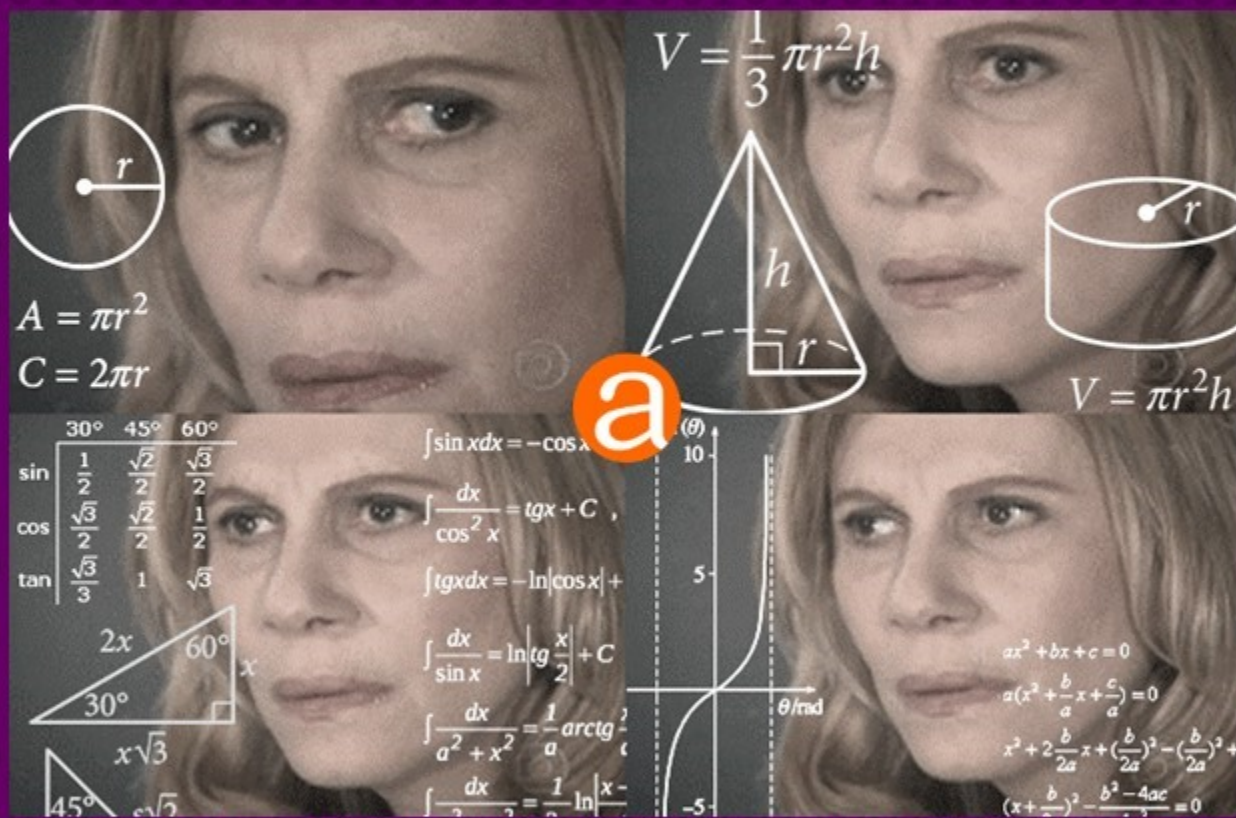


# Data Migrations



<http://boingboing.net/2016/11/08/heres-the-unexpected-origin.html>

Some Considerations when Preparing to Migrate to Atom

# Overview

---

- Assess what data will be part of the migration
- Do any in-system clean up prior to export
- Review AtoM data formats and available fields
- Establish a crosswalk between your data and AtoM's fields
- Export your data
- Perform any additional clean up as needed
- Transform your data to an AtoM compatible import format
- Import
- Review your work
- Revise and reimport if needed
- Make small clean up edits in AtoM directly



# Before Starting

## **Expect this to take time**

- Our average length for a client data migration project is around 4-6 months. Even for a simple project, there will be a lot of time needed for data clean-up, quality assurance review, and reimports.

## **Expect to do your import more than once**

- It's unlikely that everything will go perfectly on the first attempt. You'll discover some records don't quite match the same pattern as the rest, or one field didn't import, etc. Don't be discouraged, and do budget your time with this assumption in mind.

# Before Starting

## **Develop a data management plan while you migrate**

- How will you ensure you are not stranding data during the time of your migration? Will you freeze data entry entirely for the length? Manage your data in a spreadsheet? Run a small migration for new data at the end? Make sure everyone knows the plan.

## **Clarify roles, deadlines, and communication channels**

- Ensure everyone involved knows what is expected of them throughout the project, and when. Clearly identify those responsible for key roles, and where to go for support.



