Camelot Documentation

Release 1.6.16

Aug 21, 2022

Contents:

Introduction

Camelot is open-source camera trapping software for wildlife researchers and conservationists.

Camelot:

- Makes classifying camera trap photos quick and easy

- Keeps track of camera trap, camera and species data

- Gives you a head start on data analysis

- Plays nicely with other camera trap software, such as and

- Lets multiple people use it at the same time

- Runs on Windows, OSX and Linux

- Is easy to start using

You can find Camelot's project page here

CHAPTER 2

Getting started

2.1 Prerequisites

2.1.1 Java Runtime

Camelot requires Java 17.0.2 (or later) to be installed on the system it will run on before it can be used.

Camelot is supports the Temurin distribution of Java, which can be downloaded here:

If using OSX, you will need to install the "JDK". For Windows and Linux, you can install either the "JRE" or the "JDK".

Note: If you already have Java installed on your system, we recommend making sure it's the Temurin Java distribution.

2.1.2 Web browser

Camelot supports the latest versions of the following browsers:

- Chrome
- Firefox
- Edge

2.2 Installation

2.2.1 OS-specific releases (recommended)

Camelot has OS-specific releases, which are the recommended way to install Camelot:

- Windows
- MacOS

- Linux

Start Camelot by running "Camelot" application in the extracted archive.

Note: Older computers running a 32-bit OS of any of the above must instead use the Java release below.

2.2.2 Java .jar release

A Java .jar file is also available:

- Servers, 32-bit OSs and others

This release can be used on any operating system, though does not feature an administrative interface.

2.3 On startup

When Camelot is first started, it will open the administration interface & begin to start up the main Camelot application, which can take a few seconds.
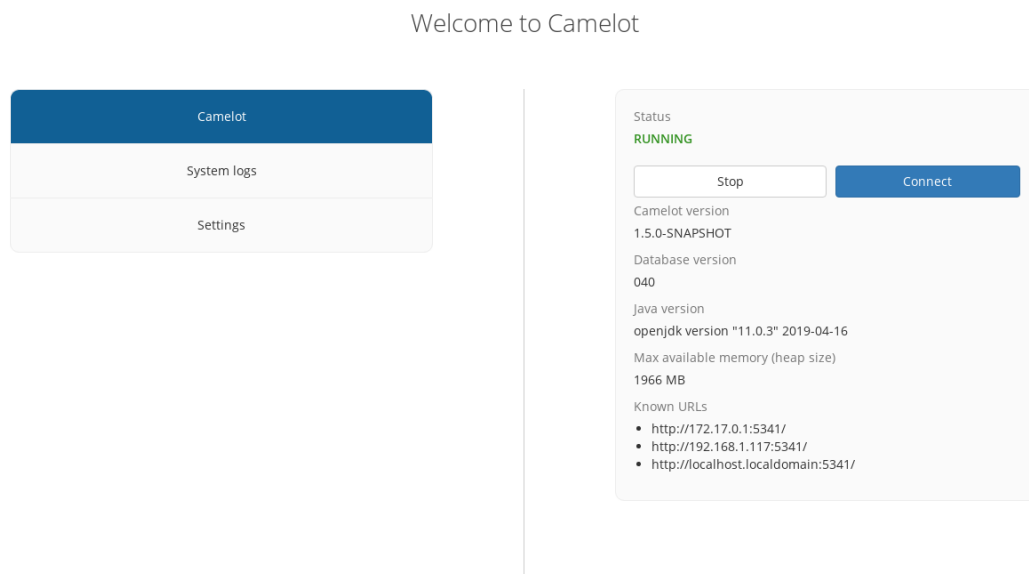
This interface can be used to start and stop Camelot, view logs and configure Camelot. See for more information.

Once the main Camelot application starts up, it will be appear immediately. You can return to the administration interface at any time by clicking the '←' icon in the bottom left hand corner of the screen.

Note: This icon will not be available in the .jar version of Camelot, or when Camelot is accessed via a web browser.

2.4 Web-interface

Camelot supports a multi-user environment where multiple other users can connect to a main Camelot instance via their web browser.

The URL which web browsers may connect to can be found in the administrative interface, under: **Camelot → Known URLs**

A user may access Camelot over the network by entering this URL into their web browser.

Note: In the event that attempting to access Camelot over the network fails, confirm Camelot is running and check the firewall rules of the computer running Camelot.

Terminology

The first time you run Camelot, you'll be taken to the 'Create Survey' screen. Now would be good time to explain what a survey is, and some other important concepts in Camelot.

- **Survey**: a survey corresponds to a research project. All of the data collected will be part of a survey.

- **Camera**: a single, physical camera. Each camera should be given a name and labeled.

- **Camera Trap Station**: one or two cameras installed at a very specific location.

- **Site**: a geographic area. Typically multiple camera trap stations will be in each site.

- **Media**: a photo from a camera trap.

With terminology out of the way, onwards to creating a survey!

Your organisation

This is where understanding the concepts in camelot is invaluable. On this menu, concepts are on the left, and the details about the selected concept are on the right.

You'll notice Sites and Cameras are not *within* a survey. This allows for some more sophisticated, *longitudinal* reports, where data is not only able to be analysed by survey, but also across multiple surveys. For the same reason, reports live in this menu too.
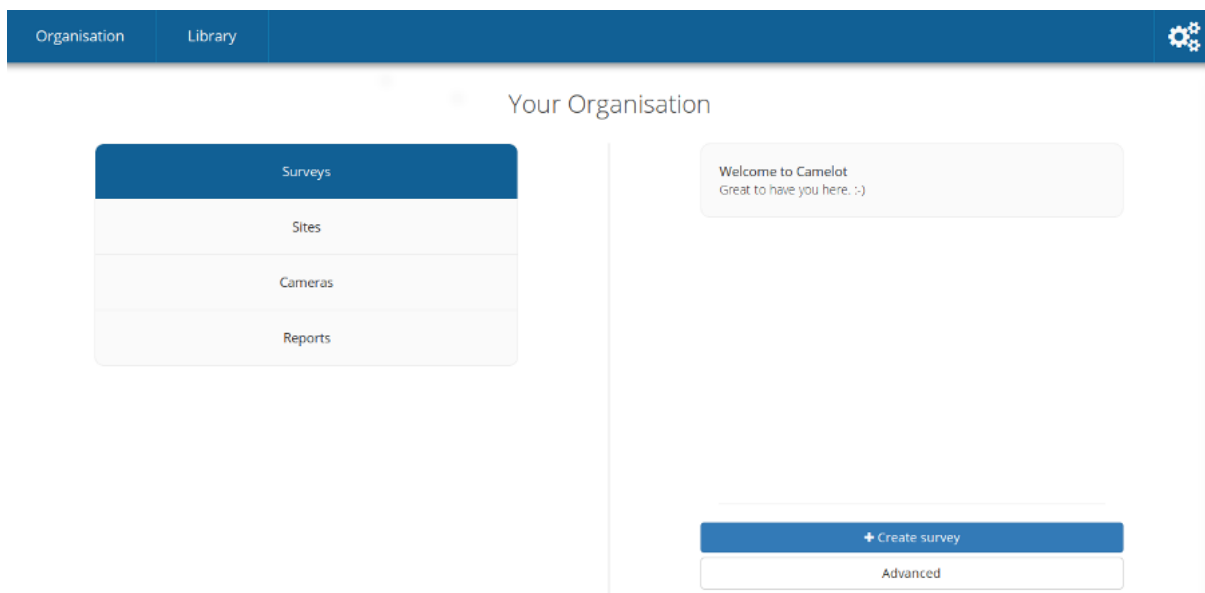
Each concept can be selected from the menu on the left, and then a specific entry navigated to using the menu on the right. You'll find that this is a common pattern in Camelot.
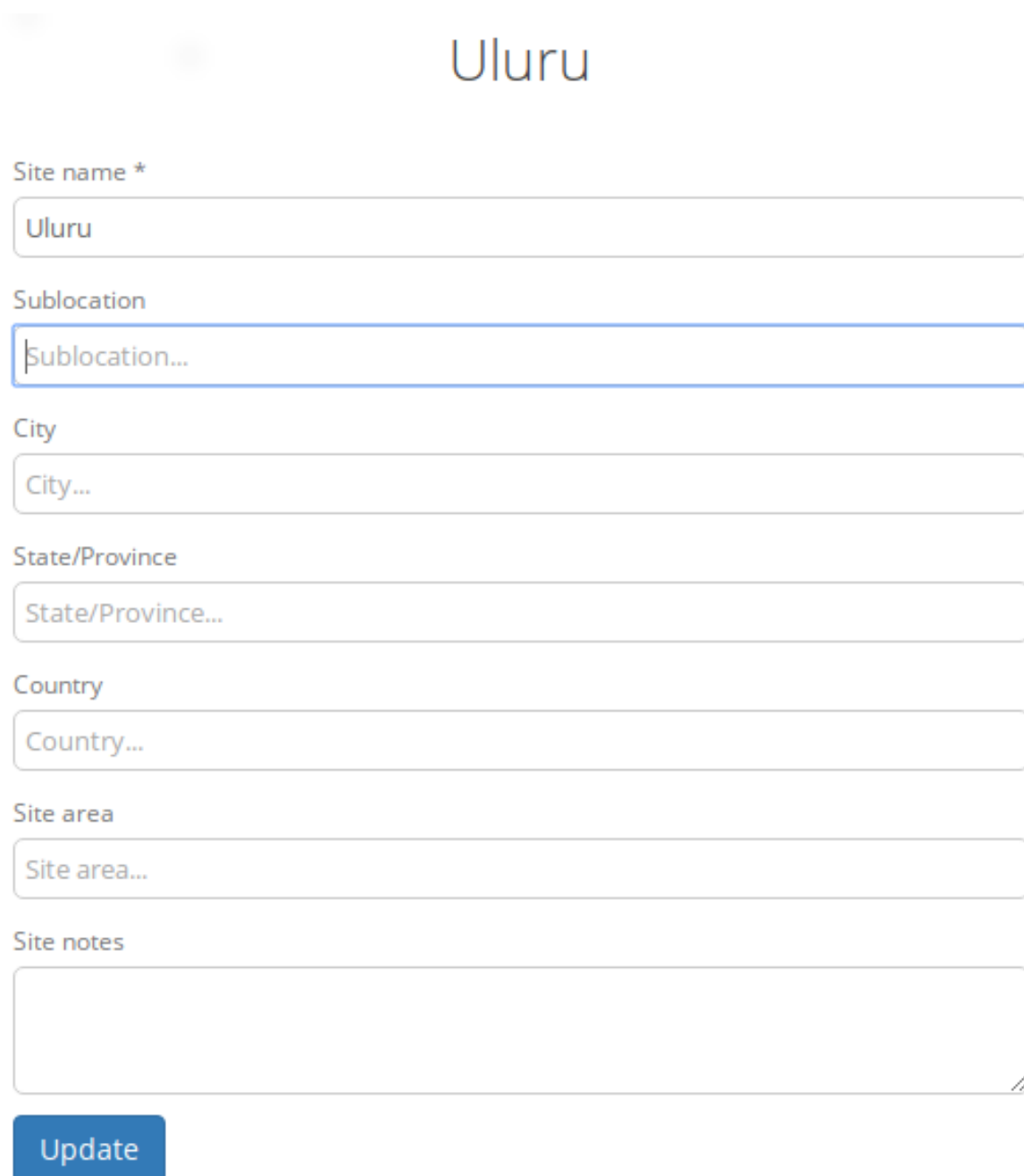
4.1 Surveys

