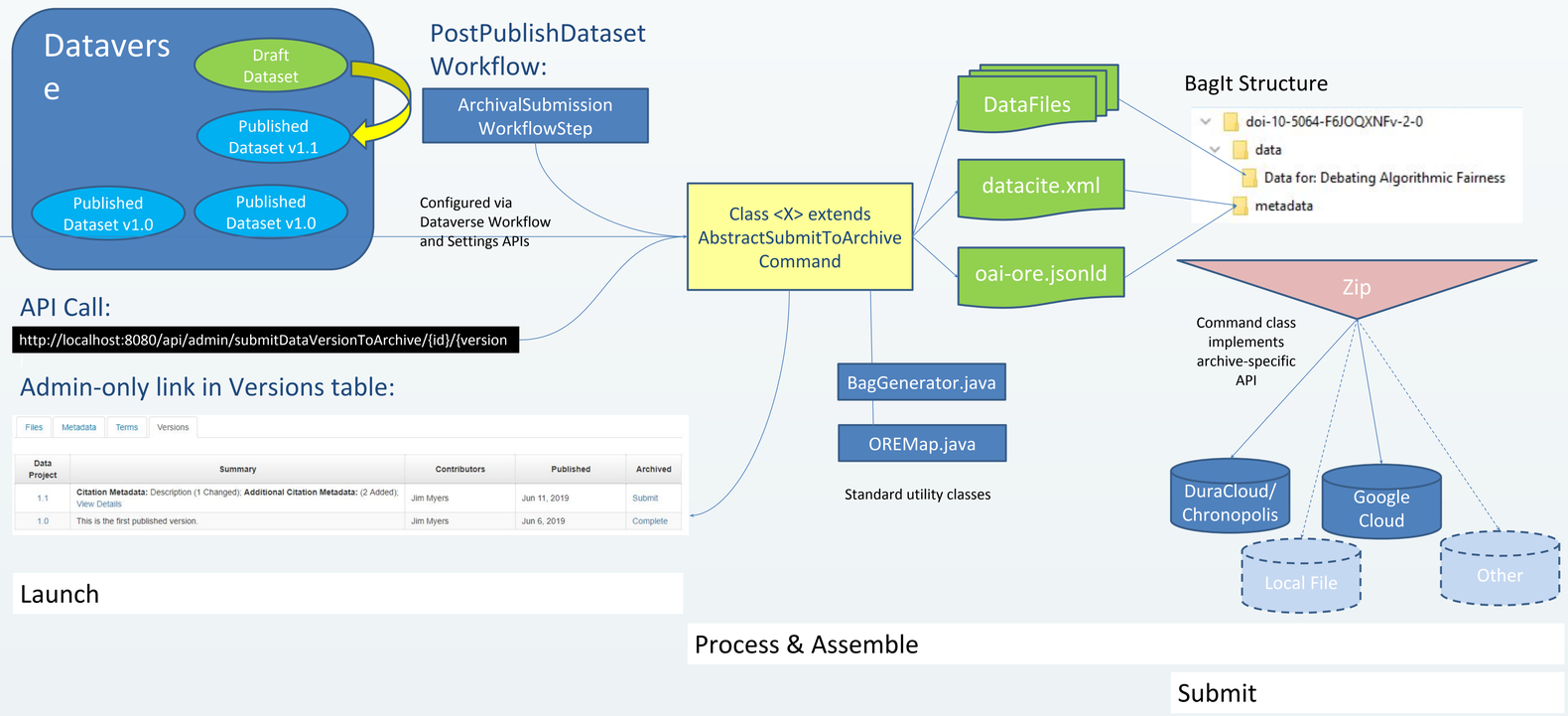
Archival copies serve a variety of purposes, from simply acting as an additional back-up copy to assuring that data and metadata can be read and understood far into the future, when the original software stack used to create a dataset may no longer exist. They can also serve as a verifiable snapshot of specific dataset versions, allow independent management of individual datasets, and serve as an export/transfer mechanism.

We describe the work done to implement an extensible, standards-based archiving mechanism within Dataverse and to support archiving to DuraCloud/Chronopolis and Google Cloud Storage. This work includes enhancing Dataverse's workflow mechanism, associating metadata blocks with external community vocabularies using the Object Reuse and Exchange (ORE) format, developing an extensible 'Submit To Archive' Command class, and adapting work from the SEAD DataNet project to create self-describing zipped archive files (BagIt) following the recommendations of the Research Data Alliance Research Data Repository Interoperability Working Group.

These contributions to Dataverse provide an archiving mechanism that is simple to configure and easy to extend to support other archives. They produce a single archive file per Dataset version that is human and machine readable, providing a complete copy of research published through Dataverse that can be preserved using any repository providing file or object (e.g. S3) storage.

