

Importation Task

- [What You'll Need](#)
- [1. Create a password for your UVaBox account](#)
- [2. Use EditThisCookie to get the key you need to access Mandala Images](#)
- [3. Set up your package installers and python](#)
 - [Open Terminal](#)
 - [Install Homebrew](#)
 - [Install Python](#)
 - [Install Pip](#)
- [4. Install other stuff you need](#)
 - [virtualenv](#)
 - [imagemagick](#)
 - [dcraw](#)
 - [exiftool](#)
- [5. Create a directory for the importation code](#)
- [6. Get the importation code and XML files](#)
 - [For Images Cataloged with Media Pro](#)
- [7. Set up the virtual environments](#)
- [8. Set up the importation code](#)
- [9. Create a collection](#)
- [10. Run the importer](#)
 - [For Images Cataloged Using Adobe Bridge](#)
 - [For Images Cataloged Using MediaPro](#)
- [Troubleshooting](#)
 - [Fixing broken images \(that have already been imported\)](#)

Picking up where you left off

If you have already completed the installation steps below and are just trying to run the importer, follow the steps below. Otherwise skip ahead to "What You'll Need" and proceed from there.

1. Change to the directory where you created your virtualenv. If you're having a hard time remembering where this was, use the command:

```
find . -type d -name venv
```

to locate the virtualenv directory. You can copy/paste that path and use the command:

```
cd PATH/TO/VEENV  
cd ..
```

to change to the right directory.

2. Activate the virtualenv using the following command:

```
source venv/bin/activate
```

If this worked your prompt should be prefixed with "(venv)", like:

```
(venv) sam@multivac:~$
```

3. Get a up to date cookie from EditThisCookie (Step 2 below).
4. Change into the mandala-bii directory:

```
cd mandala-bii
```

5. You should now be ready to run the importer!

What You'll Need

First:

- brew
- python
- pip

Then:

- virtualenv
- imagemagick
- dcraw
- exiftool

Also:

- [EditThisCookie](#) Chrome extension
- A UVaBox account

 If you're completely new to using python or the command line, these instructions assume you're working on MacOS.

1. Create a password for your UVaBox account

Final result: A password

This will let the importer access your account. It can't do this if you're exclusively using NetBadge.

1. Log in at <https://virginia.account.box.com/login>
2. Click on your initials in the top right corner, then choose **Account Settings**
3. Create your new password in the **Authentication** section

Save for later:

The password you created.

2. Use EditThisCookie to get the key you need to access Mandala Images

Final result: A KEY and VALUE

1. Using Chrome, log in to <https://images.shanti.virginia.edu>
2. On the Images page, open Chrome developer tools (**View > Developer > Developer Tools**)
 - The developer tools will open at the bottom of the page
3. Open the **EditThisCookie** tab in the developer tool
4. Look for a row where:
 - The name that starts **SESS**, and
 - The 'Secure' column is checked off
5. Copy and paste the first two cells of the row
 - The first cell (SESSXXXXXXXXXXXXXXXXXX) is your **KEY**
 - The second cell (a string of random digits) is your **VALUE**

Save for later:

The KEY and VALUE you got in step 5.

3. Set up your package installers and python

Final result: python, pip, and brew installed on your computer

These instructions assume you've never worked with the MacOS Terminal app, or any Unix-based system, or any command line interface.

Terminal is a command line interface for your Mac. Instead of clicking on buttons to accomplish what you need, you'll send small snippets of texts instructions to your computer. You'll need to learn the basics of using this interface to run the importer. (Don't worry, if you just copy and paste commands from this document, you should probably be fine...)

Open Terminal

1. Open your applications
2. Open the **Utilities** folder, then click **Terminal**

You should see the terminal now. It will look something like this:

```
Last login: Wed Feb  6 08:11:10 on ttys000
Veronicas-MacBook-Pro:~ veronicakuhn$
```

Install Homebrew

Copy and paste the following code into the terminal, then press **Return**.

```
/usr/bin/ruby -e "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install)"
```

The terminal may ask you to hit **Return** again to confirm the installation, then will request your Mac password. Enter it, then hit **Return** one last time.

Install Python

First, we'll check if you already have the right version of Python. Copy and paste the following code into the terminal, then press **Return**.

```
python --version
```

 If the terminal says you're using python 3 or higher, you can skip to the next section.

```
brew install python
```

if that doesn't work, try:

```
brew reinstall python
```

(You can do this because you installed homebrew in the previous step. Woohoo.)

Install Pip

The command to install python should have already installed pip, but let's make sure. Copy and paste the following code into the terminal, then press **Return**.

```
pip --version
```

You should see the version of pip that's been installed.

4. Install other stuff you need

Final result: These packages installed onto your computer

- virtualenv
- virtualenvwrapper
- imagemagick
- dcraw

virtualenv

Run the following :

```
pip install virtualenv
```

imagemagick

Run the following:

```
brew install imagemagick
```

dcraw

Run the following:

```
brew install dcraw
```

exiftool

Run the following:

```
brew install exiftool
```

Then:

```
pip install pyexifinfo
```

5. Create a directory for the importation code

Using the Terminal, you can create and move into 'directories'. This is just creating and moving into folders on your computer using the command line. For simplicity's sake, we're going to create a folder directly in your documents called **bhutan-import**.

First, navigate into your Documents folder by running the following command in the Terminal:

```
cd Documents
```

Now make your new directory, **bhutan-import**, by running:

```
mkdir bhutan-import
```

Now navigate into the new directory:

```
cd bhutan-import
```

If you open up Finder, you'll see there's a folder called 'bhutan-import' in your Documents.