Surveys you'll already be familiar with. The right hand side shows all surveys, and allows you to add a survey. You can click on any survey in the list to manage it, but let's first look at the other concepts.

4.2 Sites

The sites menu will have a very similar feel to the survey menu. The main difference is that a site can be created just by entering a name and clicking "Add". This lets you set up multiple sites very quickly, but if you want to come back and provide more information, you can click on the entry for that site in the list to access all the details.

Adding details to a site is just a matter of filling in the fields and clicking "Update".

Uluru

Site name *

Uluru

Sublocation

Sublocation...

City

City...

State/Province

State/Province...

Country

Country...

Site area

Site area...

Site notes

Update

4.3 Cameras

The camera menu functions almost identically sites menu just discussed. Cameras are added by entering its name and pressing add right in the right-hand menu, though you're free to edit the camera afterwards to give more details too. After creating a camera you'll notice a label alongside each camera (most likely "Available for use" if you've just added one).

A handy feature is being able to filter the list to find all cameras with a particular status or camera name. For example, imagine you have dozens of cameras, many of which are in the field, you can search "available" to see the cameras marked "Available For Use". Camelot ensures the camera status is updated as it is used in, and removed from, camera trap stations.

4.4 Reports

A *report* is an export of data to a CSV. Clicking on a report will take you to a report configuration screen, where you can set constraints for that report (e.g., to report on a specific survey) and then generate the data as a CSV.

Camelot comes with a bunch of reports out of the box. For advanced users, it also lets you build and add your own reports. Reports will be covered in much more detail later.

Creating a survey

The first thing you'll see when the main Camelot interface loads is the "Create Survey" screen. A survey represents a research project and will contain details about your camera traps and uploaded images.

The left hand side is the current survey configuration. You can give a survey a name and description. A survey will often start with one or more species are expecting to be found over the course of the study. Species can be added by searching for the scientific name using the right-hand panel. Behind the scenes, Camelot will automatically set additional details about the species, including its family and common name.

Once ready, click "Create Survey".

CHAPTER 6

Managing surveys

Survey management is the heart of Camelot, and also where it differs the most from other camera trap software. Our research shows that by understanding the next few sections, you'll be well on your way to being a Camelot expert. So listen up!

The authors recommend physically setting up the cameras, and noting the installation details, before recording those details in Camelot. If you can record the details into Camelot while in the field, even better!

6.1 Manage camera trap stations

This menu shows all past and present camera trap stations, and also allows you to add another camera trap station to the field. A quick refresher: a camera trap station is one or two cameras at a specific location.

There are two pages in setting up a new camera trap. The first page has the *essential* information. All of the fields on this screen must be completed, with the exception of the "secondary camera" in the case there was only 1 camera installed. While on the subject of cameras, don't be concerned about the distinction between "primary" and "secondary" cameras: Camelot is a fair master and treats both cameras equally.

Once you're happy with all of the essential details, click "Next". You can now set up any of the optional data for the camera trap. This should be pretty self-explanatory. This data, if provided, would usually be for reporting and analysis purposes later. Once you're happy with everything, click "Create".

If you've used other camera trap software, you may now be starting to notice some differences: the trap station, and the start date, and the cameras – they were all on the one screen! What if you need to change cameras later? Do not fear, Camelot supports all of this. Read on!

After creating a camera, you will see it under the "Manage camera trap stations" menu. All camera traps will be shown on this menu, with camera traps which are no longer active in the field being marked as "Finalised". Each of these cameras can be clicked on to view its details, or if not finalised, to record details about a *camera check*.

Add Camera Trap

Site name *

Uluru ▼

Trap station name *

TS-001

Start date *

Fri, 1 Jul 2016

Latitude *

-25.3444

Longitude *

131.0369

Primary camera *

CAM-010 ▼

Secondary camera

▼

Next ❯

So that's a piece of terminology which has not been introduced until now: a **camera check** is the process of going to a camera trap station, making sure the cameras are okay, collecting the media they've taken and doing any maintenance required (e.g., replacing batteries).

After checking a camera trap station and collecting the media, this information must be entered into Camelot. Recording a camera check requires 3 pieces of data:

- What date was the camera check performed?
- What happened with the "primary" camera?
- What happened with the "secondary" camera?

Camelot facilitates the following scenarios, for each of the cameras:

- Whether or not media was successfully recovered from the camera.
- If the camera was lost/stolen/retired or was taken to be used elsewhere ("available for use"). This will *unassign* that camera from the camera trap station.
- If the camera was previously installed and is still installed, it remains "Active in the field".
- If that camera was replaced with another, you can specify the replacement camera or indicate "No replacement camera"

Also, if there wasn't previously a second camera installed, a new secondary camera can be selected.

When all of the details are correct, click "Submit".

When submitting, one of two things will happen:

1. If there were still cameras assigned to that trap station (i.e., either a camera was still "Active in the field", or a new camera was assigned), Camelot will mark the camera check date as the last date of a the session, and start a new session on this date. Or,

2. If a check was submitted where there are now no cameras at a trap station, that trap station is no longer active and will no longer be available for management. If this happens there's nothing stopping you from adding a new camera trap station at that location later on, but right now, Camelot knows photos aren't being taken, and will take care of finishing it up for you.

Record a camera check each time you visit a camera trap in the field. A camera trap deployment will be finished automatically once there are no 'Active' cameras assigned to it.

Camera-check date

Sun, 9 Oct 2016

Primary camera: CAM-010

Media retrieved?

Media was recovered ▾

What is the new status of this camera?

Active in field ▾

Add a secondary camera

Secondary camera, if any

No Cameras Available ▾

Submit ❯

Phew, okay that was a lot to take in! Reward yourself with a cup of tea and biscuit. The main take away is that Camelot will do a stack of behind-the-scenes book-keeping work for you. Managing camera trap stations is all about telling Camelot what happened, and it will take care of the rest for you.

6.2 Upload media

That last section went on for *ages* about managing camera trap stations, and doing checks and collecting media, but didn't say how to actually upload the photo! What gives? Well, good news, now we're all set up to start uploading photos.

Click on "Upload media" on the main menu, and Camelot will show a list of cameras we've collected media for. Camelot knows this because we've submitted a camera check, for that camera, and we said we had "recovered the media". (If this is all double-dutch to you, see the section on "Managing Camera Trap Stations" above.)
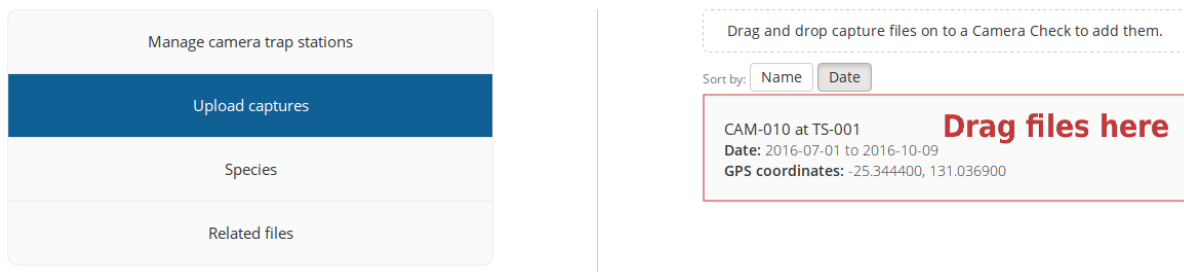
To upload the media, it's as simple as opening your file browser, selecting all the files taken by that camera over the time period shown, and dragging them on to it! Camelot will show a progress bar, telling you how the upload is going. If there are any problems, a "Show Details" link will appear below the progress bar, which will give you the low-down.

