

# IU's Example Migration from Variations to Avalon 5.1.4+

The Indiana University Libraries migrated content from our [Variations Digital Music Library](#) instance to Avalon 5.1 in Summer 2016. This page provides additional detail that may be useful for institutions undertaking a similar migration. If you have additional questions, please post them to the [avalon-discuss-l mailing list](#) for followup.

## Migrating Variations Audio Files and Metadata

We used scripts to pregenerate derivatives from the .wav master files for our Variations audio content into medium and high bitrate AAC .mp4 files. We then copied the derivatives onto the Adobe Media Server (AMS) used by our Avalon server via shell scripts. Our Variations server uses a FileNameScheme with an idFormat of aaadddd ([more information](#)), so our Variations containers have IDs that look like vad1234, with media objects having IDs like vad1234a, vad1234b, vad1234c, etc. These IDs form the basis of our filenames in Variations.

The directory structure on our AMS then looked like this:

```
vad1234/vad1234a/vad1234a-audio-high.wav
vad1234/vad1234a/vad1234a-audio-medium.wav
vad1234/vad1234b/vad1234b-audio-high.wav
vad1234/vad1234b/vad1234b-audio-medium.wav
```

A list of derivatives was generated using this script that is really specific to IU's setup so you might need to write something from scratch that does something similar.



The next step was to create records for each item in Avalon, using information from the access page XML files in Variations. We copied all the access page XML files from public\_html/audio-access on our Variations server onto our Avalon server. We then used the following scripts to import the Variations items into Avalon:



mediainfo\_monkeypatch.rb



variations\_import\_script.rb



variations\_importer.rb



variations\_structure.xml

If you want to run the import on your own system, you'll need to modify these scripts to match your content.

We also had a little trouble with media object ordering of items imported into Avalon, such as having CD 2 show above CD 1. The sections of our Avalon items needed to be reordered into alphabetical order, so to fix that, we got a list of the items that need to be fixed and fed it into this script:



variations\_reverse\_order.rb

## Migrating Playlists

Avalon has a built in Variations Playlist File Importer. Here's more information: [Import Variations Playlist file into Avalon 5.1.4+](#)

## Migrating Bookmarks

The details of exporting bookmarks to JSON format are in this gist: <https://gist.github.com/mcwhitaker/03cab8a63bf80d431fa915b235522814>

Once you have the JSON, run this rake task to import it into Avalon:

```
RAILS_ENV=production bundle exec rake --trace avalon:variations:import filename=json.out
```

## Collection/Item Permissions

We created a Unit for the Music Library and under that unit, a collection called Variations. Before importing items, we set the default permissions on the collection to restrict access to a specific LDAP group (using Avalon's "External Group" functionality to connect with IU's Microsoft Active Directory instance) and populated that Active Directory group to contain those users who should have access. Another alternative would be to use LTI and your campus Learning Management System to grant access based on authorization in the LMS. Here are the [instructions for setting up LTI](#).

Your specific permissions setup will depend on your local policies for access. For help from other users and developers in figuring out how to best map your access policies to an access control strategy in Avalon, we suggest posting to the [avalon-discuss-l mailing list](#).