EPICC Data Review Protocols

Last edit: 2018 September 26 by HL

Contents

[Additional Files 2](#_Toc509307651)

[Master Skeleton Records 2](#_Toc509307652)

[People 3](#_Toc509307653)

[Geologic Age 3](#_Toc509307654)

[Transcription Set Review 4](#_Toc509307655)

[0. Dropbox (Smithsonian)\EPICC Transcription\0. Pixel Acuity Sets 4](#_Toc509307656)

[1. Dropbox (Smithsonian)\EPICC Transcription\1. Initial Review 4](#_Toc509307657)

[2. Dropbox (Smithsonian)\EPICC Transcription\2. Scripts 4](#_Toc509307658)

[3. Dropbox (Smithsonian)\EPICC Transcription\3. OpenRefine 5](#_Toc509307659)

[Taxonomy (first pass) 5](#_Toc509307660)

[Collector Names 5](#_Toc509307661)

[Identifier 5](#_Toc509307662)

[Geography 5](#_Toc509307663)

[Geologic Age 5](#_Toc509307664)

[Locality and Collecting Event Information 5](#_Toc509307665)

[Other Numbers 6](#_Toc509307666)

[General column clean-up 7](#_Toc509307667)

[JSON script 7](#_Toc509307668)

[NOTES: 8](#_Toc509307669)

[4. Dropbox (Smithsonian)\EPICC Transcription\4. EMu Prep 8](#_Toc509307670)

[Routine EMu Records Maintenance 9](#_Toc509307671)

[Data Enhancement 9](#_Toc509307672)

[Taxonomy 9](#_Toc509307673)

[OpenRefine Second Pass 10](#_Toc509307674)

[Locality 10](#_Toc509307675)

[Stratigraphy 10](#_Toc509307676)

[Specimen Count Estimates 10](#_Toc509307677)

[Appendix A – Old Notes 11](#_Toc509307678)

# Additional Files

Additional files are used to support the transcription set review process and guide the data structure for each step. These should be regularly maintained throughout all procedures.

## Master Skeleton Records

EPICC Transcription\2. Scripts\Master\_Skeleton\_Records\_EPICC.csv

File provides a master list of all catalog numbers and associated information collected during capture in the production room. This information is essential for the first script run on all batches, which merges the batch records with their Locality Number, Locality Source, Drawer barcode, Catalog IRN, and Collections Event IRN.

Pulling skeleton records information from EMu:

1. Query EMu with the following criteria:
   1. Admin Tab -> Inserted Date = [Current Range]
   2. Collection Tab -> Collection Name = NSF ADBC EPICC TCN
2. Tools -> Report -> EPICC Skeleton Records -> Report All
   1. Location = Dropbox (Smithsonian)\EPICC Transcription
   2. Filename will be group1.csv (delete ecatalog.csv and schema.ini)
3. Open group1.csv, copy all records
4. Paste records into “Master\_Skeleton\_Records\_EPICC.csv”
5. Once all data is copied, delete group1.csv (or it will be saved over every time you run this report)

Pulling skeleton records information from image metadata:

You will use the Exiftool executable file to extract metadata from the images provided by Pixel Acuity for each transcription batch.

1. In Dropbox (Smithsonian)\NMNH\_Paleo\_TR\_Files open a command window
2. Input the following command with the folder name of the target batch folder and the output filename following the folder name
   1. Exiftool –csv –r –filename – iptc:OriginalTransmissionReference “[Folder Name]” > [output filename].csv
      1. Locality information is stored in the term “OriginalTransmissionReference”
3. The output csv will contain columns for “sourcefile”, “filename”, and “OriginalTransmissionReference”
4. Create a new column and input the formula =mid(B1,12,6)
5. Copy that formula down for each row by double clicking the lower right corner of the cell
   1. This formula will parse the catalog number from the filename
6. Copy the values from the new list of catalog numbers and the values from “OriginalTransmissionReference” into “Master\_Skeleton\_Records\_EPICC.csv”