You can upload multiple sets of files to a camera, and even be uploading to many cameras at once. You should not leave the "Upload media" menu while photos are uploading, as doing so may interrupt the upload.

Once all of the uploads are complete – you will know this, as all of the progress bars will be full – you can head on to the *Library* to start identification.

6.3 Species

This provides a list of species either expected or encountered during a survey. You can click on any species in the list in order to edit the details about the species, such as its common name, family and mass category. Note that changes made to the species details here will apply to all surveys.

To change the species available in a survey, click "Manage Species" towards the bottom of the right hand panel. From this menu, you'll be able to remove species using the menu to the left and add species using the options to the right.

Removal is simple: if you want to remove a species, click "Remove". Removing a species will only change its availability in the identification dropdown for a survey. If photos in the survey have already been identified with this species, they will still be, and as such these species will also appear on reports for the survey.

To add a species, there's a bit more involved. Camelot gives 3 different ways to add a species:

- Select the species from the dropdown, if it's already known to Camelot.

- Search by scientific name, when a species is not in the dropdown and you would like the details about the species, like Common name and Family, automatically set up.

- The final option is useful when a species hasn't been added to Camelot before, and isn't available in the search. Select "add a new species..." from the dropdown, type the scientific name of the species, and click "Add".

Click "Done" to save your changes.

6.4 Related files

It's nice to be able to keep related data all in the same place, and this is the goal of the related files. You can upload any file you like to Camelot using the file picker towards the bottom of the window, and it will be added to the list of files for this survey and available to view anytime you like.

Library

So you've had camera traps in the field for a while and collected a bunch of photos. Now's where the *library* comes in. All of your photos can be viewed, searched, flagged and identified through the library.

7.1 Viewing photos

The "viewing" portion of the library consists of 3 main parts:
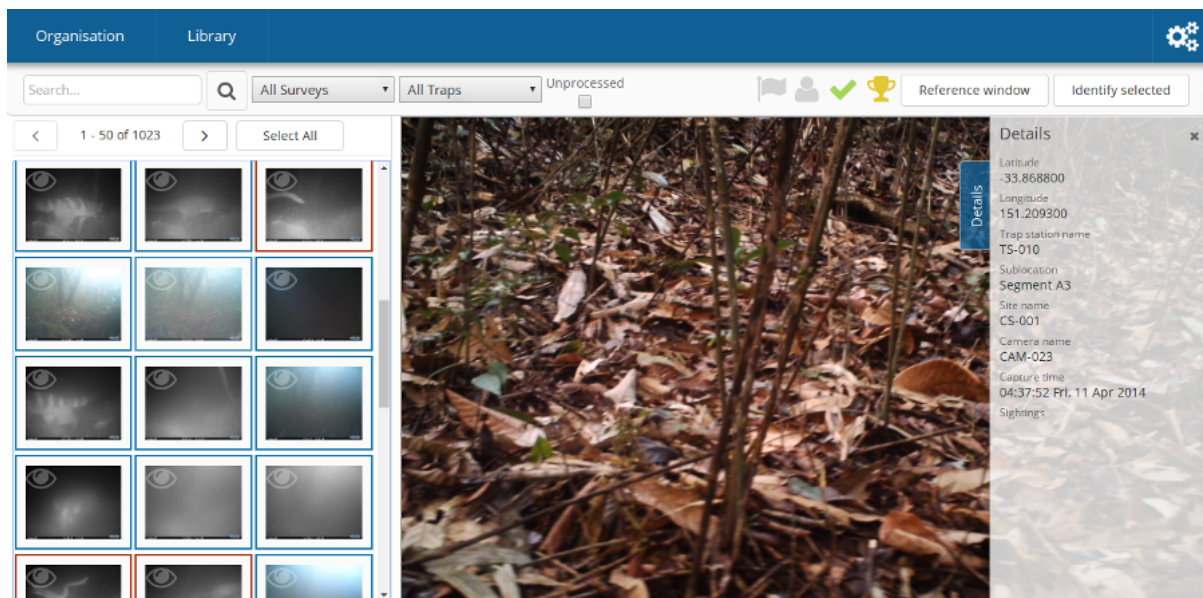
1. A photo can be *selected* by clicking on the photo on the "media collection" view on the left. When a photo has a green border, it is a selected photo.

2. When selecting a photo, a preview for it will be displayed in the preview panel in the middle of the screen.

3. Details for the currently viewed photo are available by clicking the "Details" tab on the right of screen to expand that panel.

Selecting is an important concept in the library, as most actions will apply to the current selection. Multiple photos can be selected by holding the "control" key and clicking a photo. Allowing multiple photos to be selected is one technique Camelot uses to make processing of photos quicker and easier.

Camelot's media collection shows 150 images per page. Clicking the left and right arrows immediately above will change the page of photos displayed. "Selecting all" will select all photos on the page.

A photo can be viewed without being selected (useful if you have multiple photos selected already) by clicking the "eye" icon in the top left hand corner of each photo in the media collection panel.

Finally, you might notice some photos have either a blue or red border around them. A blue border indicates the image is "processed", while a red border is "attention needed". If an image is flagged as processed and attention needed, the border will be red; attention needed takes precedence.

7.2 Enhancing photos

Sometimes photos are over- or under-exposed which makes identification difficult. Controls for brightness and contrast are available to temporarily enhance the image for the purpose of identification by hovering over the bottom left hand corner of the currently viewed image.

7.3 Flagging photos

A flag is one of the four icons towards the top right of the library:

From left to right these are:

- **Needs Attention**: mark the photo is needing further review.
- **Test fire**: the photo was triggered by someone attending to or testing the camera.
- **Processed**: identification of the photo is complete. If a photo doesn't show any species, it can and should be marked as processed without any species being identified in it.
- **Reference Quality**: the photo has a species in it, and the photo of that species is a great example to refer to when doing future identification. (See: "Reference Window")

These can be set and unset by clicking on the respective icon. It will apply to all photos currently selected.

7.4 Identifying photos

Identification in Camelot is the process of indicating which species are present in a photo. When you're ready to identify the selected photos, the mode for submitting details can be accessed via the "Identify Selected" button in the bottom left hand corner of the library.
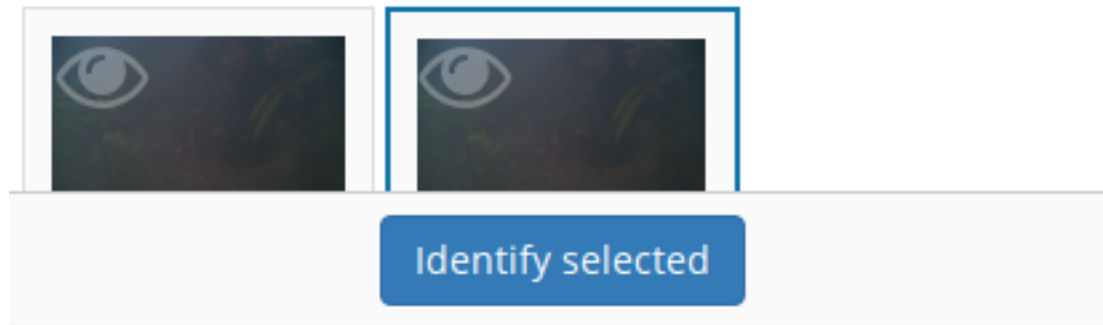
To submit an identification, set the species from the drop down, adjust the quantity if necessary, specify other identifiable details from the appropriate drop down and click "Submit". The identification panel will disappear and the photo will automatically be marked as "Processed".

Sometimes you'll come across a species you didn't expect to see in a survey, and haven't encountered before. You can add this right away by using the "Add a new species..." option in the species dropdown,

and then typing the species scientific name in the input field which appears. This species will also be added in case it's needed in future. Quick and easy.

If you need to know which species have already identified in a photo, these can be viewed (and removed) via the details panel. If there aren't any species identified in a photo, there will not be any listed in the details panel for that photo.

The identification details entered will be applied to all photos currently selected.

7.5 Removing photos

The currently viewed photo, if no longer required, can be removed via the Delete button located at the bottom of the Details sidebar.

7.6 Searching
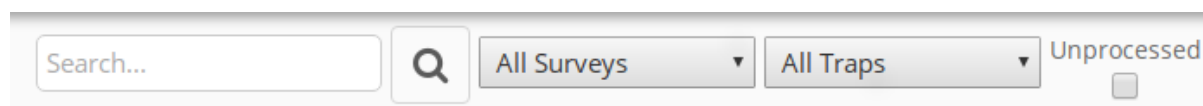
The search bar will change the photos shown in the media collection to only those which match the search. The search bar has a few common search constraints readily-available: the survey, the trap station and a checkbox to show only unprocessed. And of course a text input field where you can just type and press the search button (or "enter").

The text input field deserves a little bit more explanation. It can be used for simple searches, like typing the name of a camera to find all photos taken by it, or the name of a species (or genus). But it can also be used for much more specific searches.