Process Overview

Archiving can be configured to occur automatically as part of publication, scripted via the API, or managed manually by an administrator. In the current implementation, each version of a Dataset is archived as a single zip file (accompanied by a redundant datacite.xml file to simplify discovery). The zip file contains all DataFiles and all unique metadata (not derivable from the file contents) for the Dataset organized using multiple standards and recommendations as discussed below. Configuration using Dataverse's Settings and Workflow APIs is all that is needed to submit to DuraCloud or Google. Supporting alternate archives involves extending one class to send the zipped file to the archive using its specific API.

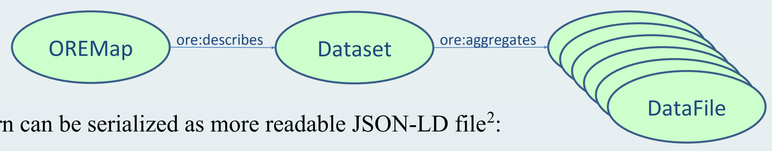


Standards for Archive Files

An important goal in archiving is to enabled stored data and metadata to be accessed and interpreted without the original software that created it. In developing an archival format for Dataverse, we've leveraged multiple standards and recommendations including those outlined below.

Open Archives Initiative Object Reuse and Exchange (OAI-ORE)[1]

The OAI-ORE standard defines a *Map* that *describes* an *Aggregation* that *aggregates* a set of *AggregatedResources*, all of which are described by global identifiers (URIs). Each of these entities can then be annotated with metadata from other vocabularies.



An OAI-ORE Map can be represented in RDF, which in turn can be serialized as more readable JSON-LD file[2]:

```
{
  "dcterms:modified": "2019-04-16",
  "dcterms:creator": "Qualitative Data Repository",
  "@type": "ore:ResourceMap",
  "@id": "https://data.qdr.syr.edu/api/datasets/export
          ?exporter=OAI_ORE&persistentId=doi:10.5064/F6JOQXNF",
  "ore:describes": {
    "@id": "doi:10.5064/F6JOQXNF",
    "@type": [ "ore:Aggregation", "schema:Dataset" ],
    "schema:version": "2.0",
    "schema:datePublished": "2019-04-16",
    "schema:name": "Data for: Debating Algorithmic Fairness",
    "schema:dateModified": "2019-04-16 16:43:12.691",
    "Subject": [ "Law", "Social Sciences" ],
    "Creator": {
      "author:Name": "Hamilton, Melissa",
      "author:Affiliation": "University of Surrey",
      "Identifier Scheme": "ORCID",
      "ORCID": "0000-0002-8593-0017"
    },
   "ore:aggregates": [
       {
         "schema:name": "Hamilton_ATI Data Overview.pdf",
         "@id": "doi:10.5064/F6JOQXNF/6FH1XU",
         "dvcore:restricted": false,
         "schema:version": 1,
         "dvcore:categories": [ "Documentation" ],
         "schema:fileFormat": "application/pdf",
         "dvcore:filesize": 259104,
         "dvcore:checksum": {
           "@type": "SHA-512",
           "@value": "03c0821651590e10abe961f8774bd04973ee00157f7
                      f48725abda12b8dbadb89c306c337a22a3ec2fb536a
                      1805084fb01dac0aeed979754ca8c3278f3a79c18d"
         },
         …
       }],
  "@context": {
    "Creator": "http://purl.org/dc/terms/creator",
    "ore": "http://www.openarchives.org/ore/terms/",
    "schema": "http://schema.org/",
    "dvcore": "https://dataverse.org/schema/core#",
    ...
```

Information about the Map

Information about the Dataset

Nested Information about the Dataset's Creator

Information about each DataFile

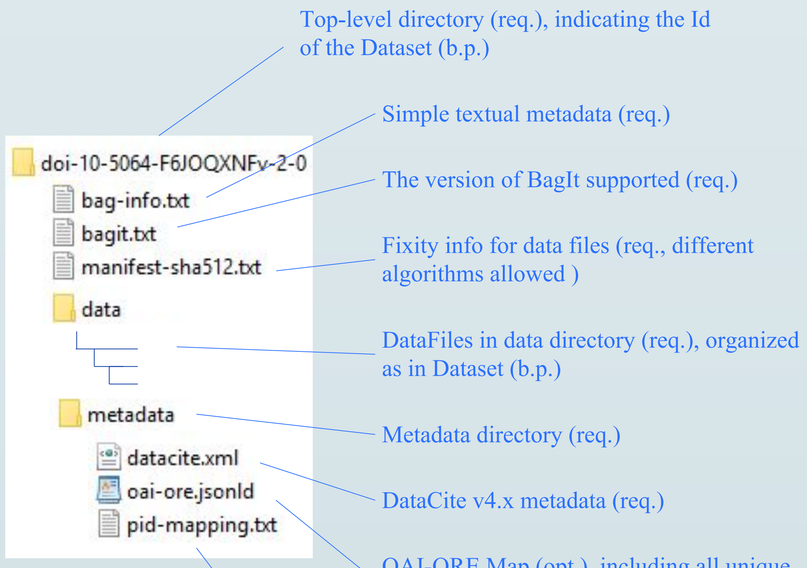
Mapping of each term/prefix to a formal vocabulary

BagIt

The BagIt standard[3] defines a hierarchical structure for storing data and metadata files for preservation and a few mandatory files providing basic descriptive metadata and fixity information.