# Data assessment

## Questions to ask in the data assessment phase:

- How many descriptions do you have? How many top-level records?
- Have the records been described based on any content standards? (e.g. ISAD(G), RAD, DACS, MAD, MODS, etc.?)
- Are there custom fields with data in your system? How many? Do they readily map to known standards or not?
- What export formats does your system support?
- Is all record data captured in the exports?
- Are some descriptions “draft” or non-public? Is this information captured in the export?

# Data assessment

## Questions to ask in the data assessment phase:

- How many digital objects do you have to migrate? What types (images, text, video, etc) and formats (e.g. JPG, mp4, etc) are represented?
- Are authority records maintained separately from descriptions? What about other entities? Accession records?
- Is the relationship between these entities and descriptions captured in the export formats available?
- Do these other record types have their own export formats? (e.g. EAC-CPF XML, SKOS XML, CSV, etc)
- How are hierarchical relationships captured in the export?



# Atom Data Formats

---

## Archival descriptions

- CSV, EAD 2002 XML, MODS XML

## Authority records

- EAC-CPF XML, CSV

## Accessions

- CSV

## Terms (Subjects, Places, Genres, etc.)

- SKOS – many serializations supported

## Repository records

- CSV

Current as of  
version 2.4

# AtoM CSV Templates

[Home](#)[Downloads](#)[Documentation](#)[Community](#)[Wiki](#)[Demo](#)

## CSV templates

[< Resources](#)[Main Page](#) > [Resources](#) > Resources/CSV templates

On this page you'll find copies of the CSV templates we provide for use when importing data into AtoM. As the CSV files change over versions, we will add sections, so that you can be sure you are using the right template for your AtoM installation.

### Tip

Did you know you can also find all the CSV import templates inside AtoM? They are located in

`lib/task/import/example`.

For more information on CSV import, see the CSV section of our [User manual](#). [↗](#) Links directly to each major release's CSV import documentation are included in the relevant sections below.

### See also

- [CSV transformation](#)

### Page organization

This page has been divided first by type - archival description, or other CSV import type (as the archival description CSV templates are the most commonly used, and also the ones most subject to further development and versioning), and then in each section, by release. If the template has not changed for a version, it will **not** be added to the newest release section - you can simply download the link in the previous version.

### ACCOUNT

- [Dan](#)
- [Talk](#)
- [Preferences](#)
- [Watchlist](#)
- [Contributions](#)
- [Log out](#)

### PAGE

- [Discussion](#)
- [Edit](#)
- [History](#)
- [Delete](#)
- [Move](#)
- [Protect](#)
- [Unwatch](#)

### SEARCH



[https://wiki.accesstomemory.org/Resources/CSV\\_templates](https://wiki.accesstomemory.org/Resources/CSV_templates)



# AtoM CSV Templates

CSV import will be the best way to get data into AtoM - because the CSV import template is a format specific to AtoM, there is no data loss and all fields are represented.

If you are able to get your data out of your legacy system and transform it into an AtoM-compatible CSV format, we recommend using this method for your migration project.

[https://wiki.accesstomemory.org/Resources/CSV\\_templates](https://wiki.accesstomemory.org/Resources/CSV_templates)

# Atom CSV Templates

	I	J	K	L
1	repository	archivalHistory	acquisition	scopeAndContent
2	Example Repository	Example fonds Archival history (ISAD 3.2.3)	Immediate source of acquisition or transfer (ISAD 3.2.4)	Example fonds Scope and content
3	Example Repository	Example item Archival history	Example item Immediate source of acquisition or transfer	Example item Scope and content
4				

In the example CSV files from v2.2 on, we have included the relevant content standard name and number in the sample data field. This means you can import the CSV template to produce a sort of “crosswalk” or key, showing you how fields in Atom map to the column headers.

[https://wiki.accessmemory.org/Resources/CSV\\_templates](https://wiki.accessmemory.org/Resources/CSV_templates)



# Atom CSV Templates


Example Repository

Holdings Quick search

Fonds F1 - Example fonds

Item IT-1 - Example item (draft)

Fonds F1 - Example fonds



	I	J	K	L
1	repository	archivalHistory	acquisition	scopeAndContent
2	Example Repository	Example fonds Archival history (ISAD 3.2.3)	Immediate source of acquisition or transfer (ISAD 3.2.4)	Example fonds Scope and content
3	Example Repository	Example item Archival history	Example item Immediate source of acquisition or transfer	Example item Scope and content
4				

Import

[XML](#)  
[CSV](#)

Export

[Dublin Core 1.1 XML](#)  
[EAD 2002 XML](#)

Finding aid

[Generate](#)  
[Upload](#)

Related subjects

Subject access point 1  
Subject access point 2

Related people and organizations

Accumulator (ISAD 3.2.1)  
(Accumulator)  
Creator (ISAD 3.2.1)  
(Creator)  
Name access point 1  
(Subject)  
Name access point 2  
(Subject)

