Notes:
2024-04-24: The first part of these instructions involve using the Bioimages Collection Manager software. As it is unlikely that new images will be added, one can simply make edits to the source CSV files and publish directly from them. For example, to change tree tour links, that can be done directly in the appropriate CSV and then run the XQueries to regenerate the web pages. Go to step 18 for that.

~~As of 2015-12-17, my work computer now has all of the capabilities to do the entire workflow on that one computer. I can just remote in from home to do everything if I'm not physically there. So the moving the files to my home computer using the external drive is now only a backup.~~

It is important that every image have some kind of a view set. If not, the software leaves the view cell empty and the XQuery breaks. So run a check for images with missing views before actually generating the CSVs. Also, if the images are not eventually going to be sent to Morphbank (see below), their "URL to High Res" value should be set to wherever they will ultimately be kept (e.g. "http://bioimages.vanderbilt.edu/highres/[filename]").

Notes from importing files from the SSMV project (first go on 2017-01-11):

A. I first inspected the CSV files they sent me to confirm that the rows and their content make sense. I had to delete one row in the image file because of an "orphan" photo that they didn't send to me. Also, check the georeferenceRemarks, continent, and country code - something is wrong there. The names.csv file didn't have a lot of the fields populated for the ITIS number they used. It was also missing the view. Manually removed underscore from URI. Negative sign missing from longitude. Missing time zone offset. Also, organism.csv had a generic name for the species in the organismName field. Changed to CC-BY since no photographer name (for permission). Assumed location determined by independent GPS measurement. Used 10 m for uncertainty. Also supressed from EOL since not IDed below family.

B. I went to the Manage CSV files screen, selected the folder where those CSVs were, and clicked on the "Import agent, determination, … CSVs" button. After a short delay, it told me that my database was up to date.

C. I went back to the Start Screen and chose view/edit existing records. I unchecked the "Load everyone's images. Under "Whose images would you like to load?" it had only steamteam and grigorjj (the new agents) in the agent dropdown. I'm thinking that this was a bug that we had encountered before where importing the new agent.csv and some other files erased existing ones.

Notes from importing files from SSMV project (second try on 2017-02-23)
A. cursory inspection looked like the files were OK.
B. Went to Manage CSV files menu, clicked on Import agent,… button and selected csvs. It told me to wait and it took a minute or two. In Importing complete popup appeared and I clicked OK.
C. Went to Edit Existing Records dialog. Unchecked Load everyone's images, Selected folder is the root…, and Load all records including previous. It was still showing that the earlier

streamteam images (first three) should be loaded and said that 2 out of 5 referenced images were found. I clicked OK anyway. It had the thumbnails (lowres) of the earlier images, and the 2 new ones.

D. On work computer, I ran the software and let it update database. I did the import, but it seemed to want the actual CSV's to be selected rather than the folder. But I think that might have just been a labeling issue because once I selected the CSVs it seemed to work fine.

14.A. Suppress flag: there is a "suppress" flag in the database tables for image, determination, and organism that does not show up in the GUI. This flag allows for suppressing certain default behaviors in particular situations. To set this flag, go to the database view and enter it in the "suppress" column, which is generally the last column in the table. Be sure to click save after any changes and before sorting, since sorting causes any unsaved changes to be lost.

The "suppress" flag is a number whose binary representation has 1's in places that correspond to features that should be suppressed. For example, if the features represented by the 4's place and 1's place are turned on and the 2's place is turned off, the binary number would be 101, or the sum of 4+1=5 in decimal.

Currently, here are the features that are in use:

Images:
1 = do not publish this image to Encyclopedia of Life [it does not appear that this has actually been implemented; existing suppressed cases are ones that have quality 3 or less]
2 = the image date does not represent the date of the occurrence; substitute the value of occurrenceRemarks before the first period as the occurrence date. For example, the photo is of a specimen taken years after the specimen was collected. The dateTime reported by the camera is the date the image was taken, but a date such as "1999-08-25.Pheromone trap #224." would be given as the value of the occurrenceRemarks and "1999-08-25" used as the actual date of the occurrence in the organism's web page and in the RDF. Valid date ranges like "1999-08-25/1999-09-01" are OK - the first part before the slash will be used to construct the occurrence and event hashes.