## People

EPICC Transcription\4. Access\parties.csv

Continue to update known parties list for names that commonly appear in datasets (e.g. collector, identifier)

* Check names against EMu and other records to ensure that they were interpreted and transcribed correctly (example: a lot of transcription of “Brentt”, but these are actually “Orcutt” or transcription of “ESJ” is equal to “ELJ”).
* On first appearance of a name check for existing EMu record. If a record exists record the IRN in the parties.csv file.
  + Use the attachments search to verified suspected matches
* [Full] will be used as the match column for populating datasets with the parties record IRN for all names. This version of the name is the version of the name that should be standardized during the OpenRefine First Pass workflow.

## Geologic Age

EPICC Transcription\4. Access\geologic\_age2.csv

The Geologic Age CSV is used to backfill correct geologic age information formatted for EMu based on the Geologic Age column transcribed in each batch. After cleaning this column in the OpenRefine step described below, all new unique values should be added to the geologic\_age2.csv. Information is then added for each EMu column, which will be pulled into the batch when the matcher.py script is run.

* New unique values should be added to the match column
* If a similar string already exists in the match column, the information for the EMu formatted data can be mirrored.
* If no other related string exists use the EMu Geologic Age picklists to find the correct values for those columns.

# Transcription Set Review

Pixel Acuity will return weekly batches with a filename format of “NMNH-Paleo-20171117.csv”. The date will signify the Friday of the production week transcribed in that batch. Versions of the original dataset will be maintained for each stage as noted by the filename suffix, resulting in a total of 6 files per transcription set for the base review workflow:

* \_accepted or \_rejected
* \_merged
* \_openrefine
* \_emuprep
* \_catalog
* \_collevent
* (Tbd enhancement sets)

**\*\***At each stage the data should follow the structure defined in the corresponding tab for that stage in: EPICC Transcription\Data\_mapping\_template\_EPICC.xlsx**\*\***

Upon receipt complete the following steps:

## 0. Dropbox (Smithsonian)\EPICC Transcription\0. Pixel Acuity Sets

* Holly to acknowledge receipt and move new files into folder “0. Pixel Acuity Sets”

## 1. Dropbox (Smithsonian)\EPICC Transcription\1. Initial Review

Final file = [batchname]\_[initials]\_accepted

* Reviewer makes a copy of file into folder “1. Initial Review” and adds initials to end of filename
  + eg: NMNH-Paleo-20171117\_HL.csv
* Check 10% of the records
  + Divide total and then review a selection of that many records from across the set
* Add either “accepted” or “rejected” to end of filename
  + Eg: NMNH-Paleo-20171117\_HL\_accepted.csv

## 2. Dropbox (Smithsonian)\EPICC Transcription\2. Scripts

Final file = [batchname]\_[initials]\_merged

Once a batch is accepted run the parse.py script. This will merge the transcription data with the skeleton record information from EMu and parse the “Additional Information” column into multiple columns sorted by label:

Holly running variant workflow until all skeleton records ingested in EMu

* Run parse.py
  + Prompt, Initials = Initials entered in filename
  + Prompt, batch = date entered in filename
* Output will create new file with “merged” suffix

## 3. Dropbox (Smithsonian)\EPICC Transcription\3. OpenRefine

Fine file = [batchname]\_[initials]\_openrefine

OpenRefine is used to clean, standardize, and enhance data. You will do two passes with OpenRefine; the first will be to clean and standardize the data and the second will be to enhance the data. The second pass occurs after initial record population in EMu.

**All columns edited and added should follow the Data\_mapping\_template\_EPICC.xlsx outline**

### Taxonomy (first pass)

* Facet taxonomy column to cleanup any obvious discrepancies
* Move any information that is not a taxonomic name to *additional information*

### Collector Names

* Split Collector Names into multiple columns
* Rename Collector Names to Collector 1, 2, 3,…
* Cluster Collector Names and check for discrepancies (& symbols, etc.)
* Facet Collector Names in each column to clean and standardize (use the way the Collector name is written in the parties.csv file [Full] column)

### Identifier

* Repeat steps from Collector Names, but replace any use of the text “Collector Names” with “Identifier”

### Geography

* Split Geography into multiple columns
* Rename Geography columns into County and State (Note: city names should go in Location Description)
* Cluster County and State and check for discrepancies (Co. instead of County, misspellings, etc.)
* Facet Geography in each column to clean and standardize

### Geologic Age

After cleanup is finalized on this column pull unique values to be added to the geologic\_age2.csv match column as outlined above.

