A Whirlwind Tour of Some Community-supported Features

Jim Myers

jim.myers@computer.org

European Dataverse Workshop, January 23-24, 2020

Simple Standards-Based Backup/Archiving

- Define a package describing a version of a dataset, with all files and metadata as a single zip file
- Enable such packages to be transferred to an external long-term store as part of publication or on-demand via API
- TBD: Enable such a zip file to be re-imported to recreate a dataset

- Standards Involved
  - **BagIt**
  - Open Archives Initiative Object Reuse and Exchange (**OAI-ORE**)
  - RDA's Research Data Repository Interoperability (**RDRI**) Recommendations
  - **JSON-LD**
  - Community Vocabularies, e.g. **Schema.org, Dublin Core**

Simple?

Basic text description

Fixity info for data files

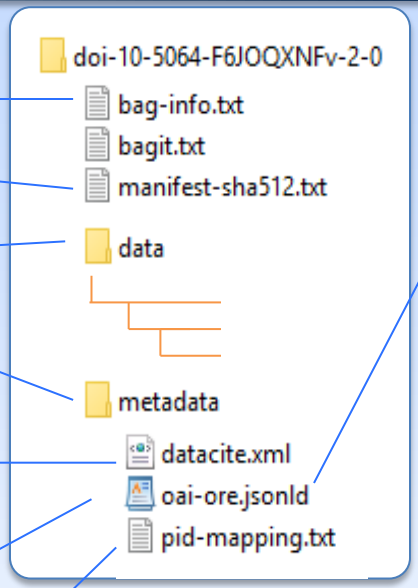
Data directory

Metadata directory

Minimal Structured Metadata

Complete Semantic Metadata

File ID to location map



- Key/Value pairs about the package, the Dataset, and each Datafile
- Using community vocabularies (any)
- In a readable format

Example section for dataset metadata:

```
{
  ...,
  "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"
    },
    ...
  },
  "@context": {
    "Creator": "http://purl.org/dc/terms/creator",
    "ore": "http://www.openarchives.org/ore/terms/",
    "schema": "http://schema.org/",
    ...
```

Just the info we need, arranged so that people not familiar with Dataverse can read it, and so that Dataverse can potentially read packages from other repositories…

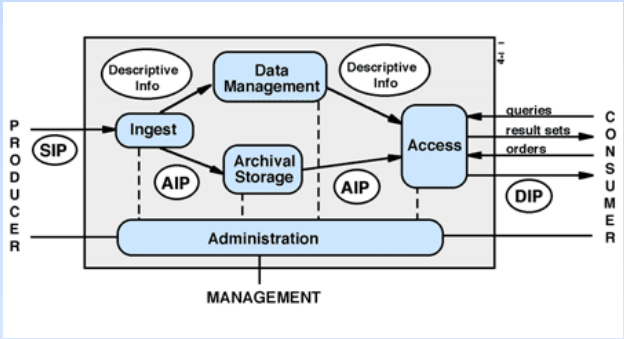
In Practice:

- Configured through Dataverse properties
- Three storage options developed:
  - Local File system -> iRODS
  - Duraspace -> Chronopolis
  - Google Cloud Storage (not yet in a release)
- Simple API/Classes to develop more
- Vocabulary mappings drawn from metadata block files (e.g. citation.tsv)
- Any version can be archived via the API
- Can configure a post-publication workflow to automatically archive any Dataset version as it is published.

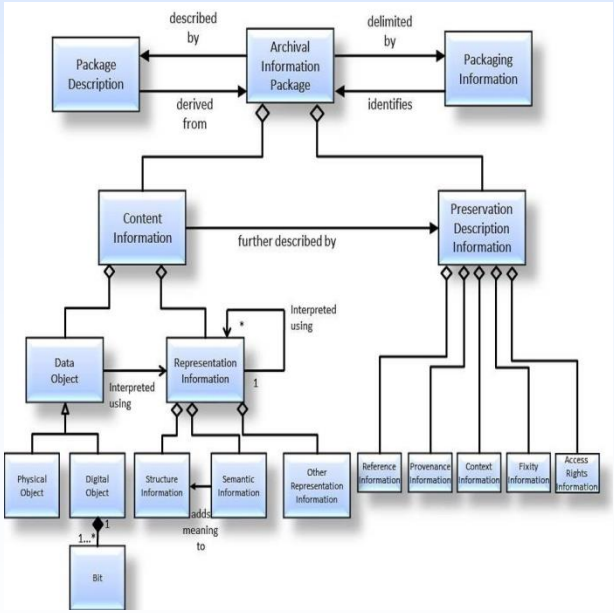
- Work in progress via RDA Grant:
  - Add any provenance files and variable-level metadata
  - Develop re-import capability
  - Test interoperability with zipped Bags created by other repositories

Open Archival Information System (OAIS)?

