

# Generating Ingest Spreadsheets from Variations XML Files

When needing to create spreadsheets for batch ingesting Variations files.

## Variations XML Files

This guide will take you through the process of creating an ingest spreadsheet from Variations XML files. The XSL code converts the XML tag content into a txt file that can be imported into a spreadsheet (using "^" as the separator).

## Step-by-step guide

1. Open an SSH Client.
2. Log into magnolia.dlib.indiana.edu
  - a. Log in with your username and password.
3. Create a file named xmlingest\_sort.xsl using the code found below.
4. Run this command:

```
xsltproc xmlingest_sort.xsl ../srv/variations/content/access/audio/files.xml >> batchingest.txt
```

- a. You'll need to replace "files.xml" with the Variations xml files you wish to ingest.
- b. Replace "batchingest.txt" with the TXT file name you wish to create.

## Create a spreadsheet

1. Add column titles with the following:
  - a. Bibliographic ID
    - i. The Bibliographic ID will be the Variations XML file name (i.e. "VAA5239" from the file VAA5239.xml).
    - ii. Variations XML tag: //RecordSet/MediaObject/Label
  - b. Bibliographic ID Label
    - i. The XSL will fill this column with the word "other".
  - c. Title
    - i. This information will be overwritten by the information that the Bibliographic ID pulls in, but it is required.
    - ii. Variations XML tag: //DisplayTitle
  - d. Creator
    - i. This information will be overwritten by the information that the Bibliographic ID pulls in, but it is required.
    - ii. Variations XML tag: //Publisher
  - e. Date Issued
    - i. This information will be overwritten by the information that the Bibliographic ID pulls in, but it is required.
    - ii. Variations XML tag: //PublicationDate
  - f. Label
    - i. The XSL will fill this column with the word "other".
  - g. File
    - i. The file name(s) will be in the format of: vaa1234a.192k.mov
    - ii. You'll need a File column for each file associated with a Variations item.
    - iii. Variations XML tag: //RecordSet/MediaObject (The XSL edits the content of this tag so that it matches the file name and adds ".192k.mov")
  - h. Label
    - i. Leave this column blank.

## Importing the Data into the Spreadsheet

1. Import the data from the batchingest.txt file
  - a. The separator character is: ^
  - b. Using a comma as the separator character causes problems (since commas are used in some of the Variations titles).
2. Edit the dates in the Date Issued column.
  - a. The format of the dates vary and may need edited (i.e. "[1934]").

## xmlingest\_sort.xsl

```

<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:output method="text"/>
<xsl:template match="/">

<!--
Bibliographic ID
Bibliographic ID Label
Title
Creator
Date Issued
Label
File
-->
<xsl:variable name="publisherTest"><xsl:value-of select="//Publisher"/></xsl:variable>
<xsl:variable name="yearTest"><xsl:value-of select="//PublicationDate"/></xsl:variable>
<xsl:variable name="variationsName"><xsl:value-of select="//RecordSet/MediaObject/Label"/></xsl:variable>

<xsl:value-of select="substring($variationsName,1,7)"/>^other^<xsl:value-of select="//DisplayTitle"/>^<xsl:if
test="$publisherTest != ''"><xsl:value-of select="//Publisher"/></xsl:if><xsl:if test="$publisherTest =
''">unknown</xsl:if>^<xsl:if test="$yearTest != ''"><xsl:value-of select="//PublicationDate"/></xsl:if><xsl:if
test="$yearTest = ''">uuuu</xsl:if>^<xsl:for-each select="//RecordSet/MediaObject"><xsl:sort select="Label"
data-type="text" order="ascending"/><xsl:if test="string-length(Label) = 8">assets/<xsl:value-of select="
translate(Label, 'ABCDEFGHJKLMNOPQRSTUVWXYZ', 'abcdefghijklmnopqrstuvwxy')"/>.192k.mov^^</xsl:if></xsl:for-
each><xsl:for-each select="//RecordSet/MediaObject"><xsl:sort select="Label" data-type="text" order="ascending"
/><xsl:if test="string-length(Label) > 8">assets/<xsl:value-of select="translate(Label,
'ABCDEFGHJKLMNOPQRSTUVWXYZ', 'abcdefghijklmnopqrstuvwxy')"/>.192k.mov^^</xsl:if></xsl:for-each>^
</xsl:template>
</xsl:stylesheet>

```



## Related articles

- [Restart Delayed Job](#)
- [Upgrade Ruby using RVM](#)
- [Generating Ingest Spreadsheets from Variations XML Files](#)