Determinations:
1 = do not display this determination as part of the index, but include it on the web page (along with an identificationRemark that it was in error) and in the RDF as a historical record. This may be used where determinations are discovered to be in error rather than just a different taxonomic opinion, or a determination to a higher taxonomic level. [I do not see that this has actually been implemented.] It also suppresses publication of information about that determination (and its associated images) to EOL. (when the GBIF export routine is created, I'm not sure how this should be handled. Don't know their policy about removing former identifications.)

14.B. Importing/merging images from another provider.
HTML blobs:
- must have single-quotes for attribute values.

  must be replaced with
’ replace with '
‘ replace with '
– replace with -
“ replace with "
” replace with "
½ replace with ½
é replace with é
° replace with °
¼ replace with ¼

15. The last step of the provider protocol instructions says to generate CSVs for locally changed records. If I'm doing editing for publishing the site, I need to save the CSVs for all records using the Manage CSVs option at the start screen of the software (i.e. uncheck the "Only export records changed locally" checkbox. **NOTE: before saving, make sure that the Bioimages folder in the local GitHub repo has been synched to avoid any editing conflicts.** The CSVs go into the c:\github\bioimages folder and replace the existing ones. Then they need to be uploaded to GitHub so that the queries can operate on them.

16. Although the function for generating extra links for the organism is not commonly used, the capability is there. The file links.csv on the GitHub site contains the information about the links and is edited manually. The field headers are pretty much self-explanatory and indicate the kind of triple linking to an external resource that will be created. In the HTML, a link using the objectDecription as the text will be created with the object IRI as the href value. Therefore the IRI must be an HTTP IRI in order for the link to work. [I don't think this actually is implemented.]

Note: Check that the images are actually rotated and not just that they have the rotate bit set. Otherwise, they will be all screwed up when they are displayed online. See the notes about this in the providers' guide. Currently I'm using Gimp as a hack to fix this, but there needs to be a more automated way to do it using a script.

17. The Generate Website function of the import tool software will generate all of the smaller versions of the images from the originals. It has to be pointed at the root of the local copy of the website and then puts the images in the correct folders for thumbnail, lower quality, and good quality:

tn folder t[filename].jpg=67x100 (100 in longest dimension)
lq folder w[filename].jpg=320x480 (480 in longest dimension)
gq folder g[filename].jpg=683x1024 (1024 in longest dimension)

If the necessary folders don't exist, the software will create them. The highres (original) files do not have to be at a particular location - the software will ask where they are. Currently, I have been moving them through a staged series of folders within the c:\image-processing\ folder. Most images should eventually end up at Morphbank, although some images with poorer quality or images taken by people without established Morphbank identities may just be put on the

server in the highres directory. Put one copy in a folder under "c:\image-processing\images-3-processed-need-morphbank-upload\" so that I remember that they need to go to Morphbank and another copy in the appropriate subfolder of "bioimages-hires" folder, which the software is using to check for thumbnails.

If the URL of the highres SAP isn't specified, it won't be included in images.csv, the RDF will not include information for a "best quality" SAP, and the Javascript doesn't link to a highres image. This will be the state for most images right after they are processed. Once they have made their way to Morphbank (see the Morphbank section below), they will have to be added.

18. Note on BaseX memory issues. For whatever reason, everytime there is an upgrade to BaseX, the memory allocation gets set back to 512m. This is not enough to run the bigger Xquery queries. To fix this, go to c:\Program Files (x86)\BaseX\bin and open basex.bat . In the line:
set BASEX_JVM=-Xmx512m %BASEX_JVM%
change 512m to something bigger. 2048m seems to be fine.

There are two XQueries that are run to determine which organism and image files need to be rewritten:

modified-images-file.xq
modified-organisms-file.xq

 in the xquery subfolder of the Bioiamges GitHub repo.

They are run from the command prompt as:

basex modified-images-file.xq

for example.

Note on 2024-04-24: to change and publish selected image or organism pages, just manually put the IRI identifiers in these two txt files without running the script that looks for modifications.

