DCE

Developing Digital Repository Solutions

NOTCH8

# Community approaches to bulk import and export

**Julie Allinson** (Notch8)
**Mark Bussey** (DCE)

Samvera Connect 2019
WashU, St Louis, MO

# Everyone has import /export use cases

Migrate from / to another system

Export - metadata for cleanup & re-import

Ingest from a mass digitization project

# Batch Import-Export Working Group

- Identified the need
- … the use cases
- … and the requirements
- Reviewed existing solutions
- Proposed next steps

# Existing solutions

At that point (and still), various solutions exist(ed) (local and community):

- Hyku built-in importer
- Bridge2Hyku
- Darlingtonia
- Donut (Northwestern)
- Hyrax-Ingest (WGBH)

# Existing solutions

- None of them are entirely community-focussed
- Many overlap
- Lots of great features!

# Zizia

# Our “Problem”

Analyzing past DCE projects, we realized we’d built over 15 custom importers over the last 6 years.  Each one started based on unique local requirements, and each one ended up looking very different.

Solving the same problem over and over does not contribute to developer (or client) happiness...

# Summer 2018 - Universal Import

We wanted to take all the requirements from all the projects and make software that let us address them all.

The outcome was software that was complex, didn’t meet anyone’s exact requirements, and was time consuming to implement even in simple cases.

# The “Solution” - Design Sprint

**What’s a Design Sprint?**...a proven methodology for solving problems through designing, prototyping, and testing ideas with users. Design Sprints quickly align teams under a shared vision with clearly defined goals and deliverables. Ultimately, it is a tool for developing a hypothesis, prototyping an idea, and testing it rapidly with as little investment as possible in as real an environment as possible.

designsprintkit.withgoogle.com/introduction/overview

# Feb 2019 - Importer Design Sprint

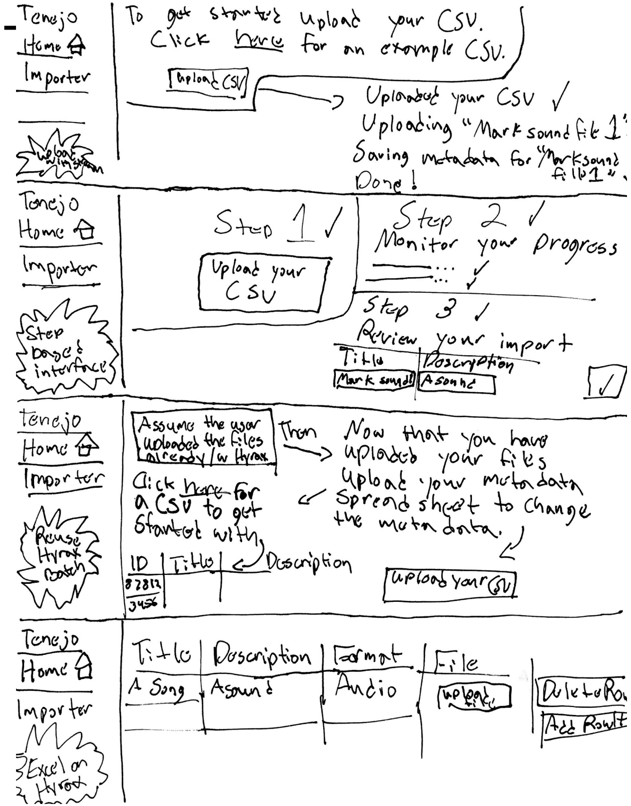
The whole DCE team takes 5 days for various design sprint exercises to:

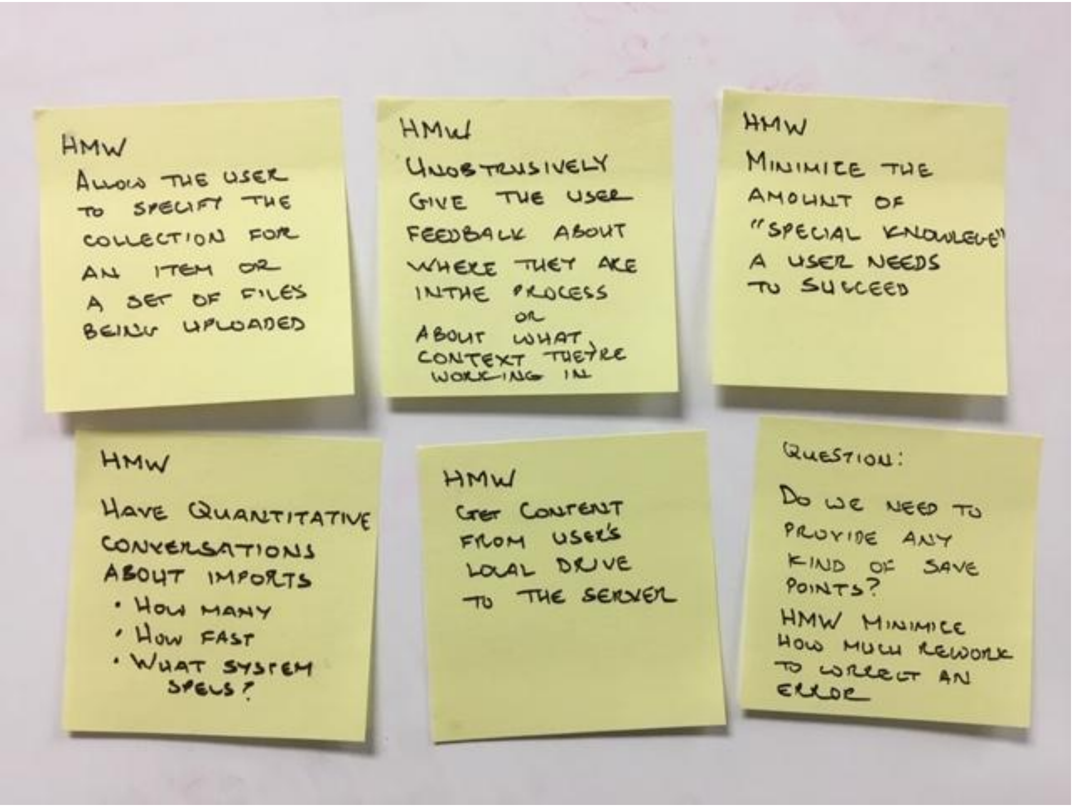
- Understand the problem space
- Define the problem scope
- Sketch multiple potential solutions
- Decide on a sketch(es) to pursue
- Prototype (1-Day!)
- Validate our prototype

# Sprint Challenge

Develop a self-service feature to upload large collections (>500 works) into Tenejo that allows non-developer users to easily upload their content and metadata in predictable timeframes.  Ideally, the system should support ingesting at least 1000 items per hour and allow them to be immediately discoverable.

**Deliverables:** 1) Design a Prototype that tests the validity of the design the team develops 2) Scope the outstanding work required to turn the prototype into released software





HMW
ALLOW THE USER TO SPECIFY THE COLLECTION FOR AN ITEM OR A SET OF FILES BEING UPLOADED

HMW
UNOBTRUSIVELY GIVE THE USER FEEDBACK ABOUT WHERE THEY ARE IN THE PROCESS OR ABOUT WHAT CONTEXT THEY'RE WORKING IN

HMW
MINIMIZE THE AMOUNT OF "SPECIAL KNOWLEGE" A USER NEEDS TO SUCCEED

HMW
HAVE QUANTITATIVE CONVERSATIONS ABOUT IMPORTS
- HOW MANY
- HOW FAST
- WHAT SYSTEM SPECS?

