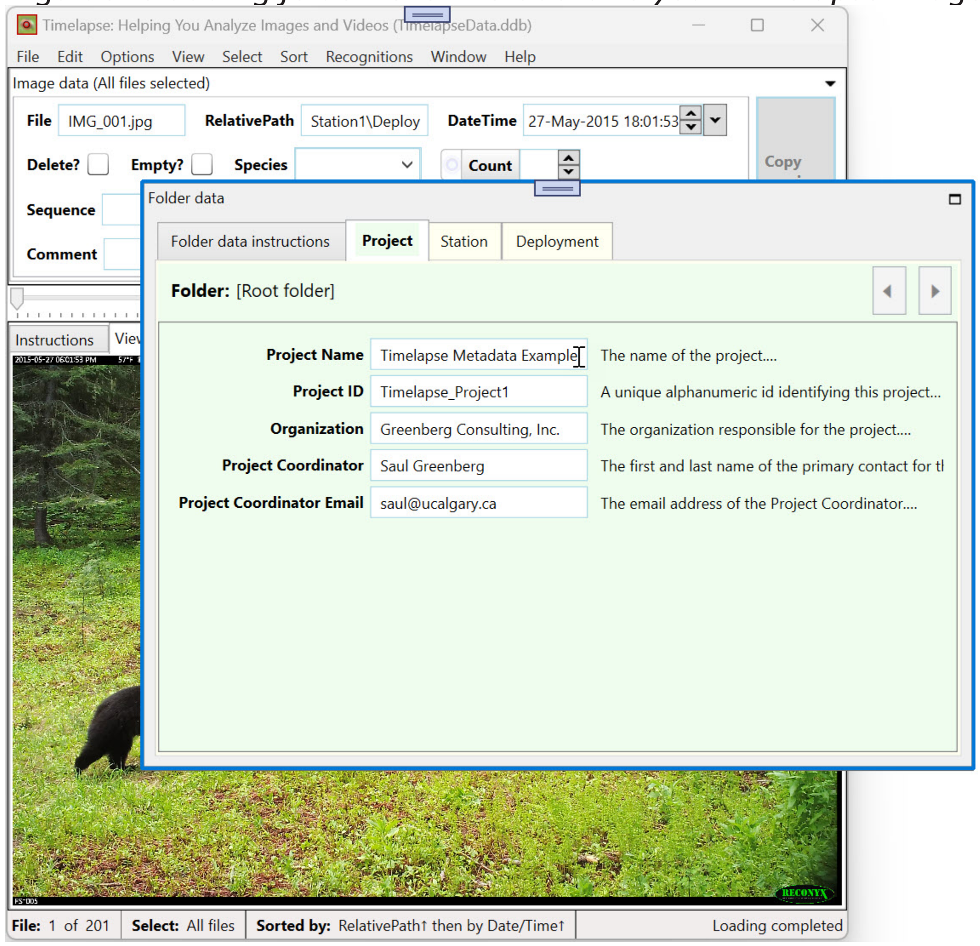
Timelapse Metadata Guide

A guide to adding folder-level metadata to your Timelapse image sets



File: IMG_001.jpg RelativePath: Station1\Deploy DateTime: 27-May-2015 18:01:53

Delete? Empty? Species Count

Sequence

Comment

Folder: [Root folder]

Project Name	Timelapse Metadata Example	The name of the project....
Project ID	Timelapse_Project1	A unique alphanumeric id identifying this project...
Organization	Greenberg Consulting, Inc.	The organization responsible for the project....
Project Coordinator	Saul Greenberg	The first and last name of the primary contact for tl
Project Coordinator Email	saul@ucalgary.ca	The email address of the Project Coordinator....

File: 1 of 201 **Select:** All files **Sorted by:** RelativePath↑ then by Date/Time↑ Loading completed

Saul Greenberg
Greenberg Consulting Inc. / University of Calgary
saul@ucalgary.ca

Part 5 of the *Timelapse Manual Series*. Last updated May 1, 2025, Timelapse Version 2.3.3.0

Timelapse Metadata Guide

A guide to adding folder-level metadata fields to your Timelape image sets[1]

This guide explains why you may want to associate folder-level data fields (metadata) with your image sets, and how to do so.