Related places

Genre access point 1  
Genre access point 2

Physical storage

Shelf: Example Shelf -  
Example Shelf Location

Identity area

Reference code F1  
Title Example fonds  
Date(s) • 2016 (Accumulation)  
• January 1, 2015 - December 31, 2016 (Creation)  
Level of description Fonds  
Extent and medium Example fonds Extent and medium (ISAD 3.1.5)

Context area

Name of creator **Creator (ISAD 3.2.1)**  
Biographical history:  
Example fonds Creator history (ISAD 3.2.2). Note that this will be added to the related authority record for Creator (ISAD 3.2.1).  
Repository **Example Repository**  
Archival history **Example fonds Archival history (ISAD 3.2.3)**  
Immediate source of acquisition or transfer **Immediate source of acquisition or transfer (ISAD 3.2.4)**

Content and structure area

Scope and content **Example fonds Scope and content (ISAD 3.3.1)**  
Appraisal, destruction and scheduling **Example fonds Appraisal, destruction and scheduling (ISAD 3.3.2)**  
Accruals **Example fonds Accruals (ISAD 3.3.3)**  
System of arrangement **Example fonds System of arrangement (ISAD 3.3.4)**

Conditions of access and use area

# AtoM EAD 2002 XML


Example Repository

Holdings Quick search

Fonds F1 - Example fonds

Item IT-1 - Example item (draft)

Fonds F1 - Example fonds



Identity area

Reference code

F1

Title

Example fonds

Date(s)

- 2016 (Accumulation)
- January 1, 2015 - December 31, 2016 (Creation)

Level of description

Fonds

Extent and medium

Example fonds Extent and medium (ISAD 3.1.5)

Context area

Name of creator

Creator (ISAD 3.2.1)

Biographical history:

Example fonds Creator history (ISAD 3.2.2). Note that this will be added to the related authority record for Creator (ISAD 3.2.1).

Repository

Example Repository

Archival history

Example fonds Archival history (ISAD 3.2.3)

Immediate source of acquisition or transfer

Immediate source of acquisition or transfer (ISAD 3.2.4)

Content and structure area

Scope and content

Example fonds Scope and content (ISAD 3.3.1)

Appraisal, destruction and scheduling

Example fonds Appraisal, destruction and scheduling (ISAD 3.3.2)

Accruals

Example fonds Accruals (ISAD 3.3.3)

System of arrangement

Example fonds System of arrangement (ISAD 3.3.4)

Conditions of access and use area

Clipboard

Add

Explore

Reports

Browse as list

Browse digital objects

Import

XML

CSV

Export

Dublin Core 1.1 XML

EAD 2002 XML

Finding aid

Generate

Upload

Related subjects

Subject access point 1

Subject access point 2

Related people and organizations

Accumulator (ISAD 3.2.1) (Accumulator)

Creator (ISAD 3.2.1) (Creator)

Name access point 1 (Subject)

Name access point 2 (Subject)

Related genres

Genre access point 1

Genre access point 2

Related places

Place access point 1

Place access point 2

Physical storage

Shelf: Example Shelf -

Example Shelf Location

Similarly, if you ensure all data entry fields in AtoM are filled in with related content standard names and numbers, you can now export as EAD XML to generate an EAD – ISAD(G) crosswalk from AtoM.



# AtoM EAD 2002 XML


Example Repository

Holdings Quick search

Fonds F1 - Example fonds

Item IT-1 - Example Item (draft)

Fonds F1 - Example fonds



Identity area

Reference code	F1
Title	Example fonds
Date(s)	<ul style="list-style-type: none"><li>2016 (Accumulation)</li><li>January 1, 2015 - December 31, 2016 (Creation)</li></ul>
Level of description	Fonds
Extent and medium	Example fonds Extent and medium (ISAD 3.1.5)

Context area

Name of creator	<a href="#">Creator (ISAD 3.2.1)</a> Biographical history: Example fonds Creator history (ISAD 3.2.2). Note that this will be added to the related authority record for Creator (ISAD 3.2.1).
Repository	<a href="#">Example Repository</a>
Archival history	Example fonds Archival history (ISAD 3.2.3)
Immediate source of acquisition or transfer	Immediate source of acquisition or transfer (ISAD 3.2.4)

Content and structure area

Scope and content	Example fonds Scope and content (ISAD 3.3.1)
Appraisal, destruction and scheduling	Example fonds Appraisal, destruction and scheduling (ISAD 3.3.2)
Accruals	Example fonds Accruals (ISAD 3.3.3)
System of arrangement	Example fonds System of arrangement (ISAD 3.3.4)

Conditions of access and use area

Clipboard

Add

Explore

Reports

Browse as list

Browse digital objects

Import

XML

CSV

Export

Dublin Core 1.1 XML

EAD 2002 XML

Findings aid

Generate

Upload

Related subjects

Subject access point 1

Subject access point 2

Related people and organizations

Accumulator (ISAD 3.2.1) (Accumulator)

Creator (ISAD 3.2.1) (Creator)

Name access point 1 (Subject)

Name access point 2 (Subject)

Related genres

Genre access point 1

Genre access point 2

Related places

Place access point 1

Place access point 2


Physical storage

Shelf: [Example Shelf](#) - [Example Shelf Location](#)

Similarly, if you ensure all data entry fields in AtoM are filled in with related content standard names and numbers, you can now export as EAD XML to generate an EAD – ISAD(G) crosswalk from AtoM.