These queries generate text files: images-to-write.txt and organism-to-write.txt that are used to control which files are re-written. These files default to the "c:\test" directory. The way the query works, IRIs get duplicated, but that's OK because the following queries ignore duplicate listings. Note: before publishing, check the image file URIs to make sure that none of them have local names beginning with underscore, e.g. "http://bioimages.vanderbilt.edu/phoebusp/_9123", which could cause undesirable effects when the html and rdf files are generated. One can force a re-write by manually adding IRIs to this list and it is also probably the best way to force a re-write of the entire site (copy the IRIs from organisms.csv or images.csv and paste them into those files). Note: these two queries, as well as tsn-redirect.xq, use the last modified date on the local computer and NOT the one published to GitHub. I'm not sure that this is a good idea. Think about this and potentially change it.

~~This is an appropriate time to add the new organisms whose metadata need to be uploaded to Morphbank in the organisms-to-upload.txt file in the c:\images-processing\images-3-processed-need-morphbank-upload\ folder. See the Morphbank upload section below for more on this. The same with the new images.~~ Note: the modified-x-file.xq queries write ALL modified records, not just the new ones. Make sure that only new records get added to the upload list.

19. The file generate.bat located in the xquery subfolder of the Bioimages github repo contains the list of queries that need to be run in order to generated the Bioimages website. It actually doesn't work well as a batch file because of the amount of time that it takes for the queries to execute, so I end up typing them at the command line anyway. The queries have to be run in order and in the end they generate all of the various index files that are necessary to view the site. They do NOT (as of 2015-08-02) generate all of the various lists for parks and stuff. There needs to be some hack to do that.

Here is the list of queries in the order they should be run:

basex organism-html.xq
basex organism-rdf.xq
basex image.xq
basex index-files.xq
basex index-rss.xq
basex index-sitemap.xq
~~basex eol.xq (see below for details)~~
basex tsn-redirect.xq (generates the new static redirect html files for the permanent URLs for taxonomic names) Does not need to be run unless new species have been added.
basex save-published-date.xq

After the last query is run, GitHub needs to be synched so that the new last-modified date gets updated. (see notes at the end for information about the effects of timing on this).

Encyclopedia of Life

2024-04-24: I haven't been in contact with EOL for years and I'm not sure if they are actually harvesting images any more, or if so, whether they use the same procedure. So I think this part of the process can safely be ignored.

~~EOL harvests the "good" quality images (those on the Vandy server in the gq folder) based on the metadata in the http://bioimages.vanderbilt.edu/eol-harvest.xml file. That file is uploaded to the server along with the rest of the site and EOL just grabs it once a month to harvest any new image files and metadata changes.~~

~~The XQuery which generates the harvest file (eol.xq) screens the images so that only images with a rating of 4 or 5 are presented for harvest. So I can prevent the harvest of poor quality images or images with uncertain ID (e.g. images for which there is only one image per individual) by giving them a rating of 3 or less. In addition, the Query suppresses submission of images whose taxonomic determination is at a level of genus or higher, or for which the~~

~~"suppress" flag for that determination is set to "1". (Eventually, I need to change this so that it does a bitwise logical AND with 1, in order to enable other types of suppression.)~~

~~The query saves the file to the root of the primary website image on my work computer. There isn't really any backup system for it - I'm not sure that one is necessary since it can be regenerated. However, it will get captured in any "snapshot" backups of the site and in the normal website backup process.~~

~~To check on the upload, go to the sign-in page and sign in with EOL credentials: username: baskaufs, pwd: secure3+. Click on "1 content partner". Then click on the Bioimages link. Then click on the Resource & Contacts tab. The date of the last harvest should show up there.~~

20. This step must be done on a network with a high speed connection. Run WinSCP. Login to the session baskausj@lamp.vanderbilt.edu. (The host is lamp.vanderbilt.edu and the protocol is SCP). The Password is the current VUNET password. Navigate to the bioimages directory on the local drive. The bioimages.vanderbilt.edu/html directory on the server should default. Select Commands, Synchronize or click on the Synchronize button (circular arrows). The settings should be Direction/Target directory: Remote, Mode:Synchronize files, Options: Delete files, Preview changes, Criteria: Modification time. After it checks through all of the directories, it will present a list of what it wants to delete and upload. Click OK and it will go to it. When it's done, the website should be updated.

