Batch Ingest Package Format

- Introduction
- Ingest Packages
  - Package Layout
  - Manifest File Format
    - Supported Field Names
    - Multiple File Ingest of Different Quality Files For a Single Avalon Item
  - Adding structure files via batch
  - Adding caption files via batch
  - Batch Processing Notes
  - MARC record ingest

This documentation is for Release 7.x. For the Release 1 version, see v.43. For the Release 2 version, see v.71. For Release 3.0.0, see v.86. For Release 3.1, see v.88. For Release 3.2, see v.108. For Release 3.3, see v.129. For Release 4.0, see v.159. For Release 5.x, see v.168 . For Release 6.x, see v.176.

Introduction

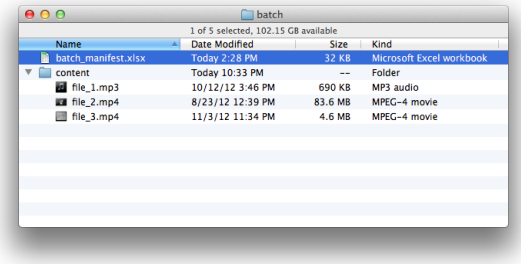
Avalon's *batch ingest* feature provides a method of building one or more media items at a time from uploaded content and metadata outside the user interface. A batch ingest is started by uploading an *ingest package* consisting of one *manifest file* and zero or more *content files* to the Avalon dropbox. For your convenience there is a demo ingest package available to download and import into test systems. Follow the instructions below to ensure a successful batch upload.

Ingest Packages

An *ingest package* is the combination of content and metadata that make up a single batch. Structural metadata documents in the form of XML may also be uploaded - one per a/v content file.

Package Layout

When a new collection is created, Avalon creates a subdirectory with the name of that collection (substituting underscores for any blanks), beneath the Avalon dropbox directory. The package (manifest file and associated content files) must be uploaded to that collection-named subdirectory or in a subdirectory beneath it. All items included in a single ingest package will be uploaded to the same collection. The following is a very simple package that has been uploaded:



Manifest File Format

The manifest file is a spreadsheet (`xls`, `xlsx`, `csv`, or `ods`) containing the metadata for the items to be created, as well as the names of the content files that make up each item. In this case, the manifest file is named `batch_manifest.xlsx`. See batch_manifest_template.xlsx for an Excel example file. Required fields are in bold. **Note: Neither the spreadsheet filename nor any folder/directory names above it can have blanks in them–substitute underscores.**

	A	B	C	D	E	F	G	H
1	Michael's First Test Batch	michael.klein@northwestern.edu						
2	Bibliographic ID	Title	Creator	Date Issued	File	Label	File	Label
3	123456	Test item 1	Klein, Michael B.	2012	content /file_1.mp3	Part 1	content /file_2.mp4	Part 2
4	789012	Test item 2	Northwestern	1951	content /file_3.mp4			

Row 1, Column A contains a reference name for the batch. This is mostly for your reference so we recommend naming the batch file according to what will help you remember the contents.

Row 1, Column B contains the submitter's email address (or username, depending on how your system is set up) to be used for notifications and exceptions. The submitter's email or user name must be listed as a manager, editor, or depositor for the collection in which this batch is deposited in the Avalon dropbox.

Row 2 specifies the names of the metadata fields supplied in the following rows. *Title*, *Date Issued,* and *File* are required. These fields are shown in bold in the Excel example file. Each subsequent row represents a single media item to be created. Metadata values are specified first, followed by a list of content files to be attached to each item. **Note: Make sure none of the field names in row 2 have leading or trailing blanks, or the field names will not be recognized by Avalon and will report an error.**

Content files listed in the manifest file must have the correct path noted for where those files are located in the Avalon dropbox, relative to the manifest file. Additionally, all content files must include a file extension. If necessary, include any directories or subdirectories (note the paths listed in columns E and G in the above example).