- Caveat Emptor!
- Nominally the zipped Bags from this work can serve as an Archival Information Package (AIP) within the OAIS model, with Dataverse managing the Submission Information Package (SIP). Importing an AIP into Dataverse again could be a way of providing Dissemination Information Packages (DIP).
- To my knowledge, no one has fully reviewed the Bag contents or export/import processes to assess them in terms of the OAIS model, although some such work has begun.
- An alternative approach – Archivematica (https://www.archivematica.org/) , which now has an ability to ingest from Dataverse was designed to support the OAIS model.



From https://nssdc.gsfc.nasa.gov/nssdc_news/dec00/oais.html



From http://www.oais.info/

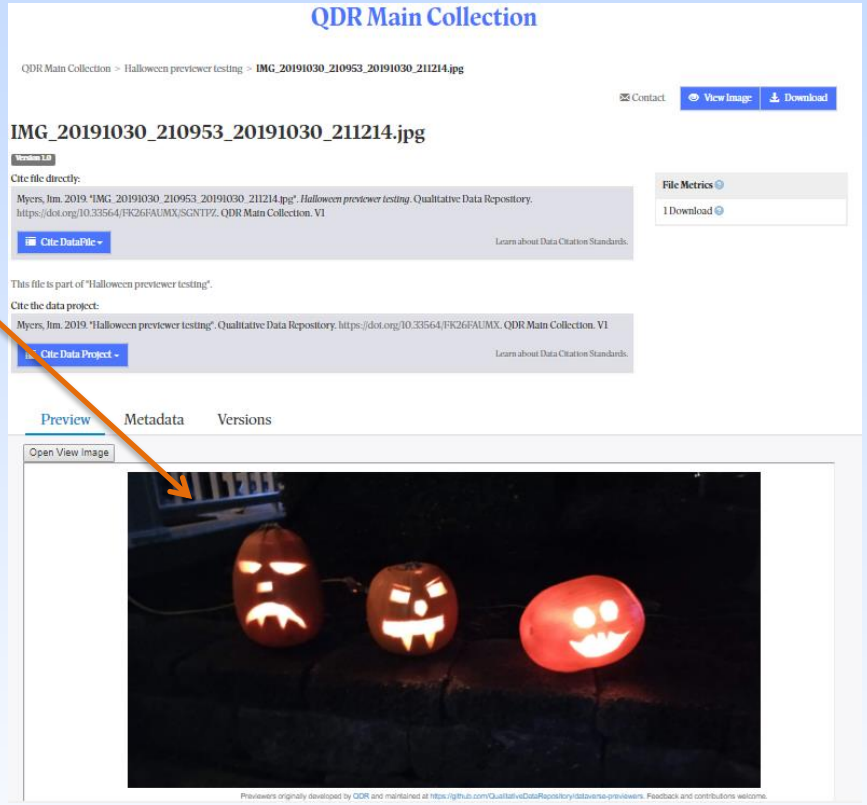
Data Previewers

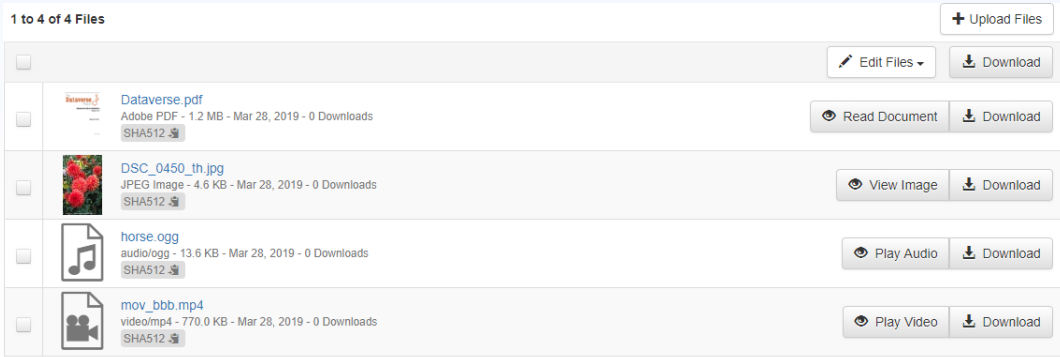
Your data format here!

Dataverse's 'external tools' API provides a way to pass data and metadata to other tools…

Including web applications that can provide previews for images, videos, audio, documents, spreadsheets, etc.!