21. Archiving data. There are two queries: image-rdf-all-hack.xq and organism-rdf-all-hack.xq that are in the github/Bioimages/xquery folder on GitHub that are used to generate the giant RDF files that are loaded into the SPARQL endpoint. After running both of them from the command line, the files c:\test\images.rdf and c:\test\organisms.rdf will be generated in the c:\test directory. Open the bioimages-rdf.zip archive in the c:\github\bioimages folder. (Note: I'm now using 7zip to manage the zip file.) Delete the old versions of images.rdf and organisms.rdf . Replace them with the new versions from the c:\test folder. (In 7zip, use the two window view and drag and drop the new files after deleting the old ones. Note: Winzip can be set to "remember" the source folders and extract the files into similar folders. If the box for "Save full path info" is unchecked then the files get extracted into the directory into which the zip is uploaded rather than a subfolder. I've made this setting change because otherwise it really messes up the endpoint.) They will be really big (many MB), so must be compressed. Synch the Bioimages GitHub repo.

NOTE 2024-04-24: This section is obsolete since that triplestore doesn't exist. The AWS Neptune instance that is currently running the https://sparql.vanderbilt.edu/sparql endpoint has its own loading instructions.

~~Next, the RDF files need to be uploaded to the SWWG SPARQL endpoint. Go to the URL https://app.jelastic.servint.net/~~
~~and login with the username: steve.baskauf@vanderbilt.edu pwd: McTy#1r# (check that this is right). Expand vuswwg and Tomcat. Click on the wrench icon (Config), which will open a window at the bottom. Navigate to `opt/tomcat/temp/upload/` Click on the gear and select upload. Upload all of the changed RDF files that need to go into the graph.~~

~~Use Postman to drop the graph that you need to replace by POSTing a SPARQL Update to https://sparql.vanderbilt.edu/sparql using Basic Authentication with usnername: admin and pwd: McTy#1r# and Content-Type header~~ `application/sparql-update`
~~In the body, issue a command in this form:~~

```
DROP GRAPH <http://bioimages.vanderbilt.edu/images>
```

~~to drop the graph, then use:~~

```
LOAD <file:///opt/tomcat/temp/upload/images.rdf> INTO GRAPH
<http://bioimages.vanderbilt.edu/images>
```

~~to load the new data into the graph that was dropped. You may get a 504 gateway timeout, but I think it still completes the load eventually.~~

See https://guides.github.com/activities/citable-code/ for info on the following processes:

To create a new release, open the web interface for GitHub, go to the Bioimages page, then click on " N releases". Click on the Draft New Release button. Use the date for the release version and "Bioimages Release 2015-08-03" (with current dated) for the name. Then publish it.

Go to https://zenodo.org . Sign in with GitHub. Click on the Upload tab. The new release from GitHub should be showing. There is about a 5 minute lag between the upload and when it gets integrated in the Zenodo system. Click on the release name, then click on Edit.
Some fields autofilled.
Type of file: Dataset
Get rid of any extra "Bioimages: " in the title.
Add ORCID 0000-0003-4365-3135
Keywords: RDF, biodiversity, informatics, images
License: Creative Commons Zero (type this in the box)
Then click on Save, then Submit. Click on the view button to see how it looks and get the DOI. I think there may be some time lag again until the data are integrated before the changes show up from their defaults. DOI will be in the format:
http://dx.doi.org/10.5281/zenodo.31130

Note: Zenodo now says that the DOI 10.5281/zenodo.594019 represents all versions and will always resolve to the latest one. So this no longer needs to be updated. ~~After this is done, change the release data and DOI on the Bioimages home page (index.htm). The HTML from the Zenodo page can be copied and pasted into the page.~~

~~Backing up the website and images:~~
~~The "bioimages synch", "bioimages highres synch", and "image processing synch" need to be run on SynchToy to get the files backed up on the Bioimages external drive and so they can get moved to my home computer. At home, run the same synch jobs so that that computer will match the one at work.~~

Button and HTML blob processing (for static organism pages)

1. To create buttons for tree tours, modify the file tour-buttons.csv in the github repo. It's pretty much self-explanatory based on the column headings. The buttons only appear on the static pages for individuals (not the dynamic ones generated by the javascript). So the software has to be forced to regenerate the individual pages. This is most easily done by changing something trivial in the organism's record and then letting the image software flag that organism for re-write.

No longer relevant as of 2024-o4-24 since Morphbank is effectively defunct.