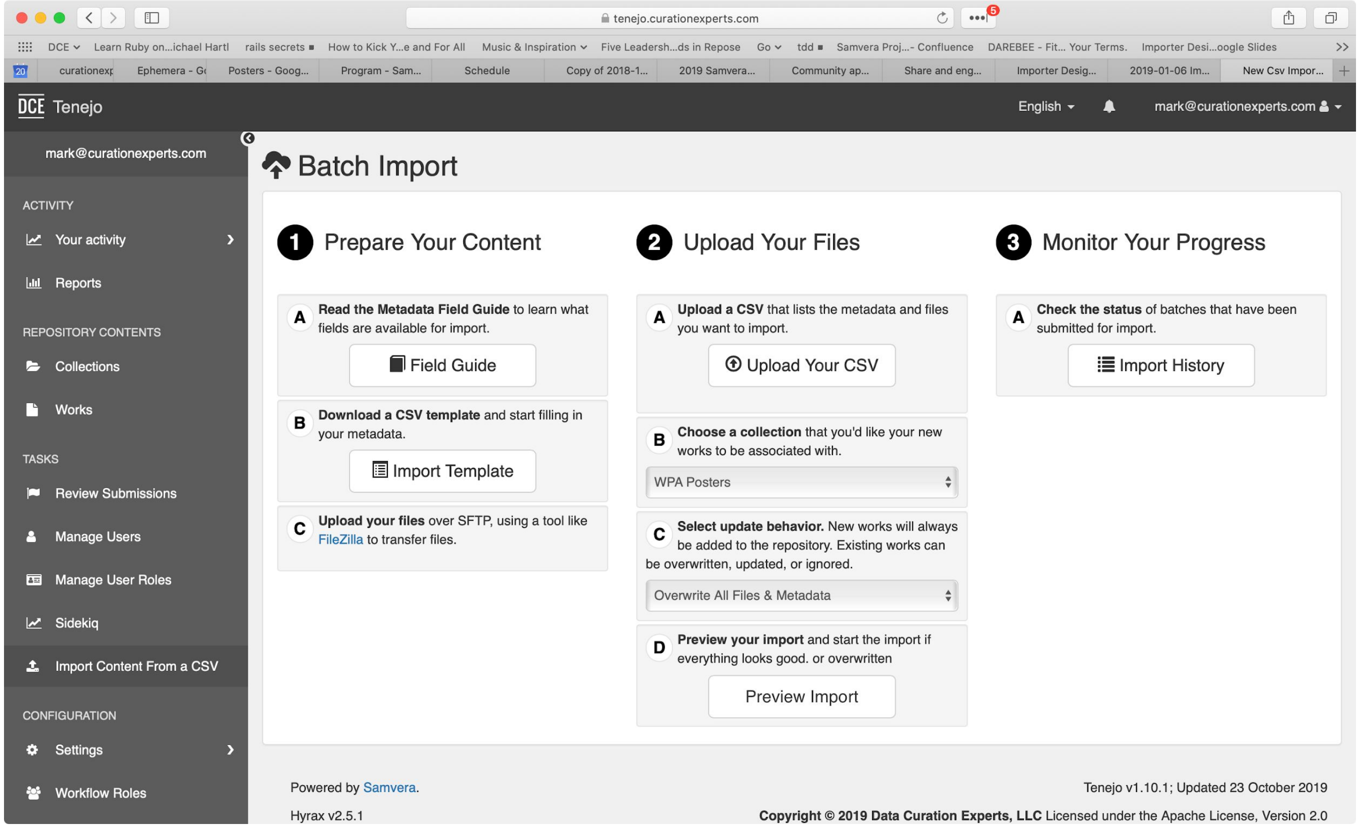
HMW
GET CONTENT FROM USER'S LOCAL DRIVE TO THE SERVER

QUESTION:
DO WE NEED TO PROVIDE ANY KIND OF SAVE POINTS?
HMW MINIMIZE HOW MUCH REWORK TO CORRECT AN ERROR

Artifacts from our design sprint - there were *many* of these over the 5 days

# Design Sprint Insights

- Provide sensible defaults for easy integration with Hyrax.
- File upload for large collections is usually handled by IT groups & not an application responsibility.
- Almost every collection can be represented as tabular data (CSV) & spreadsheet skills are ubiquitous.
- If we need a separate user instruction manual,
  It will *always* be out of date.
- Which came first, the collection or the work? (Collection)
- If we give librarians and archivists a good UI, developers don’t have to run the import!!! (eek a command line)



## Batch Import

### 1 Prepare Your Content

A **Read the Metadata Field Guide** to learn what fields are available for import.