Developers: If you see file previews on the web, you can integrate them with Dataverse!





Using Existing Data Previewers

- See https://github.com/QualitativeDataRepository/dataverse-previewers
- Use Dataverse's API call to register previewers running at github.io for the right mime-types
  - (cut/paste the curl commands)
  - Allow cross-site requests (CORS) for your Dataverse and S3 store (if using direct downloads)

- Optionally, download them to run locally

Creating new Data Previewers

- Copy an HTML template for your mimetype and write a short Javascript function:
  - The complete audio previewer function

```
$(document).ready(function() {
   startPreview(false);
});

function writeContent(fileUrl, file, title, authors) {
   addStandardPreviewHeader(file, title, authors);
   $('.preview').append($("<audio/>").attr("controls","").attr("src",fileUrl));
}
```

  - Or, if you need to get the file contents

```
startPreview(true);
…
function writeContentAndData(data, fileUrl, file, title, authors) {
}
```

DVUploader – batch file uploads

- A Java program that uses the Dataverse API to upload a file(s)/a directory tree of files to Dataverse:

```
java -jar DVUploader-v1.0.7.jar -key=<api key> -did=<dataset doi> -server=<server URL> <dir/file(s)>
```

  - Many options including:
    - -recurse – go into subdirectories
    - -ex=<pat> - ignore files with the specified pattern
    - -verify – check fixity
    - -limit=<x> – max number to process
    - -directupload – use new direct-upload-to-S3-capability (stay tuned)
  - Useful For:
    - Automating uploads
    - Many files
    - Incremental uploads (will upload new files in a dir)

DVUploader – Limitations/Plans

- Serial uploads
- Each file triggers dataset update (performance hit)
- Will soon support upload of Dataverse Bags to create/re-create Datasets

- Can also upload to NCSA's Clowder and will be part of the interoperability demonstration in the RDA grant

Make Data Count!

- MDC is a standard way of reporting metrics

- IQSS implemented the core mechanism
- Issues:
  - Setup is ~complex, involving a separate program and cron jobs
    - TDL has example cron jobs, planning to submit PRs
  - Minor bugs in creating MDC logs - have been fixed
  - MDC resets counts to zero
    - Design discussions to display prior counts along with MDC metrics
    - Implemented a separate switch in 4.18 to allow logging to start before processing is set up
- **Bottom line:**
  - New repositories could set up MDC completely now
  - Existing repositories should at least turn on logging (when at 4.18)

Multiple Stores

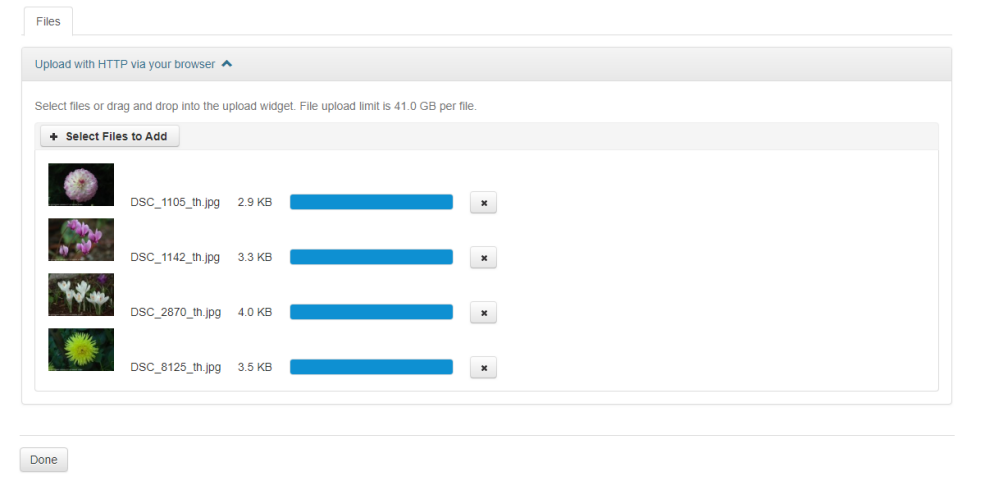
- Dataverse currently supports one store that can be a file system, or an S3 or SWIFT object store.
- An update now in review will allow multiple stores
- Use Cases:
  - Add a store without moving existing data
  - Different cost/performance tradeoffs for big/sensitive data
  - Stores managed by different stakeholders