~~**To link highres images to Morphbank:**~~

~~1. Note: morphbank website login is baskaufs with usual insecure pwd.~~

~~NOTE: the image owner/photographer has to have an assigned Morphbank user ID and that ID has to be included in the owner/photographer's entry in the agents.csv file on GitHub. The image owners and image photographers can be different entities. The photographer has to be an agent whose name is in the agents.csv file. The owner also has to be in the agents.csv file and must also have a value for morphbankUserID. If the owner is not likely to ever have a significant number of images on the site, the image can just be uploaded to the highres folder of the website instead of uploading it to Morphbank and with the highres folder URI listed in the ac_hasServiceAccessPoint field of images.csv . In that case, the image would need to be excluded from the list of images to be uploaded to Morphbank (see below).~~

~~The organisms to be included in the Morphbank XML file should be accumulated in the file organisms-to-upload.txt in the c:\images-processing\images-3-processed-need-morphbank-upload\ folder. This is simply a text file with the GUIDs for the new individuals (e.g. "http://bioimages.vanderbilt.edu/vanderbilt/11-402") as the rows. The GUIDs can be obtained from the foaf_depicts column of images.csv in GitHub. Since this file is created manually, the most recently entered individuals should be at the bottom of the file. If the individual ("specimen" to Morphbank) is already present in Morphbank, including the individual record in the XML file is not necessary. The reference to it will be made by via the external identifier reference in the image record.~~

~~Note: If the individual has multiple determinations, the software assigns the most recent determination to the individual. There may be a way in the Morphbank schema to upload multiple determinations, but I haven't investigated that.~~

~~Note: if the determination names don't have an ITIS TSN, they can be given a Morphbank ID instead. It may be necessary to create the taxon record first. Here is how (from Guillaume Jimenez) to modify the XML manually to get it to use the Morphbank identifier:~~

```
The way the determination field is handled does not really allow the use
of a morphbank tsn id using the <morphbank></morphbank> tag.
What you can do however is use the <external></external> tag with
ITIS:[morphbank tsn]. Then it should look for the right entry in the
database and find the morphbank tsn.
So the xml would look like this:
<determination>
    <external>ITIS:999019215</external>
```

~~</determination>~~

~~instead of :~~

~~<determination>~~
~~<morphbank>999019215</morphbank>~~
~~</determination>~~

~~The images to be included should be accumulated in the file images-to-upload.txt file in the c:\images-processing\images-3-processed-need-morphbank-upload\ directory. The format is the same as the individual file (using GUIDs for the image entries). This file can be generated from the dcterms_identifier column of the images.csv file in GitHub. When images are added to the images.csv file following import, they are added to the bottom of the file, so they should be easy to copy and paste.~~

~~To actually run the script, move the images-to-upload.txt and organisms-to-upload.txt to the "c:\test" directory. The Morphbank XML file is generated by an XQuery called morphbank.xq . The resulting file is stored in the "test" directory and is called "morphbank.xml". This file should be copied into the c:\image-processing\images-4-uploaded-morphbank-not-processed\ folder along with the images that need uploading.~~

~~2. The images (and morphbank.xml file) must be uploaded to Morphbank.~~

~~Use WinSCP to upload:~~
~~Open your FTP software (WinSCP).~~
~~host: morphbank.net~~
~~username: mb~~
~~password: Juj7OtDo~~
~~port: 22~~
~~and connect. Within the "writeable" directory, create a directory with a distinctive name if there isn't already one there.~~
~~Drag and drop images and data files from your desktop into the directory you created (bioimages).~~
~~3. Send an email to mbadmin@scs.fsu.edu~~
~~Let us know you've uploaded images and data files and the name of the directory you created in the sftp site. We'll send you a note when the upload is done.~~

~~3. Note: Here is a query that Ken made to check for uploads in the last 30 days:~~

~~http://services.morphbank.net/mbsvc3/request?method=changes&objecttype=Image&keywords=&geolocated=true&limit=1000&firstResult=0&user=&group=Bioimages&change=&lastDateChanged=&numChangeDays=30&id=&taxonName=&format=svc~~

~~It can be used to check whether they've succeeded in doing the upload or not. To work, the query has to have a limit, otherwise it defaults to 10. So if there are more than 1000 images, that will have to be changed. Similarly, if there are more than one update in a month, the change days will have to be changed from 30. Save a dated copy of this as morphbank-image-response-xxxx-xx-xx.xml with the raw images.~~