Field Guide

B **Download a CSV template** and start filling in your metadata.

Import Template

C **Upload your files** over SFTP, using a tool like FileZilla to transfer files.

### 2 Upload Your Files

A **Upload a CSV** that lists the metadata and files you want to import.

Upload Your CSV

B **Choose a collection** that you'd like your new works to be associated with.

WPA Posters

C **Select update behavior.** New works will always be added to the repository. Existing works can be overwritten, updated, or ignored.

Overwrite All Files & Metadata

D **Preview your import** and start the import if everything looks good. or overwritten

Preview Import

### 3 Monitor Your Progress

A **Check the status** of batches that have been submitted for import.

Import History

Powered by Samvera.

Hyrax v2.5.1

Tenejo v1.10.1; Updated 23 October 2019

Code: github.com/curationexperts/zizia      Live Demo: tenejo.curationexperts.com

# Next Steps

- Build out tracking capabilities
- Structured works (parent-child relations)
- Ordered works (manuscript pages) DONE ✓
- MORE documentation (and we're proud of what we got)
- Even easier mapping
- XML & JSON input (slam dunk)
- Round-tripping (steal from Bulkrax???)

- Try it at home github.com/curationexperts/zizia

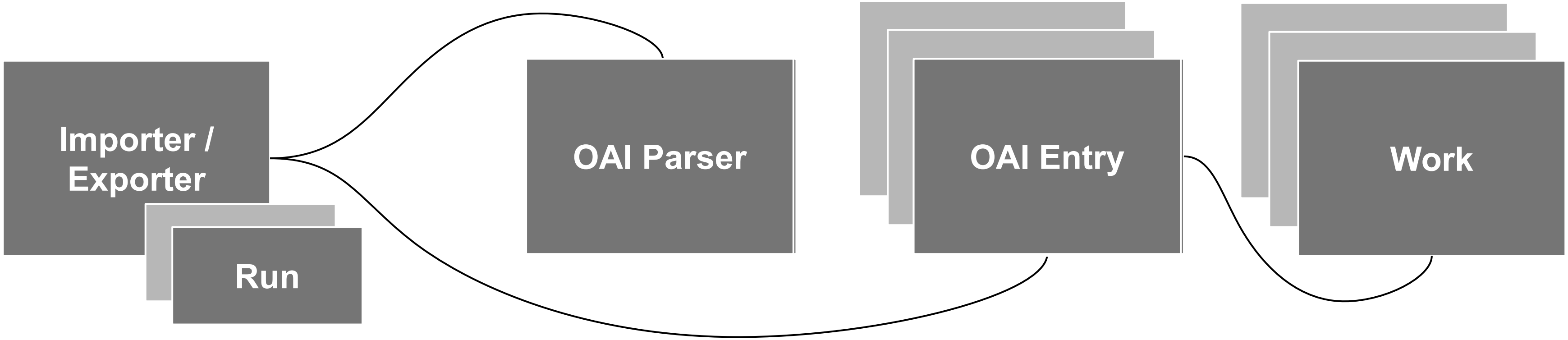
# Bulkrax

# Overview

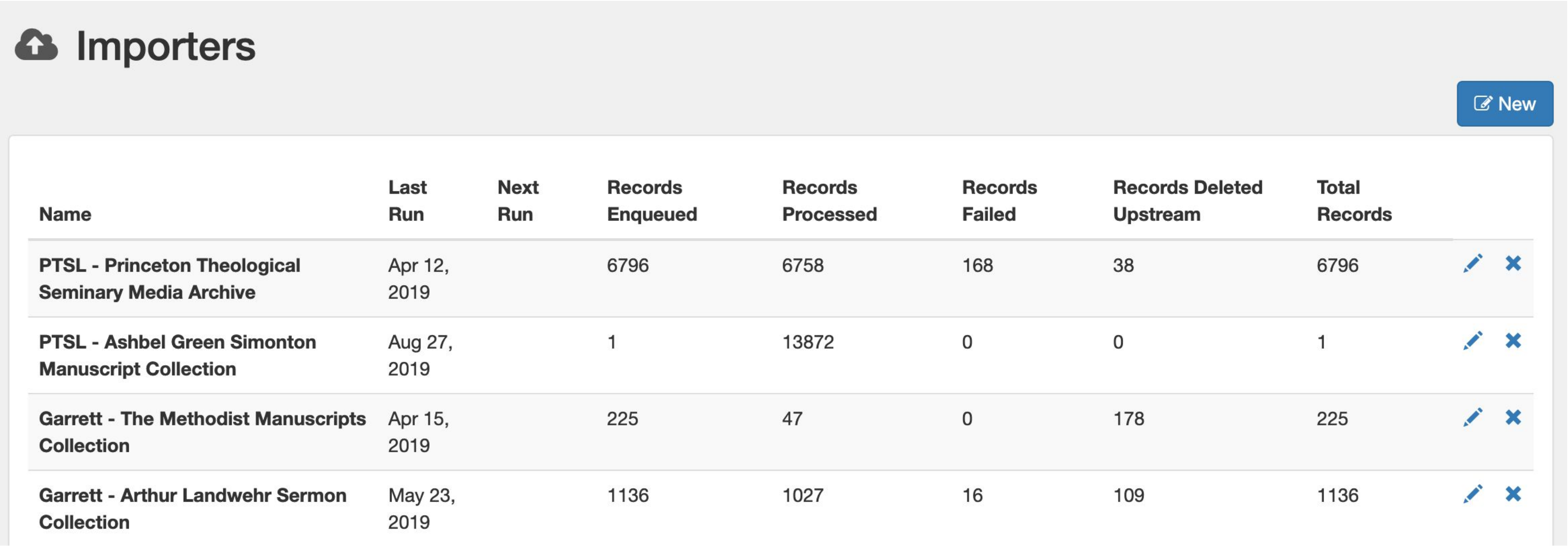
- Bulk import from CSV and OAI-PMH
- … plus a pattern for other formats
- Management dashboard available for Hyrax/Hyku
- Scheduling and repeatability

Coming soon

- Export
- More parsers; documented pattern for new ones
- Enhanced functionality and dashboard



# The Anatomy of a Bulkrax Import / Export



## Importers

| Name | Last Run | Next Run | Records Enqueued | Records Processed | Records Failed | Records Deleted Upstream | Total Records | |
|---|---|---|---|---|---|---|---|---|
| **PTSL - Princeton Theological Seminary Media Archive** | Apr 12, 2019 | | 6796 | 6758 | 168 | 38 | 6796 | ✎ ✖ |
| **PTSL - Ashbel Green Simonton Manuscript Collection** | Aug 27, 2019 | | 1 | 13872 | 0 | 0 | 1 | ✎ ✖ |
| **Garrett - The Methodist Manuscripts Collection** | Apr 15, 2019 | | 225 | 47 | 0 | 178 | 225 | ✎ ✖ |
| **Garrett - Arthur Landwehr Sermon Collection** | May 23, 2019 | | 1136 | 1027 | 16 | 109 | 1136 | ✎ ✖ |