If you wanted to find all photos at a certain sublocation, you could start typing:

```
site-s
```
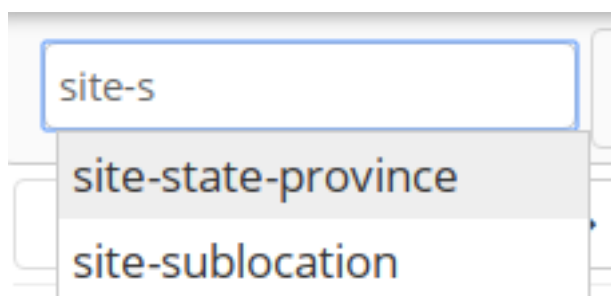

At this point you should see completions below the input field for "site-sublocation" and "site-state-province". Click "site-sublocation" to complete it for you, and, if you had set up some sublocations for sites previously, you'll notice another drop down: all of the sublocations in Camelot.

Here *:* means "is equal to", though Camelot supports other search operators too. For example:

```
captured>=2018-01-01
```

The following operators are available: *:* (aka, ==), <, <=, >=, >, and *!=*.

Sometimes you want to search based on more than one thing. For example, all photos at a site AND featuring a certain species. Can do:

```
site-name:uluru species:"Osphranter rufus"
```

Camelot can also do searches with disjunctions:

```
site-name:uluru | species:"Osphranter rufus"
```

Note the pipe ('|') in the above example: this means "OR" in a search.

Camelot also supports negation. For example, to find all photos which are flagged but do not have any species:

:: flagged:true !species:*

Want more filters, here's our list: http://bit.ly/camelot-library-filters (It's a work in progress)

We think Camelot's search is pretty handy, and hope you think so too.

7.7 Reference window

The reference window is used to help with difficult identifications and make identification more accurate by using photos you have already classified. By clicking the reference window, Camelot will open a new, specialised version of the library in another window which will display only photos marked as "Reference Quality" (see "Flagging Photos").

Once opened, the media available through in the Reference Window will change depending on the species selected in the identification dropdown in the main Camelot window. This can also be useful if there are a several possible species: by selecting the different species in the species identification drop down in the main window, you can quickly preview other photos identifying this species.

A possible workflow is that clear photos are identified first and marked "Reference Quality" where appropriate. Photos which are not so easy to identify can be marked as "Attention Needed". After making a pass through the easy identifications, you can then come back and use the Reference Window, and quality photos you have already collected, to help with making identifications of species in those more tricky photos.

This window is designed to be put on a second monitor, but if that is not available, can also be accessed quickly via "Alt-Tab".

7.8 Keyboard shortcuts

So that trap photos can be processed efficiently, the Library has a number of keyboard shortcuts:

- **Control + d**: Toggle the details panel
- **Control + i**: Open the identify panel
- **Shift + <**: Go to the previous page of media
- **Shift + >**: Go to the next page of media
- **Control + f**: Focus the filter text input
- **Alt + f**: Reapply the current filter

- **Control + a**: Select all media (or select none, if all are selected)

- **f**: Flag the currently selected media

- **g**: Mark the currently selected media as processed

- **r**: Mark the currently selected media as being of reference quality

- **c**: Mark the currently selected media as a camera-check (i.e., test-fire of the camera)

- **Shift + Arrow key** to include media in a range, starting from the last selected image.

- **Control + Arrow key** to toggle the next image in that direction in the current selection.

- **Delete**: Delete the selected media

Holding **Shift** or **Control** and clicking an image with the mouse will perform the equivalent behaviour.

Sighting fields

Some surveys require more specific data to be captured when identifying media. For this, Camelot provides *sighting fields*. Sighting fields are additional input fields available during identification.

The number and type of fields available is fully user-configurable, and can be configured on a per-survey basis. Sighting fields can be searched, assigned during bulk import and included in reports, just like any other field in Camelot.

8.1 Managing sighting fields

Sighting fields can be configured by navigating from within the main Survey menu to:

Settings → Sighting fields

You can get started with creating a sighting fields by clicking **New field** towards the bottom left hand corner. On the right, the configuration for the field will appear. Next we'll go through the options and what they mean.

8.2 Configuration options

Label: the name which will be associated with the sighting field.

Identification key: the identifier of the field. The role of the identifier is to group fields across multiple surveys, and to give a way to refer to the field when searching by field in the library, or when defining custom reports.

Any name consisting of alphanumeric characters and hyphen ('-') is a valid identifier. When being referred to, such as when defining a custom report, the identifier given here will always need to be prefixed by `field-`. (e.g., `field-individual-name`).

By default, the identification key is automatically generated by the label and it is generally fine to accept the default.

Field type: the type of input field which will be presented during identification. Some field types, such as the drop down, have additional options which may be specified when selected.

When selecting the drop-down field type, options in the drop-down may be added and removed. At least one option must be present.

Required: whether, during identification, a value must be provided.

Default value: if specified, pre-fills the input with the given value.

Affects independence?: Camelot will assess whether two given sightings are *independent* when producing some reports. Some fields have a meaningful impact on independence, while others do not. This is best explained by example.

Example: Where a sighting field identifies an individual, a sighting for "Simba" and a later sighting for "Mufasa" always depicts two distinct sightings regardless of the span of time between them. However, a field for the confidence of a sighting – say if one sighting was confidence "5", and another confidence "3" – would not alone be an indication that these are two *distinct* sightings if they were sufficiently close together.

See for the details. Sighting fields also abide the special inference behaviour for blank and "unidentified" values as discussed on that page.

Ordering: defines the order of fields will appear in the user interface. Lower values for ordering appear higher, relative to other sighting fields.

8.3 Templates

Sighting fields from other surveys may be used as a *template*. When there are sighting fields configured for other surveys, they may be used as a template for a new sighting field for the current survey and will be available via a drop-down menu above the "New field" button.

This will pre-fill the fields on the right hand side, allowing modification before the sighting field is created.

Sighting fields will be available in this drop-down so long as the survey does not already have a sighting field with that identification key already defined.

Reports

9.1 Calculated columns

Some columns in Camelot are calculated from existing data, and some of those calculations are worthy of some explanation so that it's clear what that column represents. Here are the most interesting ones:

9.1.1 Independent observations

A sighting is considered independent if two photos with the same sighting are taken at least some time threshold apart. If they are not, it is considered dependent. Whether they are considered dependent or independent effects the value of the data in this column.

Photos may be considered dependent if they are within the same Trap Station Session. That is to say, photos taken by two cameras of a Trap Station at the same time will be **dependent** if certain criteria about the two sightings are the same.

Below are some examples showing the various rules of the calculation (assume T=30 as the threshold). All rules assume sightings are within the same Trap Station Session; if that were not the case, they would always be **independent**.

Sighting	Quantity	Lifestage	Sex	T
Spp. 1	1	Adult	Male	0
Spp. 2	1	Adult	Male	5

These are **independent** as it's a different species. The number of Independent Observations is **2**.

Sighting	Quantity	Lifestage	Sex	T
Spp. 1	1	Adult	Male	0
Spp. 1	1	Adult	Male	40

These are **independent** as while it's the same species, it is separated by T=40. The number of Independent Observations is **2**.

Sighting	Quantity	Lifestage	Sex	T
Spp. 1	1	Adult	Male	0
Spp. 1	2	Adult	Male	5

These are **dependent** as it's the same species, and up to 2 were sighted within the dependence window. The number of Independent Observations is **2**.

Sighting	Quantity	Lifestage	Sex	T
Spp. 1	1	Adult	Male	0
Spp. 1	1	Juvenile	Male	5

These are **independent** as while it's the same species, one is a juvenile and the other an adult. The number of Independent Observations is **2**.

Sighting	Quantity	Lifestage	Sex	T
Spp. 1	1	Adult	Male	0
Spp. 1	1	Unidentified	Unidentified	5

These are **dependent** as while the lifestage and sex are not the same, unidentified values are *inferred*. The number of Independent Observations is **1**.

Sighting	Quantity	Lifestage	Sex	T
Spp. 1	1	Unidentified	Unidentified	0
Spp. 1	1	Adult	Male	5
Spp. 1	1	Unidentified	Female	10

Sighting 2 of Spp. 1 is **dependent** on sighting 1, due to inference on Lifestage and Sex. Sighting 3 is **independent** of both sighting 1 and sighting 2 due to that inference. The number of independent observations is **2**.

Sighting fields, if configured for a survey, may also impact the independence of a sighting depending on its .

The value of the threshold can be found in the .

9.2 Nocturnal (%)

This is simply the number of photos taken at night, divided by the number of photos. The interesting part is what is considered to be "at night".

Night is determined as a time after sunset and before sunrise, given a particular set of GPS coordinates and on a particular day using the sunrise and sunset times as calculated by .

This algorithm does not attempt to account for atmospheric or geographical features, though will typically be accurate to within several minutes of the actual sunrise and sunset times.

9.3 Abundance Index

The Abundance Index is calculated using two pieces of data: the number of number of independent observations, and the number of nights of elapsed operation for a camera trap session, or for the combined elapsed time of all camera trap sessions (depending on the report).

The calculation of this value is then:

```
100 * Independent Observations / Nights
```

Settings

There are two types of settings available in Camelot:

global settings, which affect all surveys, and # survey settings, which are specific to a single survey

10.1 Global settings

A variety of settings can be customised through Camelot's administration tool. See for more information.