Folder-level metadata can be extremely useful, although it does depend upon how you organize your image set folder structure. For example:

- Metadata can describe organizational information about your image set, such as project descriptions, camera station specifics, deployment details, and pretty well anything else.
- It can implement a ***metadata standard*** shared within or between organizations and agencies.
    - » If you adhere to that standard, you will be able to share your data with others, and conversely understand the data they may want to share with you.
    - » You may also be able to use data analysis tools that understand that standard.

Your decision on including folder-level metadata as part of your Timelapse image set is entirely optional. Timelapse does not require that you use it.

Consequently, this guide is only of interest if:

- you want to understand enough about metadata to make a decision about including it, and
- how to create metadata levels and fields if you decided to do so.

The Timelapse Template Guide is a companion to this document. Several of its sections detail how to specify folder-level data in your template.

Important note. Using folder data adds constraints to how TImelapse is used.
- The image set's folder structure must match the folder-level organization for this to work properly.
- This can make revisions to your folder organization difficult to do.
- There are added nuances in dealing with advanced Timelapse aspects, such as creating and using master databases.

Read the *Important considerations* section before deciding to use folder data.

[1]What you see when you run Timelapse or other software to examine the database may not exactly match the screen images in this guide, due to software updates made after these screen images were taken. These differences should not affect your general understanding.

Table of Contents

Credits. Camera trap images used in this manual are used with permission or obtained from public repositories. *Sources*: Parks Canada; Lila Science- Idaho Fish and Game data set; Lana M. Ciarniello, Aklak Wildlife Consulting; UTC – French Guiana – French Agency for Biodiversity. Everything else ©Saul Greenberg, 2022 to present.

Introduction

Metadata is broadly defined as 'data about data'. Consider Timelapse from that perspective. Each image or video in your image set is just data stored as bits in a file and presented pictorially via Timelapse. When you use Timelapse to enter data that describes aspects of a particular image, that entered data is actually *image metadata,* as it is data that describes each image's data.[1]

Yet there may be additional metadata that you want to record that would be unwieldy to do at a 'per image' level. For example (and this is just an example), you may also want your metadata to include:

- ***project metadata*** describing details about the project, e.g., who is in charge, the project's purpose, and contact information;
- ***station metadata*** about each camera station placement, such as the latitude/longitude coordinates specifying the exact camera location, why that location was chosen, and information about the camera used;
- ***deployment period metadata*** defining an SD card retrieval, such as its images' start/end dates, the camera used and its settings, and issues if any.

The metadata example above can be viewed as an ***information hierarchy*** containing three levels. A single project (root *level 0*) is the root of the hierarchy. It can contain and describe many camera stations (*level 1*). Each camera station can contain information about several deployment periods (*level 2*), one deployment for each time the camera is serviced and images gathered from it. The actual image data would be associated (or stored under) each deployment. Other information hierarchies may be simpler (fewer levels) or more complex (more levels) than this example.

Timelapse can structure and store hierarchical metadata defined by an information hierarchy. Essentially, its hierarchy defines expectations of how you would structure and construct your sub-folder hierarchy within your image set. As you use Timelapse to view your images, Timelapse would display both ***image-level data*** fields and f***older-level metadata*** fields that you can fill in.

To create folder-level metadata fields, you would use the Template Editor to define the information hierarchy as ***folder levels*** (e.g., Level 0: *Project*, Level 1:*Station*, Level 2:*Deployment*). For each folder level, you would then define its ***folder-level data fields*** (e.g., a *Project name* data field within the *Project* level). Details on how to do this are included in the Timelapse Template Guide.

Its not as complicated as it sounds. This guide walks you through the process of associating metadata with your folders. If you follow the tutorial steps, you should find this fairly straight forward. The hardest part is actually deciding what metadata is of interest. This only has to be done once by the project manager.

To illustrate, let's assume the template defined an information hierarchy based on the project/station/deployment hierarchy above. You would then construct your image set folders beginning with your root folder (the one containing your Timelapse Template .tdb file) to mirror that hierarchy. That is, your:

🗀*root folder* mirrors its *project data* definition (root level 0);
    🗀*station subfolders* mirror its *station level* definition (level 1)
        🗀*deployment subfolders* each mirror its *deployment level* definition (level 2)
            ▤▤▤*retrieved images* would be located in each deployment subfolder.

Depending on the stations and deployments you actually have, the mapping of your actual folders to the folder-level data hierarchy may appear as follows.