### Locality and Collecting Event Information

* Final entries should follow these guidelines:
  + For a single locality number with many records
    - Harmonize the information from across various labels into a single entry on one row leaving all other matching row with the same locality number and collecting events IRN blank (one complete set of locality and collecting events information per locality number)
  + For a single locality number with multiple collecting events
    - Follow the same guideline as above with the exception being a record for each instance of the locality number at a different date (e.g. USGS LOC 123 from 1917 is one record and USGS LOC 123 from 1923 is a separate record)
      * If the dates are within a couple of months of each other we will create one record with a date range. Enter both dates with a separator in the verbatim collection event date field.
  + If a locality number was entered incorrectly during production (e.g. the number populated from the matching script in LocSiteStationNumber):
    - Replace the number in LocSiteStationNumber with the correct number
    - Delete the IRN of the incorrect locality from both BioEventSiteRef.irn and CollEvent\_irn
      * In some cases we can simply update the information for that record, but this needs to be done very carefully to avoid unnecessary creation of duplicate records. Discuss with Holly if choosing this option.
    - If a record for the new locality number exists in EMu enter the IRN of the new locality in CollEvent\_change OR…
    - If a new record needs to be created for that locality enter “New Record” in CollEvent\_change
    - At the end of the project any Collection Events record with zero object attachments will be reviewed for deletion
* To support EMu import the final version of the dataset should include two columns with the collection events irn 1) bioeventsiteref.irn for the catalog import and 2) CollEvent\_irn for the collection events import. Entry into CollEvent\_irn should be treated the same as all other locality information with only one row populate per locality.
* Facet Locality Numbers and check for Locality Number in the ledgers
  + Compare Collector Names, Geography, Location Description, Geologic Age, Field Number, Collection Date, Notebook, and Page to ledger
    - If there is any discrepancy, check the information against the original image to determine validity. If the transcription is correct, leave information in the column as is. If the location number needs to be corrected, include a note onto the additional information string of “USGS \_\_\_\_ written on the label is incorrect and has been changed in the digital record to the correct number based on the specimen sticker and/or ledger”
    - Add information to Collector Names, Geography, Location Description, Geologic Age, Field Number, Date, Notebook, Page, and Identified as needed
  + In collection event note string include a note if locality information was found in USGS ledger
    - Locality in ledger: “Found in USGS Tertiary Paleontology Station Book \_\_”

### Other Numbers

Initial columns that could contain numbers other than the locality number include:

* Additional Information
* No. (parsed)
* Field (Parsed)
* Notebook (Parsed)
* Page (Parsed)

In some cases these columns can simply be cleaned and renamed to show that the information is finalized as outlined in the data\_mapping\_template\_EPICC.xlsc document. (e.g. Notebook, Page)

Any numbers that cannot be identified as a specific type of number should be moved to the end of the Additional Information string following this example:

* + Additional Information = (F.B. 2-76)
  + No. = 10876
  + Concatenated Additional Information = (F.B. 2-76); Other Numbers: 10876

In the following cases, numbers should be parsed into specific number columns using the column names as outlined in the Data\_mapping\_template\_EPICC.xlsx document

Field Numbers

A column titled “Field Number” should be added to the dataset for recording any number that is identifiable as a field number. Additional columns will be added with the JSON script at the end based on this column.