# Atom EAD 2002 XML

## Fonds F1 - Example fonds



Clipboard

- Add

Explore

- Reports
- Browse as list
- Browse digital objects

Import

- XML
- CSV

Export

- Dublin Core 1.1 XML
- EAD 2002 XML

Finding aid

- Generate
- Upload

Related subjects

- Subject access point 1
- Subject access point 2

Related people and organizations

- Accumulator (ISAD 3.2.1)
- Accumulator
- Creator (ISAD 3.2.1)
- Creator
- Name access point 1 (Subject)
- Name access point 2 (Subject)

Related genres

- Genre access point 1
- Genre access point 2

Related places

- Place access point 1
- Place access point 2

Physical storage

- Shelf: Example Shelf
- Example Shelf Location

Identity area

Reference code F1

Title Example fonds

Date(s)

- 2016 (Accumulation)
- January 1, 2015 - December 31, 2016 (Creation)

Level of description Fonds

Extent and medium Example fonds Extent and medium (ISAD 3.1.5)

Context area

Name of creator Creator (ISAD 3.2.1)

Biographical history:  
Example fonds Creator history (ISAD 3.2.2). Note that this will be added to the related authority record for Creator (ISAD 3.2.1).

Repository Example Repository

Archival history Example fonds Archival history (ISAD 3.2.3)

Immediate source of acquisition or transfer Immediate source of acquisition or transfer (ISAD 3.2.4)

Content and structure area

Scope and content Example fonds Scope and content (ISAD 3.3.1)

Appraisal, destruction and scheduling Example fonds Appraisal, destruction and scheduling (ISAD 3.3.2)

Accruals Example fonds Accruals (ISAD 3.3.3)

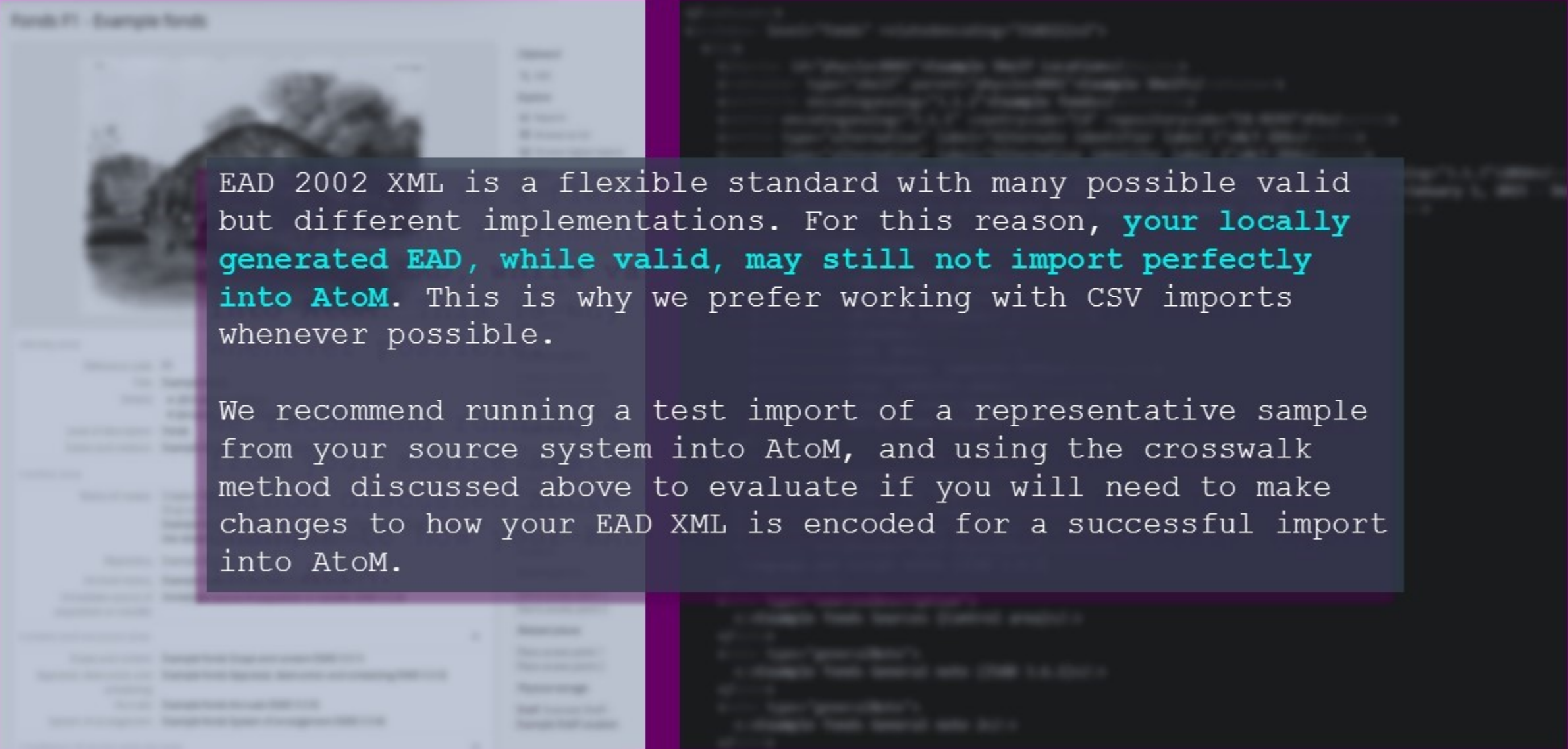
System of arrangement Example fonds System of arrangement (ISAD 3.3.4)