🗀***Root folder*** -> project metadata
    🗀***Station1*** -> station metadata describing *Station1*'s location, etc.
        🗀***Deployment1a***-> deployment metadata for *Station1*'s 1st deployment
            ▤▤▤*deployment 1a's image or video files*
        🗀***Deployment1b***-> deployment metadata for *Station1*'s 2nd deployment
            ▤▤▤*deployment 1b's image or video files*
    🗀***Station2***-> station metadata describing *Station2*'s location, etc.
        🗀***Deployment2a***-> deployment metadata for Station2's 1st deployment
            ▤▤▤*deployment 2a's image or video files*
    🗀***Station3...***

This hierarchical folder/metadata definition means that Timelapse can easily display all metadata fields relevant to an image by examining that image's location in the folder hierarchy. For example, if you were viewing an image in the *Station1/Deployment1a* folder, Timelapse would display and fill-in folder-level data fields associated with the *Deployment1a, Station1* and its *Root* folders. Timelapse uses the same method to export metadata into CSV files, discussed later.

1. Using metadata is entirely optional. If metadata is not needed for your projects, you don't have to do anything. Timelapse will work fine without it.

2. While Timelapse defines a folder-level structure that allows you to add metadata such as described above, it is agnostic to the particular metadata you want to record. It is up to you to decide what your hierarchical structure is, what metadata fields you want, and what your analysts will see and fill in.

3. Folder names can be whatever you want, although its good to have them reflect their roles in the folder-level hierarchy.

4. Organizations can define a ***metadata standard*** for its image sets, including a folder naming structure. That standard can be deployed by using a common template. A metadata standard can also span multiple organizations, where it promotes data consolidation, data consistency, and ultimately the sharing of data.

[1]Cameras also write metadata about images, which is embedded in the image file. The metadata described in this guide differs, as it comprises custom data fields that are expected to be filled in by the analyst.

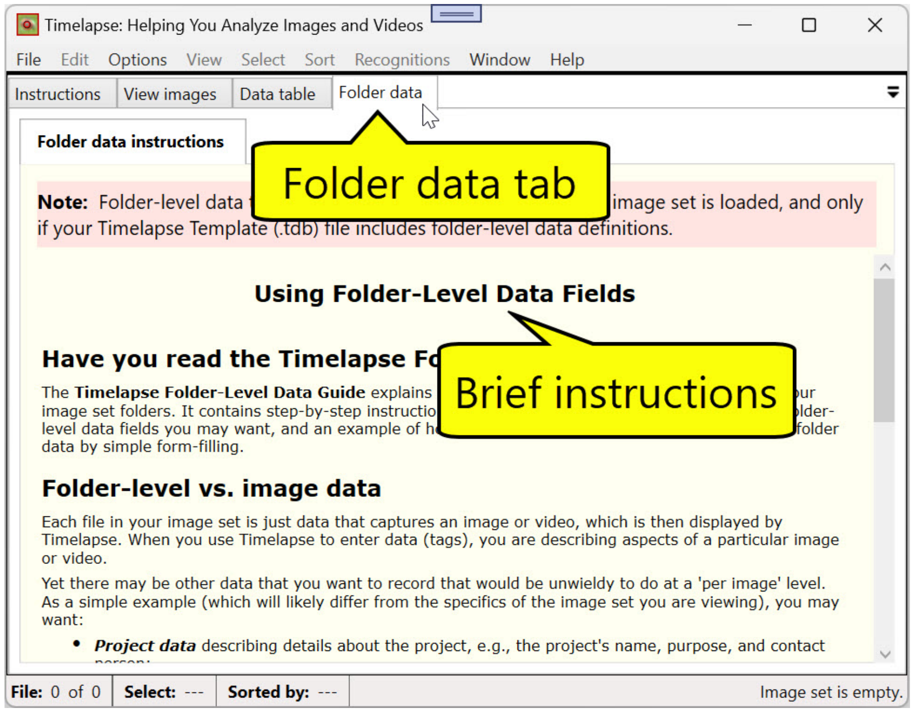
The analyst's perspective: Filling in metadata

This section relies on the Practice image set and the template and folders within it. That template defines an information hierarchy comprising three folder levels as in the previous example: *Project*, *Station* and *Deployment*. It also defines the data fields used by each level. The Timelapse Template Guide describes how a project manager can use the Timelapse Template Editor to define information somewhat similar to this template.