~~There is an XQuery script called morphbank-results.xq in c:\github\bioimages\xquery that runs this query. It generates the CSV morphbank-ids.csv in the c:\test directory, with the image IRI in the first column and the Morphbank highres SAP in the second columm. As above, the limit is 1000, so change to a higher limit if there are more than 1000 images. Store a copy of this file with the raw images.~~

~~4. Run the image management software, click on "Manage CSVs". In the Import 2-column CSVs…" section, check "Identifier in first column" and uncheck "CSV contains header row. Have pipe "|" as the Column separator, and URL of high-resolution image set in the dropdown. Then use the Select and import 2-column CSVs button and dialog to point to the file "morphbank-ids.csv" in the test directory.~~

~~5. Presumably the addition of the highres SAPs to the image database will trigger an update of the last-modified dateTime. This will cause the affected image files to be re-written the next time the site is published. To push the changes for only the images that have new highres SAPs, copy the first column out of the "morphbank-ids.csv" file, and paste it into the "images-to-write.csv" file. Then run the images.xq query, the queries to regenerate the index files, and the "save-published-date.xq" query (update GitHub). Then go through the rest of the normal process to publish the site and update the RDF.~~

~~Archiving the images that have been transferred~~

~~1. The files that were uploaded to Morphbank should be in the c:\image-processing\images-4-uploaded-morphbank-not-processed\ directory. Include a copy of the morphbank-image-response-xxx-xx-xx.xml and morphbank-ids-xxx-xx-xx.csv files for reference. Burn the images onto a DVD, label it, and put it in the backup stack in my office. Put a copy of the two index files in c:\image-processing\processing-records\morphbank\.~~
~~2. Cut and paste from the morphbank-need-to-go directory to the folder having the appropriate image range within the data\bioimages-hires-archive\ directory on the external hard drive.~~
~~3. Run SyncToy to synchronize the external hard drive with the server.~~

Stuff below here not worked out yet:

Discover Life

Note on 2013-05-17: When I changed the method of specifying the highres links to use the highres.csv file in the [default directory], I never fixed the Discover Life routine. This will have to be done before the next generation is done.

The metadata harvest Discover Life file gets generated by clicking on the Discover Life button. They supposedly will be picked up the next night after the upload to Bioimages, although I don't think the harvest is that often in reality. Because Discover Life harvests the highres images, new images must be available on Morphbank and the highres links have been made to the Morphbank repository before the new harvest file can be generated.

---

8. Note on 2013-06-12: The Discover Live routine is broken and crashes the program when the button is clicked. I haven't yet investigated why. At this point, the new images can be made available to Discover Life. This is done by clicking on the "Discover Life" button on the image management software (see the Discover Life section above). The site needs to be republished to make sure that the new file is uploaded.

Points rating system for image quality (0 to 5 - is zero an acceptable value? Check MRTG).
One point for each thing:
no distracting background
focus
light level
correct orientation for standard view
associated with enough other images for a good ID

Note: I'm not sure that this is really very useful and may abandon it as well as the "rating" metadata property. But the rating is used to control the EOL harvesting.

List processing
1. Edit the Excel version of the list which should be in the process\index-files folder on the BIOIM-FILES flashdrive.
2. Copy the first five columns and paste them into a blank spreadsheet. The first row has headers must be "tsn exemplar startBloom endBloom comments". Save the file in the [default directory] source\lists folder as a .csv file. Make sure the filename is what you want since it will get used for all of the other database files.
3. If this is a new list, you will need to edit a [listName]-header.xml and [listName]-settings.xml file in the [sitepath]list\ folder of the BIOIM-FILES flashdrive. You can start with a copy of a similar list and edit it as necessary using jEdit.
4. If necessary, add the name of the list to the lists-list.csv file in [default directory] source\lists. (There is also a file called list-categories.xml in the [sitepath]list\ folder of the BIOIM-FILES flashdrive which has to be edited manually if new general categories of lists are added.) At the same time, you will need to create an icon for the actual list and an icon for the link to the administrative organization (if any) for the unit described by the list (e.g. the park). There are templates in [default directory] source\image-files . The resulting icon (144 pixels x 144 pixels) should be saved as a PNG file in the [sitepath]list\ folder of the BIOIM-FILES flashdrive. This is the location where it will be found using the name entered in the lists-list.csv file.
5. Open Visual Basic and load the "lists" program which will be in the [default directory]image-manage folder. Run it.
6. Enter the filename of the csv file that you saved in step 2 (without the extension). Click Generate XML. The form window should close when it's done. It creates the XML needed specifically for the list (blooming date, exemplar, etc.) It does not need to be changed if there are only changes of the images included in a taxon. If the program completes successfully, the XML needed to generate the list should be saved in the [sitepath]list\ folder. The "list" program can be closed.
7. Open the image-manage software. Run it. Change anything to cause it to rewrite all of the list files, and then click on Write RDF + Quit. This will write the [listName]-tax.xml and