6. Get the importation code and XML files

Run this code to clone the importation code into bhutan-import.

```
git clone https://github.com/samchrisinger/mandala-bii.git
```

A folder called **mandala-bii** will show up in **bhutan-import** once the code is cloned.

For Images Cataloged with Media Pro

If your images have embedded metadata generated with Adobe Bridge, skip to the next step (7). If the images were managed using Mediap Pro, contact Veronica Kuhn, who will link you to an .xml file from Box. Download the file.

Put the downloaded file inside the mandala-bii folder that you created in the previous step. You can use the good ol' click and drag method, no need to use the Terminal!

Make sure to move the file

That .xml file really needs to be inside the mandala-bii folder, otherwise your importation command won't work!

Save for later:

The name of the xml file you downloaded.

7. Set up the virtual environments

Back in the Terminal, run:

```
virtualenv venv -p `which python3`
```

Then run:

```
source venv/bin/activate
```

If you've done this correctly, you should see **(VENV NAME)** at the beginning of the bash prompt.

8. Set up the importation code

Enter the mandala-bii directory by running the following:

```
cd mandala-bii
```

Then run:

```
pip install -r requirements.txt
```

9. Create a collection

In Mandala, create a collection or subcollection to house the images you're importing. As you do this, check the URL in the collections editor. This lets you find the collection id, which you'll need to direct the import to that collection. For example, for the [Chuma Laptse](#) collection, I can go into the editor and see the URL is <https://images.shanti.virginia.edu/node/1419391/edit>, so the collection id is 1419391.

Save for later:

The collection id.

10. Run the importer

For Images Cataloged Using Adobe Bridge

Use the following code, with some substitutions:

- KEY and VALUE are codes you obtained in step 2.
- YOUR_BOX_USER is your box username, which should be your UVa email address
- YOUR_BOX_PASS is the password you created in step 1.
- /IMAGE/FILE/PATH needs to be replaced with the path to the files in UVaBox.
- COLLECTION_ID is the id of the collection where you're importing the images.

```
python import.py -s Encoded -c KEY=VALUE -i "/IMAGE/FILE/PATH" -u https://images.shanti.virginia.edu/admin/content/bulk_image_import/api -v -cid COLLECTION_ID -cv -cw ImageMagick --ftp --ftp_user YOUR_BOX_USER --ftp_pass YOUR_BOX_PASS
```

For Images Cataloged Using MediaPro

Use the following code, with some substitutions:

- XML_FILE_NAME is the name of the file you downloaded in step 6. **Please make sure you've moved this file to the mandala-bii folder, or the importation will fail!**
- KEY and VALUE are codes you obtained in step 2.
- YOUR_BOX_USER is your box username, which should be your UVa email address
- YOUR_BOX_PASS is the password you created in step 1.
- /IMAGE/FILE/PATH needs to be replaced with the path to the files in UVaBox.
- COLLECTION_ID is the id of the collection where you're importing the images.

```
python import.py -s MediaPro -x "XML_FILE_NAME.xml" -c KEY=VALUE -i "/IMAGE/FILE/PATH" -u https://images.shanti.virginia.edu/admin/content/bulk_image_import/api -v -cid COLLECTION_ID -cv -cw ImageMagick --ftp --ftp_user YOUR_BOX_USER --ftp_pass YOUR_BOX_PASS
```

Don't worry if you get a stream of messages along the following lines at first:

```
File LH_KHOMA_2013_11_21_WALL_PAINTING_IN_BERPA_KHOMA_KHARPHU_LHAKHANG12697.JPG skipped because it has already been imported.
```

But if the run finishes without ever changing, that's a problem.

You can ignore any other messages as long as the importation is still running. Any imported images should be immediately visible at your collection on Mandala.

Troubleshooting

Fixing broken images (that have already been imported)

Sometimes an image is imported incorrectly during importation. This can occur because of an issue with converting the image, or for other reasons. The importer script provides a way to selectively re-upload or "repair" an already imported image. Use the steps below to do so:

Get a list of the images you want to repair- you will need a CSV file with an entry containing the node id, task id, and filename for all of the images. The easiest way to get this list for a catalog is to:

Log in to Acquia Cloud and visit the detail page for the Prod environment (<https://cloud.acquia.com/app/develop/applications/14584a7f-64a0-4816-ad7f-b1d9a6da429a/environments/23134-14584a7f-64a0-4816-ad7f-b1d9a6da429a>)

Copy the "SSH URL" to your clipboard, and from a terminal run:

```
ssh SSH_URL
```

replacing **SSH URL** with your clipboard contents.

Run:

```
cd /var/www/html/shanti.prod/docroot/sites/images/
```

You are about to run some commands in MySQL. **Please proceed with caution.** Updates and deletes are committed to the Production database immediately.

To find your catalog name run:

```
drush sqlq "select distinct(catalog) from bulk_image_import_task;"
```

Then run:

```
drush sqlq "select node_id, tid, filename from bulk_image_import_task where catalog = 'YOUR CATALOG NAME';" > /home/shanti/import.tsv
```

Enter CTRL-d to exit the SSH session

To copy the file down locally, run:

```
scp shanti.prod@ded-18858.prod.hosting.acquia.com:/home/shanti/import.tsv ~/Downloads/import.tsv
```

Then run:

```
cat ~/Downloads/import.tsv | tr "\\t" "," > ~/Downloads/import.csv
```

You now have a file (import.csv) that contains a list of node id, task ids, and filenames for each entry in your catalog. You can now run the importer's repair script:

```
python import.py -s Repair --csv "import.csv" -c COOKIE=VAL -i "PATH TO IMAGES FOLDER" -u https://images.shanti.virginia.edu/admin/content/bulk_image_import/api -v --ftp --ftp_user YOUR_USERNAME --ftp_pass 'XXXXXX' -cv -cw ImageMagick
```