Conditions of access and use area

```
</eadheader>
<archdesc level="fonds" relatedencoding="ISAD(6)v2">
  <did>
    <physloc id="physloc0001">Example Shelf Location</physloc>
    <container type="shelf" parent="physloc0001">Example Shelf</container>
    <unittitle encodinganalog="3.1.2">Example fonds</unittitle>
    <unitid encodinganalog="3.1.1" countrycode="CA" repositorycode="EX-REPO">F1</unitid>
    <unitid type="alternative" label="Alternate identifier label 1">ALT-ID1</unitid>
    <unitid type="alternative" label="Alternative identifier label 2">ALT-ID2</unitid>
    <unitdate id="atom_4704_event" datechar="accumulation" normal="2016/2016" encodinganalog="3.1.3">2016</un
    <unitdate id="atom_4703_event" normal="2015-01-01/2016-12-31" encodinganalog="3.1.3">January 1, 2015 - De
    <physdesc encodinganalog="3.1.5">Example fonds Extent and medium (ISAD 3.1.5)</physdesc>
    <repository>
      <corpname>Example Repository</corpname>
      <address>
        <addressline>123 Example Street</addressline>
        <addressline>New Westminster</addressline>
        <addressline>British Columbia</addressline>
        <addressline>Canada</addressline>
        <addressline>V3L 3A7</addressline>
        <addressline>Telephone: (604)555-5555</addressline>
        <addressline>Fax: (604)555-6666</addressline>
        <addressline>Email: contact@example.com</addressline>
        <addressline>http://www.example.com</addressline>
      </address>
    </repository>
    <langmaterial encodinganalog="3.4.3">
      <language langcode="eng">English</language>
      <language langcode="fre">French</language>
      <language scriptcode="Latn">Latin</language>
      <language scriptcode="Cyr1">Cyrillic</language>
      Language and script notes (ISAD 3.4.3)
    </langmaterial>
    <note type="sourcesDescription">
      <p>Example fonds Sources (Control area)</p>
    </note>
    <note type="generalNote">
      <p>Example fonds General note (ISAD 3.6.1)</p>
    </note>
    <note type="generalNote">
      <p>Example fonds General note 2</p>
    </note>
  </did>
</archdesc>
```



# Atom EAD 2002 XML



EAD 2002 XML is a flexible standard with many possible valid but different implementations. For this reason, **your locally generated EAD, while valid, may still not import perfectly into Atom**. This is why we prefer working with CSV imports whenever possible.

We recommend running a test import of a representative sample from your source system into Atom, and using the crosswalk method discussed above to evaluate if you will need to make changes to how your EAD XML is encoded for a successful import into Atom.

# Crosswalking

**Crosswalking** is the process of mapping your source data fields to equivalent AtoM ones.

To do so, you must understand how AtoM handles some data (such as authority records, terms, etc.) first.

There will be cases where there are no 1:1 equivalencies either - you will have to make decisions about how to combine or split apart your existing data to make it work with what is available.



