

# Zotero to Aquifer MODS

## Zotero Metadata-related Tasks

Zotero's MODS importer does not currently map all MODS fields correctly into the Zotero data model. An MWG subgroup met with Steve Toub (liaison for Zotero local implementation) to define the work of specifying the preferred mappings.

After the specification is complete, Aquifer needed to either submit the relevant patches to Zotero or ask the Zotero developers to patch. Originally, our thought was that Zotero would be able to round-trip MODS correctly (i.e., to import all MODS fields to Zotero and be able to export from Zotero back to MODS without losing any information), but ultimately, priority has been given to improving Zotero capture from MODS records in Aquifer.

Laura Akerman completed a preliminary analysis based on examination of the most recent MODS to Zotero mapping and has forwarded it to Connie Moon Sehat, who is leading a partnership of librarians and scholars at Emory University and George Mason University to extend Zotero functionality, to be added to the pipeline of improvements.

[Preliminary analysis of MODS capture by Zotero, with examples](#)

## Ensure Zotero imports MODS records correctly

1. Create or isolate MODS test cases (does MWG already have this?)
2. Figure out the best technical environment where we can import test records.
  - a. Steve has ruled out Unix/Mac/DOS file format differences but is still seeing discrepancies between importing from the icon in address bar and importing from a MODS file. He has started a [thread on the Zotero forums](#).
3. Document existing [Zotero data model](#)
4. Document existing [MODS to Zotero mapping](#) based on real-world imports not Steve's guesses based on the database schema.
5. Decide how far we want to go with Zotero "itemTypes" in: A) MODS to Zotero mapping and B) icon displayed in the Firefox address bar when using the Aquifer portal
  - Comments:
    - See [Zotero itemTypeFields](#) for details on which fields appear for which itemType
    - The generic itemType is "Document"
    - The import logic for fields (other than the itemType field itself) isn't changed when the itemType changes
    - The code that determines which icon displays in the Firefox address bar is defined in the detectWeb function in the translator; this is completely different than the doImport function that defines how the MODS fields get mapped to the Zotero data model
6. Specify preferred [MODS to Zotero mapping](#) for each itemType we are going to target; highlight changes from as-is behavior
7. Implement changes in test copy of Zotero code
8. Test with all MODS test cases until everything works as desired
9. Patch Zotero code and release in production

## Ensure Zotero exports records to MODS format correctly (optional)

1. Create or isolate Zotero test cases (e.g., imported data from different source formats)
2. Document existing [Zotero to MODS mapping](#)
3. Specify preferred Zotero to MODS mapping; highlight changes from as-is behavior
4. Implement changes in test copy of Zotero code
5. Test with all Zotero test cases until everything works as desired
6. Patch Zotero code and release in production