10.2 Survey settings

10.2.1 Survey details

A survey can be configured via:

Settings → Survey details

Apart from the name and description, there is the option to configure the *sighting independence threshold*. The sighting independence threshold is the number of minutes between two sightings where they may be classified *dependent*. Dependent sightings will be counted as a single sighting in some reports.

For more information about how the sighting independence threshold affects reports, see .

10.2.2 Sighting fields

Sighting fields can be configured via:

Settings → Sighting fields

Sighting fields allow for the recording of additional information along with each identification. This is a powerful tool, and is described in much more detail .

Bulk Import

A Bulk Import in Camelot provides a means to import a survey's worth of data at once. You're not left to your own devices when it comes to preparing the data for import though, nor is it (entirely) on your shoulders to make sure the data is correct – Camelot will do its utmost to help you with the import. This also means that there's a few things to know about when it comes to bulk import!

Bulk Import

Hello World

Step 1. Download a template of all data scanned in the survey folder.

Path to survey directory

D:\Conservation\Surveys\Camelot-2016

Download template

Step 2. Modify the template as you see fit, then start the import once you're ready.

Import from template

11.1 Step 1: Creating a CSV

The first thing to know is that a Bulk Import is a 2-step process, with the first step being to create a CSV. If you already have all of the data for every media file, you can move straight to the next step. However Camelot is a tool for data management, which usually means the data, well, may not be quite managed yet. But that's okay. A common source of data is one of, or a combination of, spreadsheets, folder names and image tags, and so the focus of Step 1 is collating the latter two pieces of information into a spreadsheet where you can use your spreadsheet prowess to fill in the blanks.

In Step 1, you will enter the path to a folder containing all of the images for a single survey (this path can be copied out of the address bar in Windows Explorer or Finder on OSX). Camelot will take a moment to scan this directory and give you back a CSV file as a download. The CSV file contains a row for every image, and columns for the file's location, the file metadata, and folder names (labelled `Path Component 1-n`).

The resulting CSV often contains hundreds of columns and tens of thousands of rows, and so can take some time to generate. You can take this CSV and move on to Step 2. It may be that there is yet some additional data that needs to be added to the CSV, though in the Step 2 Camelot will indicate the data is required, if there are any mismatches. If adding data manually, see below for the nitty-gritty about what Camelot expects from the data.

11.1.1 Data expectations

Depending on how much data can be gleamed from the aforementioned sources, there may be additional data you need to add in to this CSV manually before it can be uploaded. This might include the Site Name, Camera Name, Trap Station Name, GPS Longitude and Latitude and session start and end dates. This is the guide to *how* that data should be represented in the CSV.

First, every mappable field in Camelot must have its own column in the CSV. For example, Camelot's session start and end dates cannot be represented by a single column which contains a date range. It must be represented by two columns. The exact format of the different types of data are below:

- **Latitude and Longitude**: specified as separate columns and in decimal GPS (where North and East are positive decimal numbers, South and West negative).

- **Boolean fields** (e.g., *Media processed*): any of 'Yes/No', 'Y/N', 'True/False', 'T/F' or '1/0'. Not case-sensitive. Representations may be mixed.

- **Sighting Lifestage**: 'Adult/Juvenile' or 'A/J'. Not case-sensitive. Representations may be mixed.

- **Sighting Sex**: 'Male/Female' or 'M/F'. Not case-sensitive. Representations may be mixed.

- **Dates & Times**: Camelot is forgiving about Date and Time formats (though the author recommends `YYYY-MM-DD hh:mm:ss`). Note that `DD/MM/YYYY` or `MM/DD/YYYY` are not permitted due to room for ambiguity. Be aware that spreadsheet software may default to using these date formats.

- **Path name**: (e.g., *Absolute path*) Must be a path name to a readable file.

- **Integer**: (e.g., *Sighting quantity*) Must be an integer.

The mapping screen in Step 2 offers a separate section for required and optional of fields. For a required field, every cell in the mapped column must have a valid value. For an optional column, some cells may be blank (though non-blank cells must be valid for that field type).

11.1.2 Bulk Import when running Camelot in a client/server model

(If you are using bulk import from the same computer Camelot is running on, you may safely ignore this section.)

The data must be accessible to the server itself so that it can be scanned and uploaded. Clients can provide the path to the survey data local to their machine (e.g., they can provide the path of the data in a network drive, where the contents of the network drive are stored on the Camelot server). Camelot will work out the location of the data relative to the server. If running Camelot in a Client/Server configuration, it is strongly recommended (for both performance and security reasons) to set the ":root-path" configuration property in the config.clj on the server and restart Camelot.

This is a somewhat complex topic. If there are any questions about Bulk Import in relation to running Camelot in a system where separate 'client' computers may perform a bulk import, please ask.

11.2 Step 2: Columns mapping and import

The next step is uploading the CSV and assigning columns mappings from the CSV to fields in Camelot.

Use the file selector to choose the CSV. Camelot will take a moment to check the contents of the CSV

---

and present a screen so that CSV columns can be assigned to fields. Where possible, Camelot will provide default mappings. You may attempt to assign any column to any field in Camelot. If there is a problem, an error describing why the column isn't valid for the field will be presented. This may be useful for working out what corrections may be needed to the CSV before it can be uploaded.
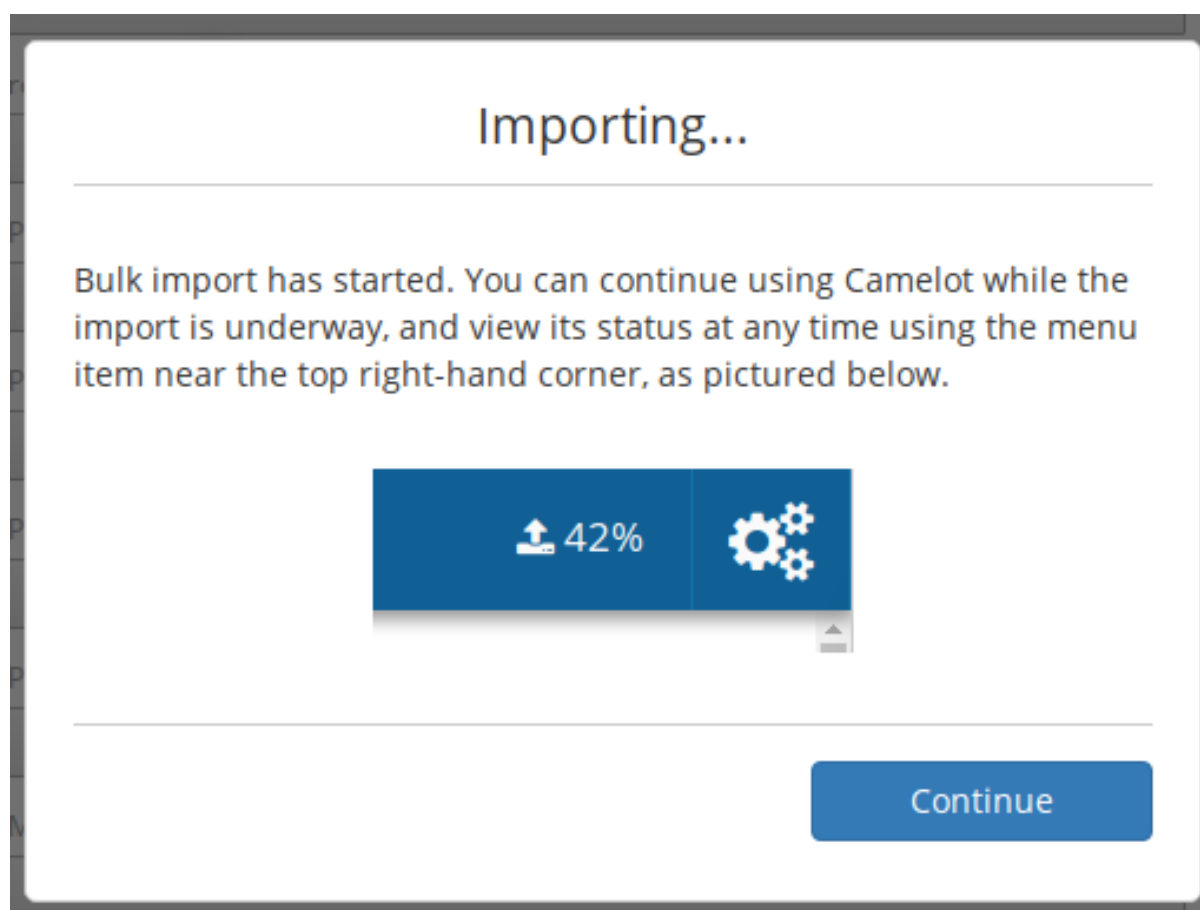

Should you find that corrections to the CSV need to be made, make the necessary changes in the CSV and upload it again. Where possible, all mappings will be preserved and all mapping checks will be performed again against the new CSV data.

Once the mapping configuration is valid, you will be able to submit the data for final verification and to start the import.

11.3 Status of a Bulk Import

As a Bulk Import can be a time-consuming process, Camelot will run it in the background and, in the meantime, Camelot is fully-functional.