Original USNM Numbers

* Most likely USNM MO series. Can be checked against MO ledger images to verify and pull missing data.
  + S:\PaleoBiology\USNM\_MO\_ledgers\_Jpg
* USNM #s written on labels occasionally incorrect from what the number is for that specimen in the ledger. We are updating the digital record to have the correct previously assigned USNM # from the ledger.
  + When a number is corrected include a note onto the additional information string of “USNM \_\_\_\_\_ written on the label is incorrect and has been changed in the digital record to the correct number based on the ledger.”

### General column clean-up

* Use high resolution images to clean up any symbols (\*, !) in columns.
  + For Taxonomy, the EPICC Taxonomic Dictionary can also be helpful in identifying taxonomic names where interpretation of handwriting proves difficult

### JSON script

* Double check that column names align with those listed in the data mapping template csv
* Once all editing is completed the final step before exporting data out of OpenRefine is to run the JSON script stored in EPICC Transcription\3. OpenRefine\epicc\_columnedits.json
  + Under Undo/Redo select Apply
  + Copy and paste text from “epicc\_columnedits.json” into the provided text box
    - Number of collectors needs to be checked to ensure the correct scripts are included
  + Check columns to be sure that all necessary information was added

Export final CSV of the openrefine dataset to “3. Dropbox (Smithsonian)\EPICC Transcription\3. OpenRefine”

### NOTES:

* Complex records
  + Cut the record from the current file and create an issues file for that batch within the “3. OpenRefine/Issues” folder
* Things to keep in mind
  + You’ll start to see patterns and similarities between labels. These can help you standardize across records, or recognize a word or name more accurately in one label that can then be used to correct data from another. Don’t overreach though.
  + Be careful with dates. Make sure that you correctly specify between things like 1904 and 2004. Excel will try to change these and tends more towards the 2000s.

## 4. Dropbox (Smithsonian)\EPICC Transcription\4. EMu Prep

Final file = [batchname]\_[initials]\_emuprep

After all steps under 3. OpenRefine are completed another script is run to incorporate additional Geologic Age information and to match EMu IRNs for all parties identified in the batch. This python script will reference the “geologic\_age2.csv” and the “parties.csv” files.

Before Script:

* Check \_openrefine file for:
  + formatting of columns edited by json script
  + column names that are used in the script
  + geologic age information and that it is entered in the geologic age csv file
* Edit the matcher.py file to include the correct OpenRefine csv to be run
* Run matcher.py

After Script:

* Check new batch\_emuprep file for any errors
* Double check:
  + Check for any parties matches that require new parties records to be created
  + If there are new records needed: create the new parties record and update the parties.csv file
  + Check that all Geologic Age strings were matched
  + For any duplicate catalog numbers. There should only be one row per catalog number
  + All column names
* Split the batch\_emuprep file into the catalog import set and the collection events import set following the EPICC Transcription\Data\_mapping\_template\_EPICC.xlsx
  + Final file = [batchname]\_[initials]\_catalog in Catalog Module folder
  + Final file = [batchname]\_[initials]\_collevent in Collection Events folder

Once the two import files are ready we can begin the EMu import process within each module.

* Open EMu and load desired module
* Navigate to Tools -> Import
* Find desired file -> select custom import type
* First pass select validate format to check all columns names
* Second pass select Import Data -> next -> next -> next -> next
* Import Identifier = EPICC [batch name] [initials] -> next -> next
* Import will run and show any attachments to other modules or errors
* If errors are found a csv will be produced with the records that could not be imported correctly for your review

**DONE WITH FIRST PASS**

# Routine EMu Records Maintenance

**EMu data checks**

Collection Events Record:

1. Quick check on new records
   1. Search last ingest (based on date range)
      1. Check that all sources are correctly imported
      2. Check that all site numbers are correctly imported
      3. Merge duplicate locality records
         1. Best to use Attachment Check List view
2. Long term plan - In depth check:
   1. Compare newly ingested records with existing records to find duplicates
   2. Compare related records to enhance data

# Data Enhancement

## Taxonomy

After the initial review of a transcription batch and ingest of the batch into EMu further treatment is needed on the taxonomy information. In order to increase efficiency and quality for the EPICC project, new taxonomy records will be created in EMu unless a quick match can be found with the Scientific Names python script. The following workflow will be used to find and/or create taxonomy records based on transcribed names, parse all qualifiers and notations specific to individual catalog records, create a taxonomy match file, and integrate updated taxonomy based on the EPICC taxonomy dictionary.