# Crosswalking

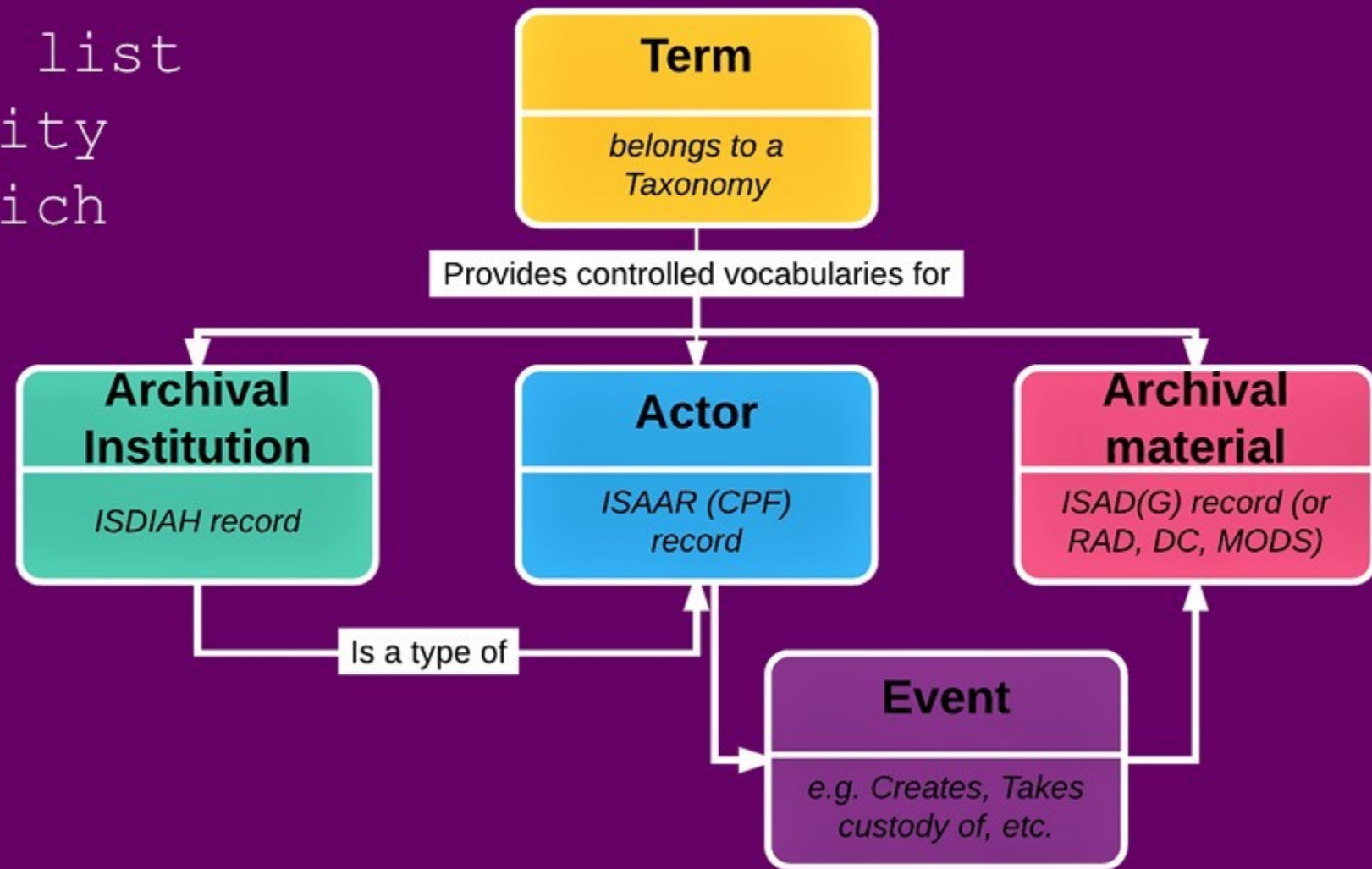
AtOM is standards-based.

This means you can focus on crosswalking to the content standard you know best. Use the guidance provided in the relevant standard to help inform your mapping.



# AtoM Entity Types

A(n incomplete) list of the main entity types around which Atom was built.





# Accessions

---

Accession records have their own CSV import format. As there is currently no international accessions standard, you will need to review the available fields in AtOM closely and determine where to map your data.

Descriptions can be linked to Accessions via the `accessionNumber` column in the description CSV templates. We recommend importing your Accessions first, then your descriptions with the corresponding accession number, to establish links.

See:

- <https://www.accesstomemory.org/docs/latest/user-manual/import-export/csv-import/#import-accessions-via-csv>

# Actors

---

In AtoM, creators and name access points are maintained separately as **authority records**, so they can be re-used and linked to multiple descriptions.

This means **any creator name or name access point you import with your descriptions will create an authority record**, or link to an existing match!

Make sure that names are consistent in your data, and the biographical/administrative history is about the actor only – not specific to the description.

See:

- <https://www.accesstomemory.org/docs/latest/user-manual/add-edit-content/authority-records/#authority-bioghist-access>
- <https://www.accesstomemory.org/docs/latest/user-manual/import-export/csv-import/#on-authority-records-archival-descriptions-and-csv-imports>



# Actors

---

The Actor data you can add to a description CSV is minimal - if you **do** maintain authority records, then you may want to import them separately via AtoM's authority record CSV templates.

There are **3 actor CSV templates** - the main actors template, 1 to supplement relationship data (between actors and/or resources) and 1 to supplement alternative forms of name.

We recommend importing authority records before descriptions, so you can link them on description import.