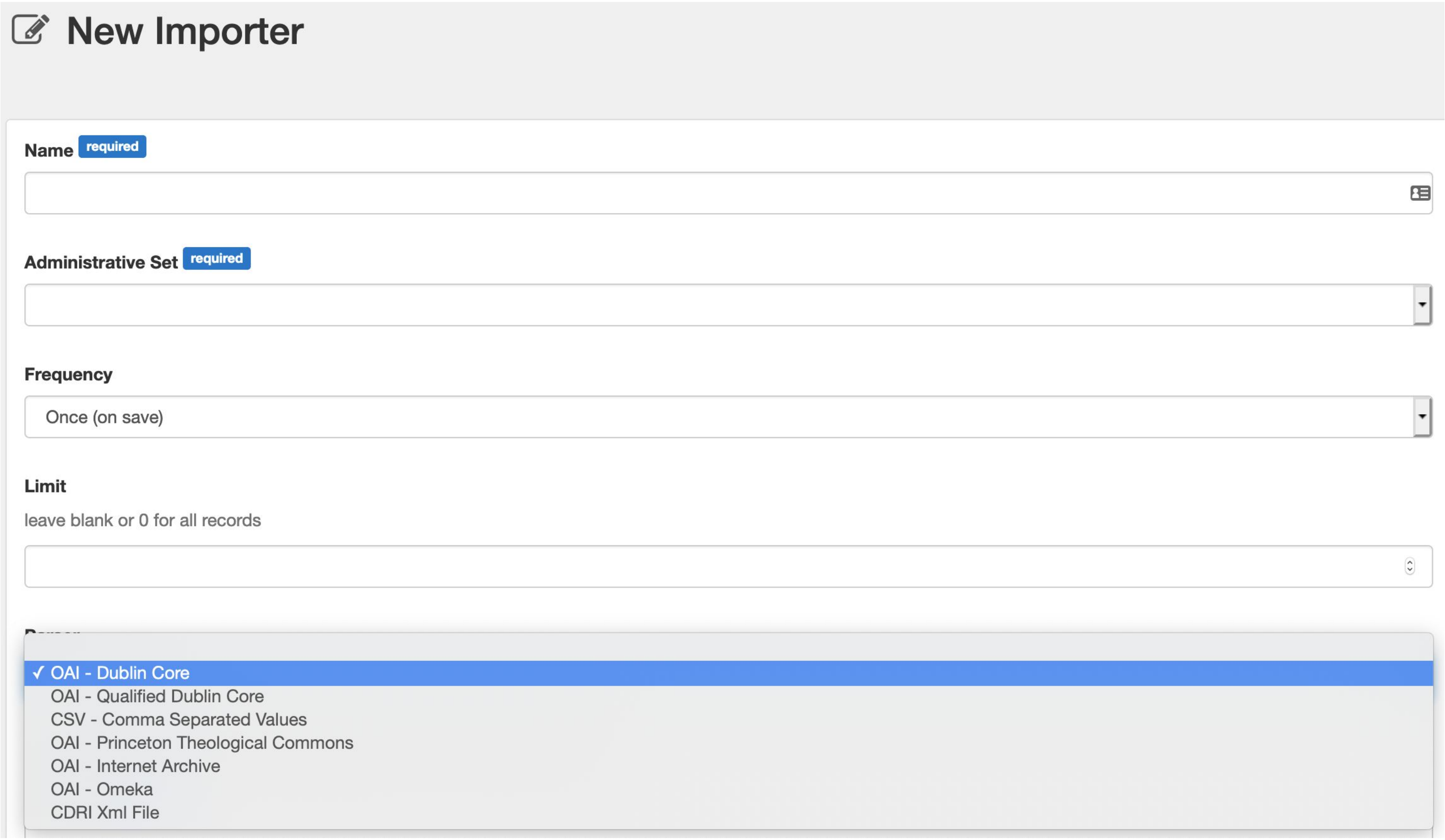
Importers dashboard in Hyrax

# OAI Import Example

Frequency to schedule updates

Limit the number of records to import

Standard and Custom Parsers



## New Importer

Name required

Administrative Set required

Frequency: Once (on save)