Research Data Alliance Research Data Repository Interoperability (RDRI) Working Group

RDRI recommends[4] using the BagIt and DataCite (v4+) metadata standards and includes OAI-ORE Maps as choice for metadata. It further defines a standard location and naming scheme for metadata files. Together with best practices (b.p.) developed by projects participating in RDA, these provide useful standardization beyond the core BagIt standard.



Dataverse Enhancements

Providing archiving within Dataverse required a wide range of enhancements, many of which are useful on their own as well:

- **Choice of hash algorithm used for fixity checksums**: admins can select MD-5, SHA1, SHA256, or SHA512. An API call will validate checksums created with one algorithm and generate ones using the new algorithm.
- **Mapping of metadata block terms to external vocabularies**: An additional column in the .tsv files used to define metadata blocks allows admins to map all terms in a block, or specific terms, to matching terms in external vocabularies (e.g. Subject -> http://purl.org/dc/terms/subject).
- **Expanded Workflow functionality**: Workflow steps can now access Dataverse settings and/or an API Key, as specified in their admin-controlled configuration, and have up-to-date information about datasets (e.g. when configured as post-publication workflows).
- **OAI-ORE Map Metadata Export format**: The Map file included in the archive is also available directly as one of the Metadata Export options for published Datasets.
- **Efficient Bag creation**: The utility class added to Dataverse for BagIt Bag creation uses multiple threads to directly assemble the zipped Bag and can stream it to external archives minimizing creation time and memory/storage use.
- **Extensible archiving framework**: Base classes created to enable archiving to be triggered by publication, scripted via the API, or manually run by an admin can be extended to target any archive. These classes can leverage the utilities to generate OAI-ORE Maps and zipped Bags but are also free to create alternate/additional export formats. Admins can use Dataverse settings to configure which archive to use.
- **DuraCloud/Chronopolis and Google Cloud Storage support**: QDR has created classes that send zipped BagIt archive files to DuraCloud (from which content can then be sent into Chronopolis) or Google's Cloud Storage (including the least-expensive 'cold' storage).

Conclusions and Future Work

The core archiving functionality presented here was released with Dataverse v4.11. Some newer features (Google Cloud Storage, folders and provenance in OAI-ORE Map and Bags, admin GUI) are in production use at QDR but are not yet in a Dataverse release. The following notes assess how well the current functionality aligns with the goal of long-term archiving and where potential future work could add value:

Concern	Current Status	Future Options
Manageability	A single-file-per-dataset solution that supports fixity checks, automated and manual submission, no archive requirements beyond file/object storage	No support for batch operations or validation from within Dataverse
Completeness	Includes all unique metadata from Dataverse except provenance files	The OAI-ORE Map and Bag could include provenance files and derived metadata (e.g. variable-level metadata from tabular files)
Versioning	Dataset versions are archived independently	Support for external data references in the BagIt specification could allow later versions to reference DataFiles from earlier versions to minimize storage requirements
Interoperability and Import	Uses current standards and best practices	The mapping of Dataverse terms to external vocabularies could be improved over time. Tools such as the DVUploader[6], which was originally designed to read OAI-ORE/BagIt archives could be adapted to re-import archival Datasets into Dataverse or other repositories
Long-term Access	Archive files must be downloaded from archives and unzipped	Software (<2K lines) developed by the SEAD project (https://sead-data.net/) provides DOI landing pages and access to metadata and individual data files via the Web from OAI-ORE/BagIt archives

Acknowledgements & References

QDR is funded by the National Science Foundation with additional support for research from the Alfred P. Sloan Foundation, and hosted by the Center for Qualitative and Multi-Method Inquiry, a unit of the Maxwell School of Citizenship and Public Affairs at Syracuse University. The authors thank the Dataverse Team and Community for helpful discussions and support in integrating this work into Dataverse.

References:

[1]Open Archives Initiative Object Reuse and Exchange, https://www.openarchives.org/ore/
[2]ORE User Guide - Resource Map Implementation in JSON-LD, http://www.openarchives.org/ore/0.9/jsonld
[3]The BagIt File Packaging Format (V1.0), https://tools.ietf.org/html/draft-kunze-bagit-17
[4]Research Data Repository Interoperability WG Final Recommendations, https://www.rd-alliance.org/group/research-data-repository-interoperability-wg/outcomes/research-data-repository-0
[5]Package Serialization Using BagIt, https://releases.dataone.org/online/api-documentation-v2.0.1/design/DataPackage.html
[6]DVUploader, a Command-line Bulk Uploader for Dataverse, https://github.com/IQSS/dataverse-uploader