Multiple Stores - Configuration

- Uses JVM options as current store
  - Each store gets a name/id/type(file, s3, swift)
  - Current store options apply to specific store, e.g.
    - dataverse.files.s3.bucket-name='bucket-one'
    - dataverse.files.otherS3.bucket-name='another-bucket'
- One store is the default, others can be assigned per dataverse (by a superuser)
- Making the name of the current store match its type allows backwards compatibility
  - Current datafile entries in the db would not change

Direct S3 Upload – a simple ‘big’ data option

- Uploaded files go directly to S3
- Uses pre-signed URLs – secure and access controlled – like the S3 download option
- Same user interface
- Supported in DVUploader (v1.0.7)
- **Tested to 39 GB so far**
- Configured per store

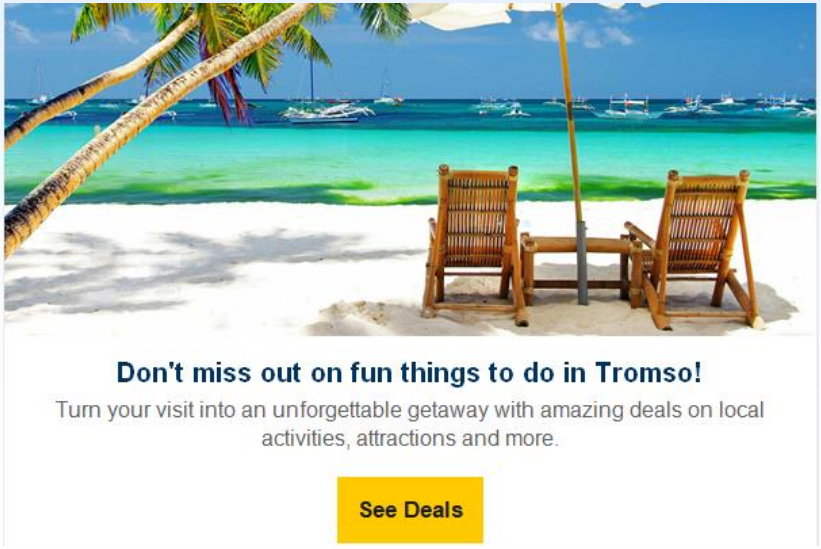


Current Limitations:

- Pull request awaiting review
- MD5 hashes only
- No Unzip
- No file inspection to determine mimetype
- Abandoned files are in S3

Questions

- Much of this functionality is ~v1.0
  - In use or planned for use
  - Usable, but likely to evolve
- Additional Information is in the Documentation
- Please ask, comment, complain, report bugs, request changes, contribute!



-- Thanks!

DVUploader – Hands On!

- Documentation: https://github.com/IQSS/dataverse-uploader/wiki/DVUploader,-a-Command-line-Bulk-Uploader-for-Dataverse
- Download: https://github.com/IQSS/dataverse-uploader/releases/download/v1.0.7/DVUploader-v1.0.7.jar
- Easiest: Put the jar in the parent folder of whatever folder(s)/file(s) you want to upload
- Login to Dataverse
  - Find/create a Dataset and get it's DOI
  - Create/copy your API Key
- Run

```
java -jar DVUploader-v1.0.7.jar -key=<api key> -did=<dataset doi> -server=<server URL> <dir/file(s)>

For example:
java -jar DVUploader-v1.0.7.jar   -key=8599b802-659e-49ef-823c-20abd8efc05c
  -did=doi:10.5072/FK2/TUNNVE   -server=https://dataverse.tdl.org   testdir
```

DVUploader – Hands On!

- Should see
- Also check timestamped logfile for details
- Try other options/flags
- Try <CTRL-C> and restart!

Dataverse Mode: Uploading files to a Dataverse instance Using apiKey: 8599b802-659e-49ef-823c-20abd8efc05c Adding content to: doi:10.5072/FK2/TUNNVE Using server: https://dataverse.tdl.org Request to upload: testdir

PROCESSING(C): testdir Found as: doi:10.5072/FK2/TUNNVE

PROCESSING(D): testdir\Capture3.JPG Does not yet exist on server. UPLOADED as: MD5:b2d8726f4ddba30705259143dbb283e3 CURRENT TOTAL: 1 files :9506 bytes

PROCESSING(D): testdir\Capture4.GIF Does not yet exist on server. UPLOADED as: MD5:3b9b536bd0abaf9c2677846f62d77ed9 CURRENT TOTAL: 2 files :23973 bytes

PROCESSING(D): testdir\Capture5.PNG Does not yet exist on server. UPLOADED as: MD5:ce26585c19bd1470b7229b2cfcc879f0 CURRENT TOTAL: 3 files :35448 bytes