Multivalued fields are specified by multiple columns with the same header, e.g. *Topical Subject* in the following example:

	A	B	C	D	E	F
1	Michael's Second Test Batch	michael.klein@northwestern.edu				
2	Title	Creator	Date Issued	Topical Subject	Topical Subject	File
3	Nachos: A Memoir	Klein, Michael B.	2012-12-22	Meat	Cheese	content/tasty_tasty_nachos.mp4

Supported Field Names

Please see Supported Field Names for information about fields available and supported in the batch spreadsheet.

Multiple File Ingest of Different Quality Files For a Single Avalon Item

Avalon supports ingest of multiple derivatives that may be selected with the High/Medium/Low gear-buttons of the video player during playback (or High/Medium for audio). The “File” field in the manifest and the naming convention of the files in the Avalon dropbox directory must be formatted correctly for the batch ingest to be successful. Avalon will know what filename to look for from the manifest file, find the quality levels specified in the dropbox directory, and ingest the formatted files accordingly. It is not required to have all three quality tiers for multiple file ingest.

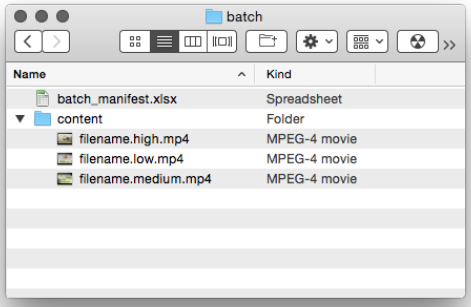
For a single Avalon item, input a filename in the “File” field and input “Yes” in the “Skip Transcoding” field of the manifest file. Add multiple files for this Avalon item to the dropbox directory. The “File” field as well as the file names of your different quality files in the Avalon dropbox directory must be formatted with the following convention:

File Name in Manifest File	filename.mp4
Files in Dropbox Directory	filename.high.mp4; filename.medium.mp4; filename.low.mp4

> Please note that files must match this convention strictly; extra periods are not allowed. **filename.test.high.mp4** is invalid; **filename.high.mp4** is valid.

Example manifest file for multiple file ingest of different quality files for a single avalon item:

	A	B	C	D	E
1	Michael's Third Test Batch	michael.klein@northwestern.edu			
2	Title	Creator	Date Issued	File	Skip Transcoding
3	Multiple Quality Ingest	Klein, Michael B.	2015	content/filename.mp4	Yes



Adding structure files via batch

The Batch Ingest Package can include XML structure files. One structure XML file can be attached per media file. See the demo ingest package at the top of this page for an example structural XML file included in a batch.

If the manifest lists a file named test.mp4, it will look for a structure file named test.mp4.structure.xml - you can edit the xml later via the user interface "Structure" tab in Avalon.

For more information about structure files (schema expectations and examples), see Adding Structure to Files Using the Graphical XML Editor.

Adding caption files via batch

The Batch Ingest Package can include WebVTT or WebSRT captions files. One captions file per media file. If the manifest lists a file named test.mp4, it will look for a captions file named test.mp4.vtt. If one is found, it will be attached to the media file as captions. This captions file can be updated or removed later via the user interface "Structure" tab in Avalon.

Batch Processing Notes

Each batch will generate 2 emails to the user listed at the top of the manifest.

Once Avalon detects the presence of an unprocessed manifest file, it will first verify that the necessary metadata columns are present in the manifest and that the file is not broken. At this stage, only the manifest file itself and **not** the metadata listed in the manifest has been validated.

If the manifest is incomplete or includes errors, such as invalid metadata values, only items that are valid (metadata which is valid, media file paths which are valid) will be created. An email will be sent to the email address specified in the manifest detailing the outcome, whether successful or not, listed in the manifest.

If a Bibliographic ID is provided for a resource but fails to process, the error email will only indicate that required fields are missing and will not indicate that the Bibliographic ID failed or was invalid.

To re-run a completed batch, follow the instructions in the email sent by the system after the batch is fully processed. It will contain a special filename that can be used to run the batch job again.

MARC record ingest

Please see information about MARC record ingest with Supported Field Names.