Cheat sheet for interpreting taxonomic names: taxonomy\_cheat\_sheet.docx

Master taxonomy file: EPICC\_label\_taxonomy.csv

1. Following the same workflows as geologic age and parties, pull unique taxonomy strings from each batch and add these to the master taxonomy file under the “transcribed taxonomy” column. Ensure that Catalog Numbers associated with these strings are also included.
2. Parse the taxonomy string into specific fields for each type of information
3. Find unique genus and species combinations and run these against the EMu taxonomy file using the taxonomy\_irn.py script
4. Separate names that have a matched IRN and names that need records created
5. Create EMu import sheet for all new taxonomy
6. Update match sheet with with initial unique values and parsed qualifiers for cat module and new taxonomy irns
7. Repeat steps 3-5 for taxonomy from the EPICC taxonomic dictionary
8. Add a mirrored set of columns to the match sheet for the EPICC taxonomy information to create multiple taxon entries for specimens with updated names

HOLLY, ANNA, and ALEX to test and develop steps for assisting with this in OpenRefine

* Document csv and import structures
* What information should be pulled from EPICC Master Dictionary?
  + Synonyms? Authors? Etc…
* OpenRefine scripts for parsing columns based on number of words? Certain characters?
* Finalize interpretations of notations and qualifiers
* Issue list to run by Tom Waller
* Notes on current work:
  + Export a sheet from each batch with catnumber and taxonomy
    - Name file with new suffix of \_taxonomy.csv
  + Finding updated taxonomy
    - First try to match to EPICC taxonomic dicationary
    - If not match found look in WORMS
      * If match found in worms look for match on that name in EPICC taxonomic dictionary
      * Document path of matching and note if a worms name still can’t be found in EPICC taxonomic dictionary

## OpenRefine Second Pass

In addition to the taxonomy workflow outlined above, further data editing on locality and stratigraphic information is necessary to create a full specimen record in EMu.

### Stratigraphy

* Review of Locality String, ledgers, and USGS publications to
  + Split into Group, Formation, Member, Stratigraphy\_Note
  + backfill any information

## Specimen Count Estimates

1. Counting and data entry in EMu
   1. Query EMu with the following criteria:
      1. Admin tab -> Import Identifier = MDPP
      2. Admin tab -> Date Inserted = Date of last images added to EMu that haven’t been counted
      3. Multimedia tab -> Title = image 01
   2. Possibly count from the images in EMu and enter number directly into Catalog Record
      1. If all specimens are visible enter count in []
      2. If count it an estimate because of large number or obscured specimens enter count in []
2. If exporting images from EMu
   1. Dropbox (Smithsonian)\EPICC Transcription\5. Specimen Counts
   2. Tools -> Report -> Image Report -> Report All
      1. Location = Dropbox (Smithsonian)\EPICC Transcription\5. Specimen Counts
      2. and Data Cleanup Protocols
      3. Filename will be emultime.csv
   3. Run movefiles.py
      1. All tray images will be moved to a new images folder
   4. In images folder:
      1. Copy template excel sheet, replace “template” with the date of the image set (Date Inserted)
      2. Highlight all image files, shift+right-click -> copy as path
      3. Paste list of paths into column A [Path] of excel sheet
      4. Count specimens in each image and insert count beside the matching file path
      5. Once a full set of images has been counted, images can be deleted
      6. Add “COMPLETE” to the end of the excel sheet filename

# Appendix A – Old Notes

Notes: Holly to add in more information about notebook and page, additional fields for notes, instructions for memoranda, more to the Geologic Age query, more on Field Number

Other

Dates:

To consider - <https://github.com/kurator-org/kurator-validation/wiki/CSV-File-Date-Validator>

Validating data with GBIF, Kurator, and iDigBio tools