The percentage of all running Bulk Imports is displayed towards the top right-hand corner of the screen. You can hover over this for more detail about the upload status, including time estimates and details of

any errors.
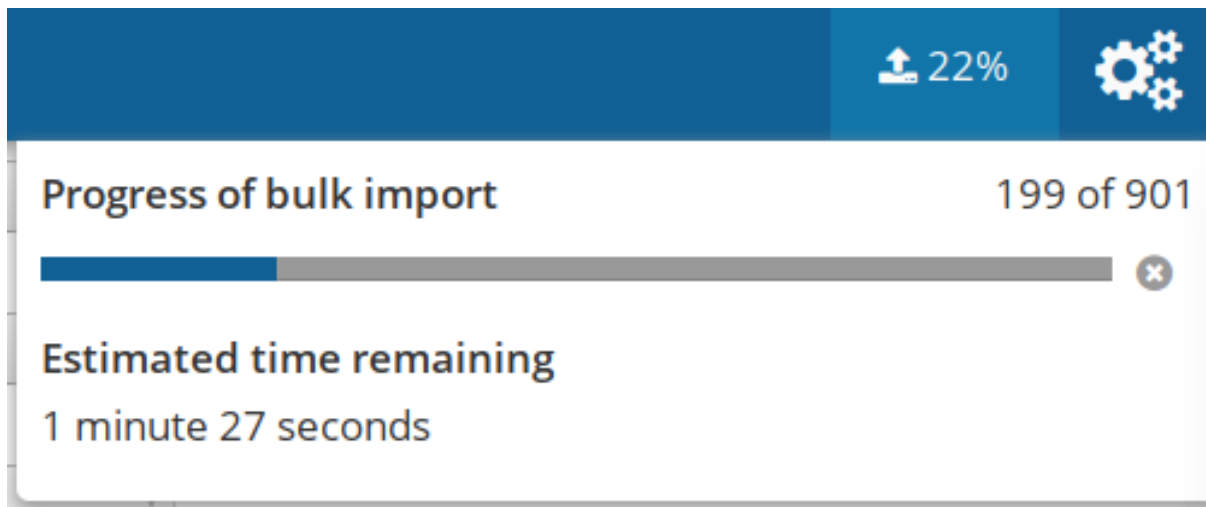
A running import can also be cancelled from this status pop-up using the button immediately to the right of the progress bar, though be aware that doing so will not remove anything already uploaded to the survey.

The survey will no longer be in "Bulk Import Mode" once its bulk upload completes; from this point on Camera Trap Stations can be added manually as usual.

Administration interface

Camelot's administration interface makes it easy to manage Camelot. This includes starting, stopping and accessing Camelot, viewing logs, and making configuration changes.

Welcome to Camelot

Camelot
System logs
Settings

Status
RUNNING

| Stop | Connect |

Camelot version
1.5.0-SNAPSHOT

Database version
040

Java version
openjdk version "11.0.3" 2019-04-16

Max available memory (heap size)
1966 MB

Known URLs
- http://172.17.0.1:5341/
- http://192.168.1.117:5341/
- http://localhost.localdomain:5341/
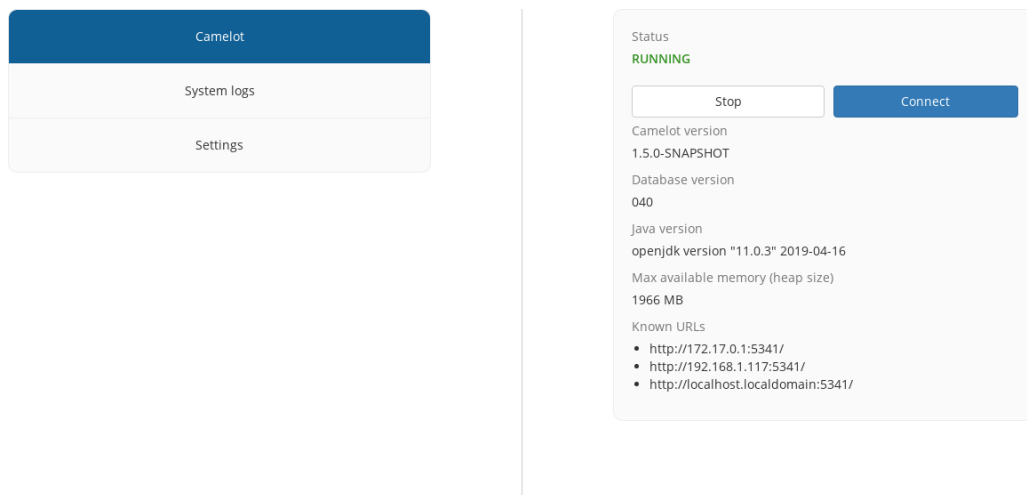
We'll walk through each of the menu items in turn.

12.1 Camelot

The Camelot menu allows you to:

- Start and stop Camelot

- Connect to a already-running Camelot

- View status information about Camelot, such as which addresses it is accessible from.

The latter is most helpful when looking to connect to Camelot from .

12.2 System logs

Sometimes not everything goes as planned. This is where the system logs may prove handy to work out what is going wrong. Should you encounter an error and need to raise , attaching these logs to that request may help to get the issue resolved more quickly.

12.3 Settings

Here you may configure Camelot's global settings. These settings will take effect the next time Camelot is started.

These settings fall into 3 broad categories:

1. Application settings
2. Advanced settings

12.3.1 Application settings

Allows customisation of the way that the Camelot or its user interface behaves.

- **Species name style**: how the name of a species is shown in Camelot's user interface. This can be the scientific name or the common name of a species.
- **Send anonymous usage data**: if enabled, the Camelot team will receive anonymous data about how the software is being used, in order to help improve the software.

12.3.2 Advanced settings

Advanced settings are described below. These are considered advanced as they provide a great deal of flexibility, however may have surprising or undesirable results.

Should you have any doubts about these settings please reach out via the .

Storage locations

These settings allows customisation of where Camelot stores its data.

Important: Camelot will not move your data when customising any of these locations. When changing any of these locations, any existing data you wish to continue to use **must** be moved to this location manually.

- **Installation location**: The folder where the camelot .jar file is stored.
- **Log file location**: The location where Camelot log files are stored.

System configuration

Allows customisation at the system level around how Camelot is started and how it runs.

- **Java command**: the location or name of the "java" command used to run Camelot.

- **JVM maximum heap size**: the amount of memory to allow Camelot to use in megabytes. This may need to be customised if working with .

- **JVM options**: other options to pass to the "java" command. Refer to the documentation of your JVM for more details.

- **HTTP port**: the TCP port on which to run Camelot's HTTP server. This may be useful if you need to use a specific port due to firewall rules, for example.

Multiple datasets

Camelot offers the ability to manage multiple isolated collections of data (a "dataset") within a single running Camelot.

There are other situations where you really want to keep data isolated from each other, for example, as a contractor using Camelot to conduct research for several clients. For this, Camelot offers datasets.

Essentially you can think of each dataset as being a different Camelot database.

13.1 Configuration

In the you can add additional datasets and configure the location of the folders which comprise a dataset (more on this soon). You can also remove datasets. Note that the removal of datasets does **not** delete the data; removal means only that it will no longer be available through Camelot.

Note that after changing dataset configurations, you must stop and then restart Camelot for those changes to take effect.

13.1.1 Storage locations

These settings allows customisation of where Camelot stores its data.

Important: Camelot will not move your data when customising any of these locations. When changing any of these locations, any existing data you wish to continue to use **must** be moved to this location manually.

- **Media location**: where Camelot's images should be stored. **Note** that this should not be the same location as *you* store your images. Camelot will maintain its own copy of any images, folders and images stored in this location **should not** be modified.

- **Database location**: the folder where Camelot will store its database. The Database folder **must not** be modified. If you have an existing database you wish to use, this should be set to the **parent** of the "Database" folder.

Welcome to Camelot

- **FileStore location**: any files related to a survey can be uploaded to Camelot under the "survey files" menu. These files will be stored in this folder. Files and folders in this location **should not** be modified.

- **Backup location**: Camelot will automatically back up its database before doing a database upgrade. These backups will be stored here.

13.1.2 Best practices

This section describes the recommended approach to configuring these locations, and calls out particular problematic configurations which must be avoided.

- **DO** store **Database** on high-throughput, low latency storage, such as an SSD, if available.

- **DO** store **Media** on larger, slower storage (e.g., HDDs or network storage), if available.

- **DO** store **Backups** separately to **Database**.

- **DO** take your own backups, ensuring backups are taken for all storage involved and ensuring Camelot is stopped before these backups are taken.

- **DO** store your backups separately to all Camelot storage locations.

- **DO NOT** store **Database** on storage crossing a WAN (e.g., Google Drive, OneDrive, Dropbox and other online storage). This will severely degrade Camelot's performance and poses a **high risk of unrecoverable data corruption**.

- **DO NOT** store **Database** on local network storage unless disk write caching for that volume is disabled. (This is typically not the default.)

- **DO NOT** rely on Camelot's backups alone; they are intended for recovery from a very limited set of problems only.

- **DO NOT** use "incremental" backups. Any backups taken of **Database** MUST be full backups.

- **DO NOT** store dataset data within the folders configured for other datasets.