[listName]-ind.xml files in the [sitepath]list\ folder of the BIOIM-FILES flashdrive that are needed as the database which will run the list if the entire database isn't loaded.
8. ~~If necessary, add code to the javascript in metadata.htm which will launch the list.~~ I'm not sure that there is actually any changes that need to be made in the javascript.
Edit the main entry HTML page on the website (e.g. parks.htm, regional.htm) to add the list and its icon.
9. When publishing the site, I think all of the files that need to be updated are in the [sitepath]list\ folder. So it isn't necessary to synchronize the entire site if only one or a few lists are added or changed.

Occurrence records (2014-07-16)
The output routine only gets run when you click on the little "out occ" button. That runs another routine that generates the occurrence arrays from the individual and image data. That generation routine also gets run when everything is written at the end because the individual web pages needs the occurrence dates. The output files end up in the [default directory]/dwca folder as images.csv (the extension file) and occurrences.csv (the core file). The two XML files in the folder don't ever need to change unless the column format of the CSV files changes. After the two CSV files are generated, they need to be zipped up into the gbif-bioimages.zip file and then copied manually to the root directory of the website and then be published. With the file compression, it is relatively tiny. (about 500 kB).

NOTES:
2015-08-04 I updated Abi's images to say that they were geolocated from Google Maps. But not republished.

Suppress circumstances:
bad determination: should be displayed on the image page along with identificationRemark but should not show up in the index pages.

Notes from Ken about the effects of last-published:
I think there are two things to watch out for. First, you should make sure that when uploading CSVs that contain new records (as opposed to just changes to pre-existing records) that you upload all linked CSVs together, rather than waiting. So if you have new records in images.csv that link to new records in organisms.csv, make sure you're uploading both of those files fairly close together and definitely upload both before you update last-published.xml.

The second and more important thing is when you use my software to export CSVs to publish as a new release. You need to make sure the metadata in my software is up to date with what you have published to GitHub before you export. If you only use my software on one computer and you never make changes to CSVs outside my software again then the metadata in your software will always be up to date. However, if you use multiple computers (and I designed it so you should be able to) it is possible for one computer's database to not be up to date. This can happen when you upload CSVs to GitHub but don't update the last-published.xml. No other computers will know their metadata are out of date and that new versions are available for download. So if you tried publishing from any other computer than the one you used last time to export the latest CSVs that you sent to GitHub, then those recent changes on GitHub will be lost because no other

computer had those new records and there was no indication that new ones were available. Does that make sense? But aside from you publishing the CSV files to GitHub, it really doesn't matter to any other user because there isn't the danger of data loss.

The software currently checks for updated CSVs when it first starts, and at most it will check once every 24 hours. I will add an option to force checking for updates without the 24-hour limit so you can do so before exporting CSVs and know for certain you are in sync with the last-published.xml version on GitHub. If you are using a single computer to process images then this makes no difference and no metadata can ever be lost.

When I release new versions of the software I check last-published.xml to make sure my database is up to date and I set as record in the database with the date-time value in last-published.xml. So as long as you do eventually update last-published.xml sometime after you upload changed CSVs, then my software will see there's a new date in last-published, that it's newer than the date recorded in my database, and notify the user that updates are available. I may be making things more confusing for you, but what I'm saying is that even if there's a lag between updating CSVs on GitHub and updating last-published.xml, as long as you eventually do update last-published.xml then users will get the updated files then.