See:

- <https://www.accesstomemory.org/docs/latest/user-manual/import-export/csv-import/#creator-related-import-columns-actors-and-events>
- <https://www.accesstomemory.org/docs/latest/user-manual/import-export/csv-import/#import-authority-records-via-csv>

# Event Dates

- Display date
- Start date
- End date

Date(s) \*

Type	Date	Start	End	
Creation ▾	[ca. 1980] - 199-?	1980-01-01	1999-12-31	×
<a href="#">Add new</a>				

**Display date (what the public user sees)**  
Can use typography to represent approximation or uncertainty

**startDate**  
Internal ISO-8601 formatted date fields for searching  
Use: YYYY, YYYY-MM, or YYYY-MM-DD format

**endDate**

Description edit templates have 3 date fields. The Display date is what the end user will see - it is free text. The start and end dates must follow ISO 8601 (YYYY-MM-DD, etc) formatting. These fields are used to support AtOM's date range search.



# Event Dates

During CSV import, Creators and Dates are paired (as Events - see Entity types diagram).

Use the | pipe character to add multiple creators/dates.

You can use a literal NULL value in your CSV file to keep the spacing correct for dates without actors or vice versa:

F	G	H	I	J	K
eventActors	eventActorHistories	eventTypes	eventDates	eventStartDates	eventEndDates
Test Creator 1 Test Creator 2	History of Test Creator 1 History of Test Creator 2	Creation Accumulation	2001-2002 January 1, 1983 - December 31, 1984	2001 1983-01-01	2002 1984-12-31

See:

- <https://www.accesstomemory.org/docs/latest/user-manual/import-export/csv-import/#creator-related-import-columns-actors-and-events>

# Access Points

---

In AtoM, access points on a description (e.g. subjects, places, genre terms) are maintained separately as **terms** in a taxonomy so they can be controlled and reused.

This means that **access point data in your description imports will either create new terms or link to existing ones**. Make sure your data is consistent so you don't have near-duplicates later! (e.g. "cars" vs "car" vs "automobiles")

The exception is name access points - these are authority records!

See:

- <https://www.accesstomemory.org/docs/latest/user-manual/add-edit-content/terms/#term-name-vs-subject>



# Hierarchies

	A	B	C	D	E	
1	legacyId	parentId	qubitParentSlug	identifier	title	
2	1			F1	Example Fonds	→ Will create a new top-level record, "Example Fonds"
3	2	1		F1-S1	Example Series	→ Will create a new record, "Example Series" as a child of "Example Fonds"
4	3	2		F1-S1-IT1	Example Item	→ Will create a new record, "Example Item" as as child of "Example Series"

Hierarchies are managed in the description CSV templates via the `legacyId` and `parentId` columns. Parent records **must** import in a row above child records. The children should have the `legacyId` value of the parent record in the `parentId` column.

See:

- <https://www.accesstomemory.org/docs/latest/user-manual/import-export/csv-import/#hierarchical-relationships>

# Digital Objects

AU	AV	AW	AX
publicationStatus	digitalObjectPath	digitalObjectURI	physicalObjectName
Published	/path/to/my/local/exampleDigitalObject.pdf		Example Shelf
Published		http://www.example.com/example-external-image.jpg	

Can be imported with descriptions using the `digitalObjectURI` or `digitalObjectPath` columns.

**URIs** point to external, web-accessible resources – must end in file extension!

**Paths** point to a local directory added to your server prior to import.

See:

- <https://www.accesstomemory.org/docs/latest/user-manual/import-export/csv-import/#digital-object-related-import-columns>



# Data Transformation

You have 3 main options when it comes to transforming your data into an AtOM-compatible format:

- Manual data transformation
- Tools such as **OpenRefine**
- Transformation script

# Data Transformation



**OpenRefine** is “a free, open source power tool for working with messy data and improving it.”

There are many great free resources to help you get started.

See:

- <http://openrefine.org/>
- <https://github.com/OpenRefine/OpenRefine>



# Data Transformation



## Use OpenRefine to:

- Add AtOM column headers
- Normalize names and terms
- Standardize identifiers or accession numbers
- Split source data into separate columns
- Combine data into a single column
- Delete unnecessary rows
- Global search/replace
- etc

You can also use OpenRefine to clean up XML data!

# Data Transformation

A **transformation script** is generally a script prepared by a developer that takes an input (your source data) and runs a series of operations to transform the data into the desired output (an AtOM-compatible file).