13.2 Switching datasets

Once there are multiple datasets configured, you can move between them using the dataset switcher in the top right hand corner.
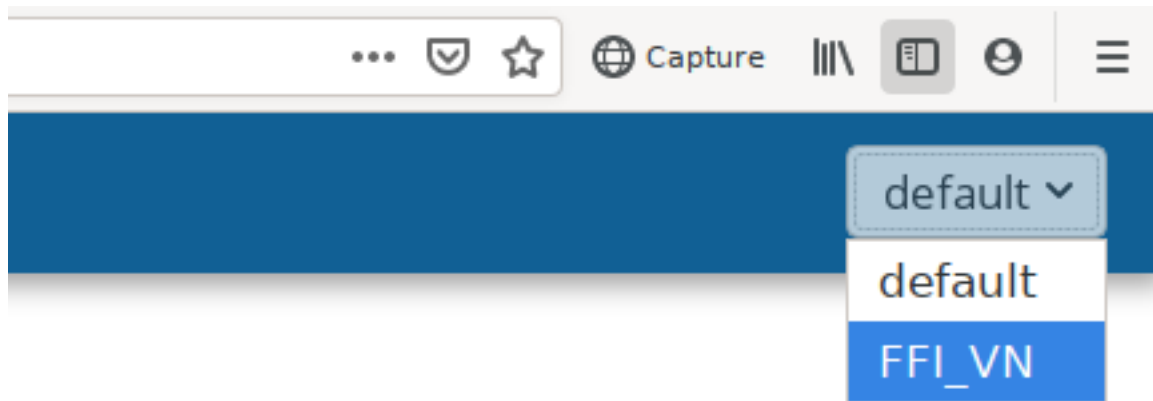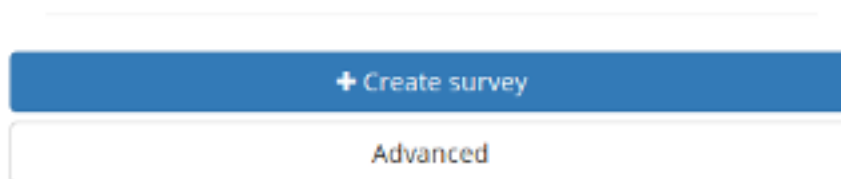
Advanced menu

Occasionally in Camelot you may find "Advanced" buttons, like this one in the Surveys menu:

You might want an Advanced menu should you find an unusual scenario arise, and you find yourself needing to bypass the usual workflows. For example, when attending to a trap station, it is found that the cameras between it and a nearby trap station had been mixed up. As there are many unusual scenarios just like this, Camelot caters for them with a series screens which give fine-grained control over the data.

An advanced menu screen is specific to one type of data. In the above, the screen is specific to Trap Stations. All advanced menus follow the same pattern:

1. On the left is the Sidebar. The sidebar shows you a list of entries. This list contains everything Camelot knows about that type of data (for example, everything Camelot knows about Trap Stations). You can click on an item in the sidebar to view it, or use the "**+**" button at the top to create a new entry.

2. While viewing an entry, you can see a drop down menu which provides a series of *Actions*. This dropdown is very important when working in the advanced mode, and nearly all operations in the advanced mode use this menu, including navigation! In the above example, this dropdown menu it allows for Deleting the selected entry, Editing it, and Viewing its associated *Sessions*.

14.1 Navigating the advanced menu

Let's take a step back, and have a look at a map of Camelot's advanced menus:

```
         Surveys
            |
            v
      Survey Sites  <---------- Sites
            |
            v
     Trap Stations
            |
            v
     Trap Station
       Sessions
            |
            v
      Trap Station   <-------- Cameras
    Session Cameras
            |
            v
         Media
          / \
         /   \
 Photo <'     `> Sightings <--- Species
Metadata
```
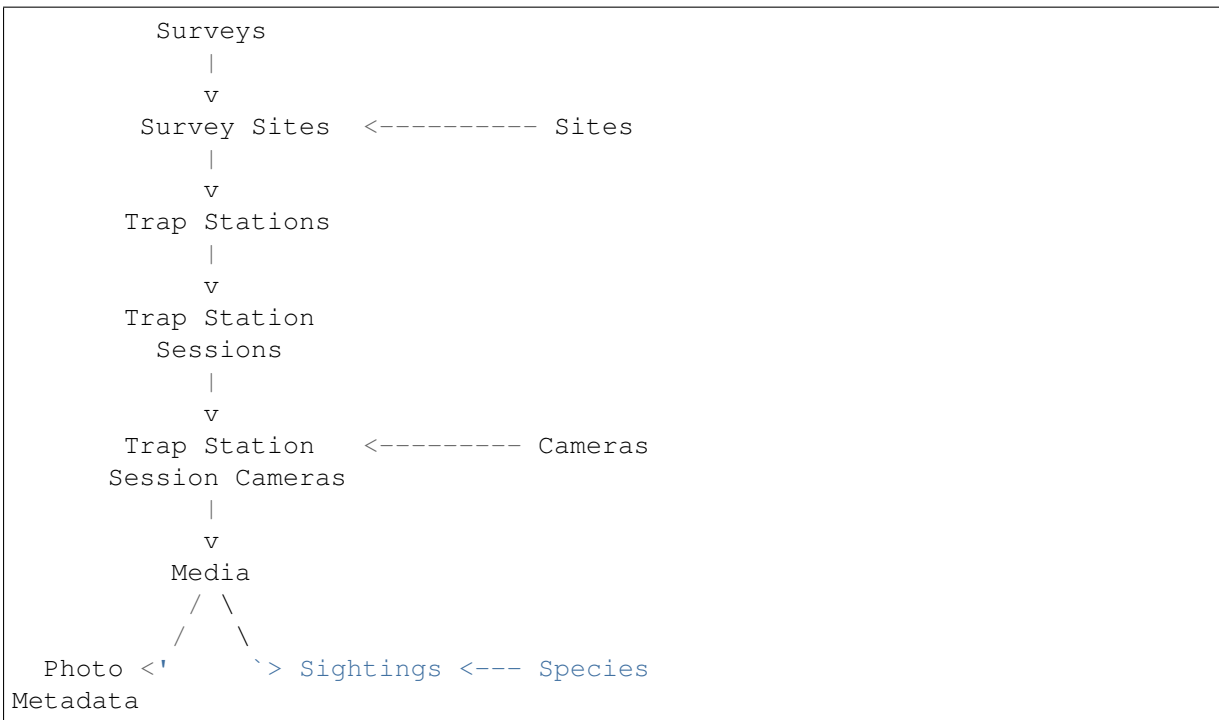
Phew! Hopefully you'll agree that Camelot's normal interface sure does simplify this.

The most important things to take away are these:

- Survey, Site, Camera and Species are the roots of the data structure. To get to these, you must go directly to their Advanced menu. Advanced menus for everything else can be, and often are, accessed via other Advanced menu. (The exception is that there's is an Advanced menu button in the Camera Trap Station details screen, which is a handy shortcut!)

- To go from the Survey level, to the Media or Sightings levels requires going through each level in between.

The pattern to go from one level to the next is:

- select from the Sidebar the entry you want to descend in to, then,

- select the *View . . .* option from the Actions drop-down menu.

14.2 Example usage

Coming back to the example given earlier on when and where to use the Advanced menus: we've just realised that the cameras assigned two trap station (sessions) have been mixed up!

Here's a step-by-step process on how you might go about fixing that:

1. Organisation -> Cameras -> Create a new camera called "Temporary".

2. Access the Advanced menu for Problem Trap Station 1: Organisations -> Surveys -> MySurvey -> Manage Camera Trap Stations -> MyTrap1 -> Details -> Advanced

3. Select the session from the Sidebar where the cameras are known to be wrong

4. Actions -> View Cameras

5. Select the camera wrongly assigned, then Actions -> Edit

6. Set the Camera to "Temporary", as we created earlier

7. Access the Advanced menu for Problem Trap Station 2: Manage Camera Trap Stations -> My-Trap2 -> Details -> Advanced

8. Now we repeat the process: select the session, Actions -> View Cameras, select the wrongly assigned camera, Actions -> Edit

9. Assign the camera we unassigned from Problem Trap Station 1.

10. And we repeat the process again: select the session, Actions -> View Cameras, select the wrongly assigned camera, Actions -> Edit

11. Assign the camera unassigned from Problem Trap Station 2 to complete the swap.

12. Finally, Organisation -> Cameras -> Remove "Temporary" to clean up the temporary camera.

CHAPTER 15

Working with large datasets

You'll have noticed in setting up Camelot, that you probably did not need to provision servers, configure relational databases, network equipment and so on. Camelot does *not* treat working with massive volumes of data as its priority use case. However, we've spent much time ensuring you won't encounter growth pain, no matter how much data you want to throw at it (within reason, of course).

This page describes what you can expect, and recommendations should you be dealing with large datasets.

15.1 Limitations

Camelot has no known upper limit on the amount of data it can support, however some parts of Camelot will take longer to load as the size of the dataset grows. The authors have simulated datasets with **2 million** images to ensure that Camelot will perform well for 99% of datasets Camelot could be used for.

The main performance considerations with this volume of data is performing the initial load of the library, searching the library, and the CPU and memory constraints required to produce reports.

To serve as a guide for how Camelot may perform for large datasets, below is the timing for working with a dataset of 2 million images using a high-end laptop in 2017 (Dell Precision 5510; 16GB RAM; Intel i7-6820HQ; SSD) with a maximum JVM heap size of 14GB:

- **Full Export report**: 368 seconds

- **Library load time**: 13 seconds

- **Library search time (basic search)**: 6.5 seconds

- **Library search time (full-text search)**: 28 seconds

15.2 Considerations for your dataset

This section will describe considerations relevant for using Camelot with larger datasets on a Camelot server, and offer some guidelines for what the authors would expect to be reasonable configurations

under various scenarios.