Limit
leave blank or 0 for all records

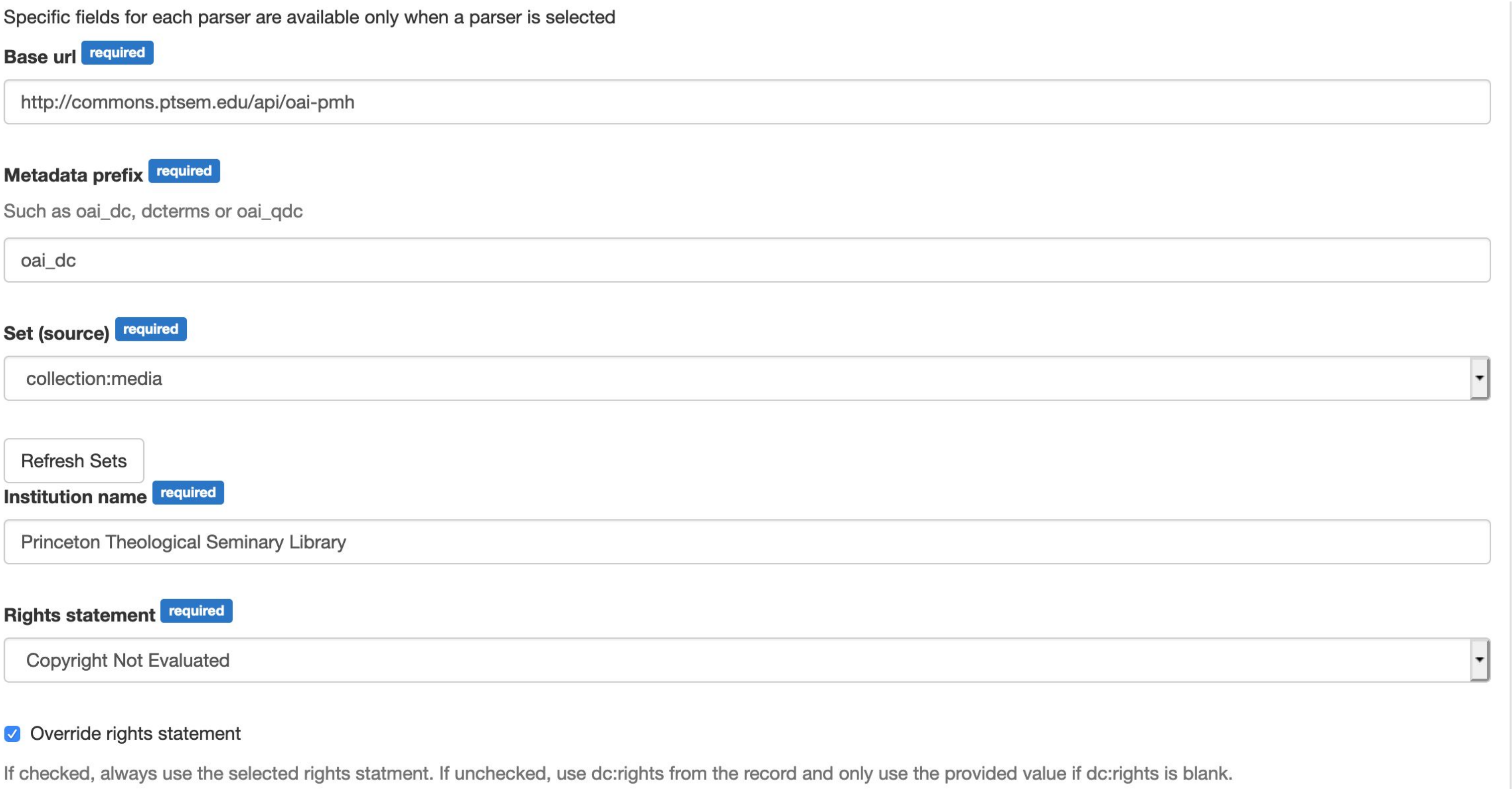
- ✓ OAI - Dublin Core
- OAI - Qualified Dublin Core
- CSV - Comma Separated Values
- OAI - Princeton Theological Commons
- OAI - Internet Archive
- OAI - Omeka
- CDRI Xml File

Base URL

Metadata Prefix

Set

Set a rights statement (with override option)