These can be prepared in many programming languages (e.g. PHP, Python, etc).


```
9
10 import csv
11 import sys
12 import codecs
13
14 from transformations.mapper import *
15
16
17 def transform_csv(filename, fonds_id):
18     with open(filename, encoding='utf-8') as f:
19
20         # strip a UTF-8 BOM from the beginning of the file, if present
21         if not f.read(3) == codecs.BOM_UTF8:
22             f.seek(0)
23
24         csv_data = csv.DictReader(f)
25
26         # csv_data's file will close after this, so we can't use a generator.
27         print('> Transforming rows...')
28         return [
29             transform_row(sanitize_newlines(row), fonds_id) for row in csv_data
30             if row['Record Type-Name'] != 'Archival Box'
31         ]
32
33
34 def usage():
35     print('Usage: {} <source file> <destination file>'.format(sys.argv[0]))
36
37
38 def sanitize_newlines(row):
39     """
40     Perform two sanitizations on every column of data in the row:
41     * Replace newlines with spaces. Existing newlines in the data
42       is not in meaningful places.
43     * Replace the string '\n' with a newline.
44     """
45     for k, v in row.items():
46         if v is not None:
47             row[k] = v.replace('\r\n', ' ').replace('\n', '\r\n')
48     return row
49
50
51 def main():
52     if len(sys.argv) != 3:
53         usage()
54         return
55
56     # Just get fonds id from filename for now
57     fonds_id = sys.argv[1].split('/')[1].replace('_CSV.csv', '')
58     transformed_csv = transform_csv(sys.argv[1], fonds_id)
```



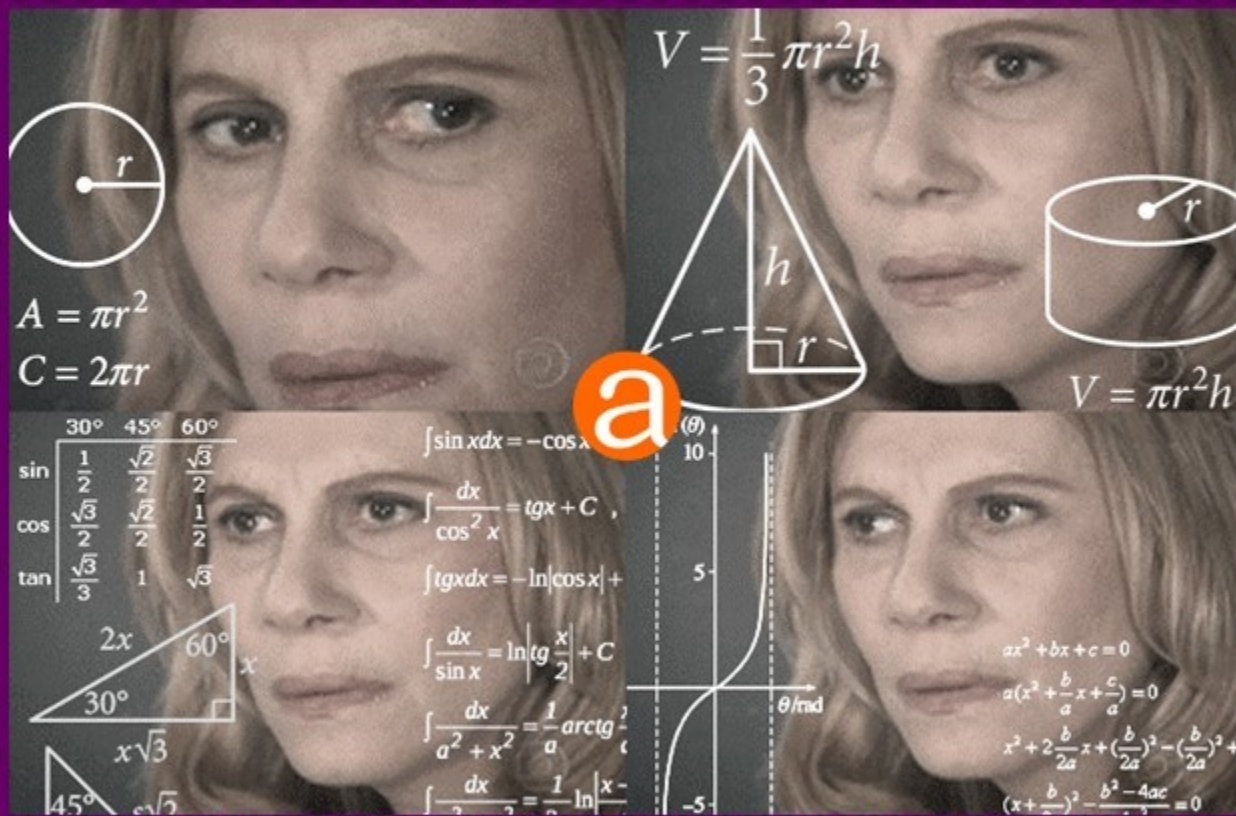
# Import Ordering

If you are working with several different types of data, you may need to perform multiple imports, possibly in different formats. If so, we recommend proceeding in this order to link entities together as your imports proceed.

We also recommend running a smaller sample test first!

- 
1. **Terms**
  2. **Repositories**
  3. **Actors**
  4. **Accessions**
  5. **Descriptions**

# Questions?



<http://boingboing.net/2016/11/08/heres-the-unexpected-origin.html>

info@artefactual.com