Data for Camelot's performance is known for only a small number of scenarios, thus the numbers offered are very much guidelines and you may find different considerations important in your use case.

15.2.1 Memory

Memory can be a difficult consideration to judge. Generally speaking, while there is sufficient memory, it has little impact upon performance. However when there is not enough memory, it may result in things which will take considerably longer to complete, or may never complete. There are two main aspects to memory in Camelot: the physical memory available, and the memory available to the JVM heap.

The most important consideration is maximum size of the JVM heap. Regardless of how much physical memory is available to a machine, Camelot may only use the memory available in the JVM heap (and some small additional amounts, as per other JVM configurations), and this is something should be configured manually for large datasets.

What you can expect without configuration

By default, Java will (typically) make available 25% of the total memory available in your machine to Camelot. This gives the following approximate minimum memory requirements for various report sizes:

Dataset size (images)	Heap size (MB)	Physical memory (MB)
20,000	256	1024
50,000	512	2048
100,000	1024	4096

From Camelot's perspective, the important column here is the Heap Size, which is the amount of memory which it can actually use. However, Camelot can use any and all available physical memory with the appropriate configuration of Java.

Configuring the Heap Size becomes increasingly important for resource-efficient use of Camelot.

Configuration

The JVM heap size should not exceed the size of physical memory available, and ideally should not impinge upon the resources required by other applications on the machine Camelot is running upon.

Use the to set the heap size to your needs.

As a (very) rough guide, the authors suggest an additional 800MB of heap space for every 100,000 images, with a starting heap space of 1GB.

15.2.2 Storage

High throughput and low latency storage for a database is always nice to have, though should not be strictly necessary to use Camelot with large datasets. However, using Camelot with remote storage, for example, running Camelot on a laptop and connecting to a database stored on a NAS over wifi, is unlikely to result in nice performance characteristics. Effort should be made to reduce the latency and maximise throughput between where Camelot is running, and where its database resides.

That said, much of the volume of data for an installation of Camelot is occupied by images. Images may reside on lower-throughput and higher latency storage as it does not impact significantly on Camelot's performance profile.

15.2.3 Concurrent users

We currently do not have sufficient data on how the number of concurrent users impacts upon Camelot's performance in Real World usage. It depends heavily on the usage of those users. Should you be using Camelot with a large number of users, and have any questions or feedback about your particular use case, please do reach out via the .

15.3 Client machines

This section applies to client machines: those connecting to a Camelot server, which do not run a copy of Camelot themselves.

Generally speaking, any computer able to achieve an acceptable degree of responsiveness should be a fine candidate for accessing Camelot running on a remote machine. The main consideration of client machines is less-so performance, than it is screen resolution. In common usage, there should be no discernible degradation on performance for large datasets for client machines.

Integrating camelot

Camelot is designed with integration in mind. This page describes the process around some common integrations.

16.1 CamtrapR

CamtrapR assumes images to be stored in certain ways in order to read-in data for analysing. Camelot itself does not store data in this way.

Instead, Camelot has two CamtrapR-specific reports:

1. Camera Trap Export

2. Record Table

These reports provide data in an equivalent format to CamtrapR's `recordTable` and `CTtable`

Here's an example of how these might be used in R to generate a species richness plot:

```
recordtable = read.csv("record-table_yyyy-mm-dd_xxxx.csv") # read the
↪Record Table report
camtraps = read.csv("camera-traps_yyyy-mm-dd_xxxx.csv") # read the Camera
↪Trap Export report

Maps <- detectionMaps(CTtable     = camtraps,
                      recordTable  = recordtable,
                      Xcol         = "gps_x",
                      Ycol         = "gps_y",
                      stationCol   = "Station",
                      speciesCol   = "Species",
                      printLabels  = TRUE,
                      richnessPlot = TRUE,
                      speciesPlots = FALSE
)
```

Networked usage

To use Camelot on multiple computers, the approach is to run Camelot on one computer, and then connect to that computer over the network from all other computers needing to use it. That is to say, Camelot itself is only ever running on one computer, and other computers merely access Camelot over the network. This is called a Client/Server model.

Camelot can be accessed by other computers on the network using the *address* of the computer actually running Camelot. The URL which web browsers may connect to can be found in the , under: **Camelot → Known URLs**.

These addresses will change for each computer network Camelot is used on. Alternatively, you can get the hostname or IP Address of your computer manually (the process to do this is specific to each operating system & outside the scope of this guide) and appending ":5341".

Using a similar approach, Camelot can be accessed remotely over the Internet. This often necessitates use of a VPN or similar (configuring a VPN Server is also outside of the scope of this guide and may require professional IT services to set up).

Community

To keep up to date with any new releases, or if you have any questions that haven't been answered here, please join the Google Group:

CHAPTER 19

License

This program is free software: you can redistribute it and/or modify it under the terms of the GNU Affero General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Affero General Public License for more details.

You should have received a copy of the GNU Affero General Public License along with this program. If not, see <https://www.gnu.org/licenses/>.

CHAPTER 20

Acknowledgements

Upgrading Camelot

21.1 Upgrade Process

Take a Backup Take a backup of Camelot's Database directory. Camelot's database can be backed up by copying the the Database folder while Camelot is not running. The default location of this folder varies, and is <a href="https://camelot-project.readthedocs.io/en/latest/datasets.html#storage-locations">documented here<a>.

<a href="https://camelotproject.org">Download the latest version<a>, unzip and put the file contents into your exisiting Camelot folder and start Camelot using the process you usually would.

Camelot will commence an upgrade to its database the first time it is ran, it may mean that it will take a little longer to start up than usual. The time depends on the amount of data, though should usually be a number of seconds and no more than several minutes.

21.2 Rollback to a Previous Version

This is not recommended, but sometimes required. Check with Chris & Heidi in the Google group first.

Configuration Internals

This setting describes the internals of Camelot's data storage locations and configuration files. This is intended as a resource for very specialised use cases and for IT professionals.

Camelot has two directories: one for configuration, and one for data storage. The location of these directories depends on the OS.

22.1 Data locations

By default, Camelot stores all data under the current user's home folder for the operating system.

22.1.1 Data

- **Windows**: %LOCALAPPDATA%\camelot

- **MacOS**: $HOME/Library/Application Support/camelot

- **Linux**: $HOME/.local/share/camelot

22.1.2 Configuration

- **Windows**: %APPDATA%\camelot

- **MacOS**: $HOME/Library/Preferences/camelot

- **Linux**: $HOME/.config/camelot

22.2 Data directory

The data directory will, by default, contain three subdirectories: `Database`, `Media` and `FileStore`. Database is an Apache Derby database. Imported media is not stored in the database, but in the `Media`

folder. Finally, the `FileStore` contains files for the Survey's "Related files" feature.

A custom data directory can be set using the `CAMELOT_DATADIR` environment variable. The Database and Media directories will be created (if necessary) and stored within that nominated directory. If `CAMELOT_DATADIR` is not set, Camelot will fall-back to using the standard locations (as above). The usage of `CAMELOT_DATADIR` can be selectively overridden by specifying paths to alternate directories, as described below.

Each of the `Database`, `Media` and `FileStore` directories should be backed up routinely.

22.3 Config directory

22.3.1 config.json

`config.json` is the global camelot configuration file. Some values in this file can be set via the administration interface in Camelot. Care should be taken if editing this file manually.

`config.json` supports the following properties:

- `datasets`: describes the datasets configured. Each key is the dataset name, whose value is a *dataset* object. A *dataset* object has two keys:

  - `name`: the display name of the dataset. This value is a string.

  - `paths`: the paths for the dataset. The value of paths is an object. The following dataset-level paths are supported:

    * `media`: the folder path to the directory where media is to be stored.

    * `database`: the base to the folder which is to *contain* a `Database` directory.

    * `filestore-base`: the folder path to where Camelot should keep its uploaded survey files.

    * `backup`: the folder path to where Camelot stores backups.

- `paths`: describes the various application-level paths to be used by Camelot. The following paths are supported:

  - `log`: the folder path to where Camelot should store its logs.

  - `application`: the folder path to where Camelot can find a Camelot .jar file.

  - `root`: the root folder which Camelot can search for external data. For example, for scanning for bulk import

- `server`: an object describing Camelot server-specific configuration

  - `http-port`: a number describing the port to run Camelot's HTTP server on (default: 5341)

  - `media-importers`: the number of concurrent threads for importing media (default: 4)

  - `max-heap-size`: a number representing the desired maximum heap size (in MB) for the Camelot server JVM (default: <blank>)

  - `jvm-extra-args`: additional arguments you wish to pass to `java` when starting the Camelot server. (default: <blank>)

- `java-command`: the path to the `java` executable on this system (default: `java`)

- `open-browser-on-startup`: whether the Camelot administration software should automatically view Camelot after startup (default: `true`)

- `send-usage-data`: whether to publish anonymous usage data to the Camelot team for helping to improve the software (default: `false`)

- `species-name-style`: the style to show species names in the Camelot user interface; options are `scientific` or `common-name` (default: `scientific`)

All of the above properties are optional and default values will be used where not specified.

In practice, a simple `config.json` might look as follows:

```json
{
  "datasets": {
    "default": {
      "name": "Default",
      "paths": {
        "media": "/path/to/ext-hdd/Media"
      }
    }
  },
  "paths": {
    "log": "/path/to/ext-hdd/Logs"
  }
}
```