Specific fields for each parser are available only when a parser is selected

**Base url** required

http://commons.ptsem.edu/api/oai-pmh

**Metadata prefix** required

Such as oai_dc, dcterms or oai_qdc

oai_dc

**Set (source)** required

collection:media

**Institution name** required

Princeton Theological Seminary Library

**Rights statement** required

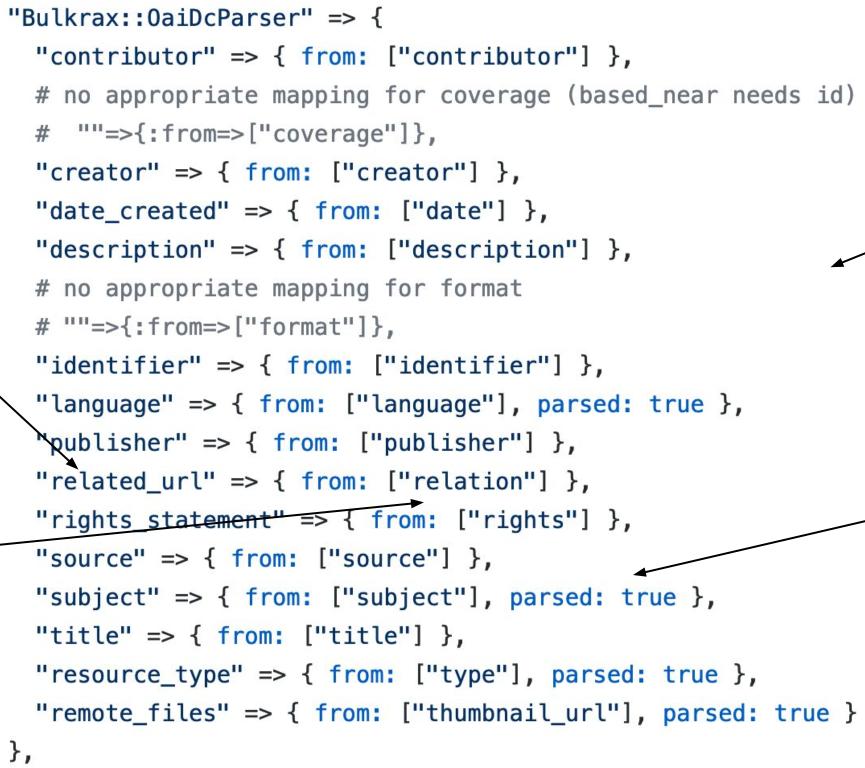
Copyright Not Evaluated

☑ Override rights statement

If checked, always use the selected rights statment. If unchecked, use dc:rights from the record and only use the provided value if dc:rights is blank.

# OAI Import Example

# Mappings from source data to (your) Hyrax



```
"Bulkrax::OaiDcParser" => {
  "contributor" => { from: ["contributor"] },
  # no appropriate mapping for coverage (based_near needs id)
  #  ""=>{:from=>["coverage"]},
  "creator" => { from: ["creator"] },
  "date_created" => { from: ["date"] },
  "description" => { from: ["description"] },
  # no appropriate mapping for format
  # ""=>{:from=>["format"]},
  "identifier" => { from: ["identifier"] },
  "language" => { from: ["language"], parsed: true },
  "publisher" => { from: ["publisher"] },
  "related_url" => { from: ["relation"] },
  "rights_statement" => { from: ["rights"] },
  "source" => { from: ["source"] },
  "subject" => { from: ["subject"], parsed: true },
  "title" => { from: ["title"] },
  "resource_type" => { from: ["type"], parsed: true },
  "remote_files" => { from: ["thumbnail_url"], parsed: true }
},
```

Hyrax property

DC property

Default …
can be overridden
locally in config

Fields can have custom cleanup method - could be simple, eg. trim off trailing periods, or more complex, eg. lookup the value in an authority list and reject if it isn't there

**Adding this configuration to the dashboard is on the roadmap**

# Take Home Messages

# A system is more trusted if it has robust import and export support

# No single solution will meet all needs (but we really don’t need THAT many solutions)

As a community we can coalesce on common patterns and share code