This section illustrates what an analyst would see and do when filling in folder-level metadata for the PracticeImageSet. For the purpose of this tutorial, we won't illustrate entering image-level data. That has already been described in the Timelapse Quickstart Guide. Feel free to fill in the some or all of the image-level data if you haven't already done so.

The Folder Data Tab

Start Timelapse. You will see that it now includes a ***Folder data*** tab. The Metadata tab will initially display an introductory descriptive message, as illustrated below.



Timelapse: Helping You Analyze Images and Videos
File Edit Options View Select Sort Recognitions Window Help
Instructions View images Data table Folder data
Folder data instructions
Folder data tab
Brief instructions

Note: Folder-level data [illegible] image set is loaded, and only if your Timelapse Template (.tdb) file includes folder-level data definitions.

Using Folder-Level Data Fields

Have you read the Timelapse F

The **Timelapse Folder-Level Data Guide** explains [illegible] ur image set folders. It contains step-by-step instructio [illegible] lder-level data fields you may want, and an example of h [illegible] folder data by simple form-filling.

Folder-level vs. image data

Each file in your image set is just data that captures an image or video, which is then displayed by Timelapse. When you use Timelapse to enter data (tags), you are describing aspects of a particular image or video.

Yet there may be other data that you want to record that would be unwieldy to do at a 'per image' level. As a simple example (which will likely differ from the specifics of the image set you are viewing), you may want:

- ***Project data*** describing details about the project, e.g., the project's name, purpose, and contact

File: 0 of 0 | **Select:** --- | **Sorted by:** --- | Image set is empty.

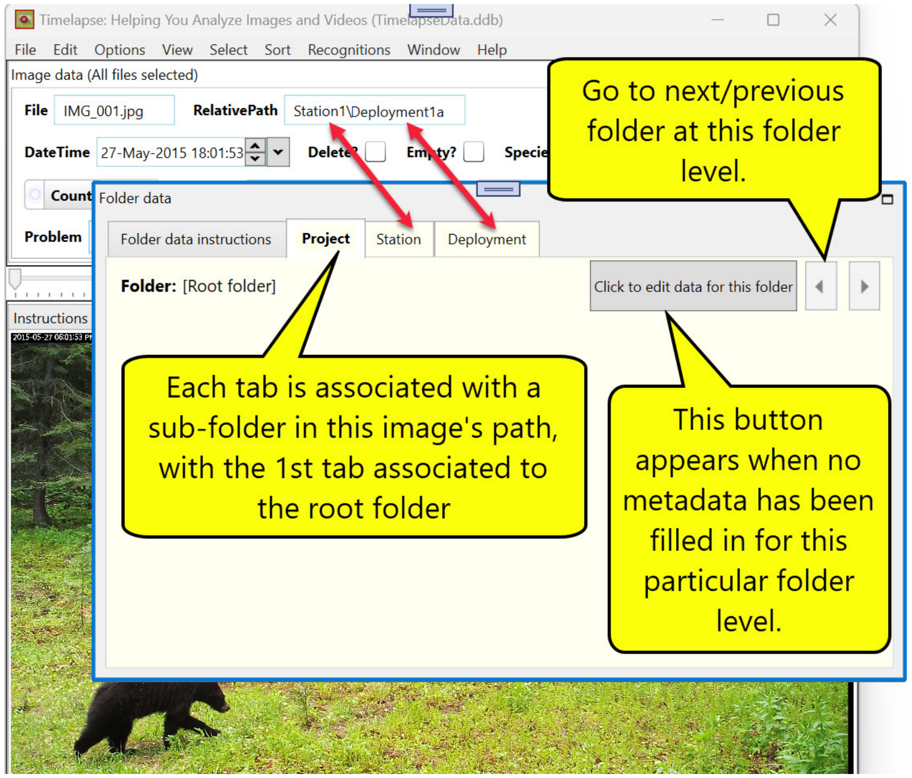
Filling in the folder-level metadata

When your Timelapse Template (.tdb) file defines folder-level metadata, Timelapse will display a folder-level tab for each defined level within its *Folder data* tab. Do the following to see how this works.

1. *Open the PracticeImageSet image set* in Timelapse.
2. *Navigate to your first image: Station1\Deployment1\IMG_001.jpg*.
3. *Select the Folder data tab and pull it out* by dragging its title.

   » A row of tabs are displayed, with the Root folder currently selected. As the current image is located in <root>/*Station1/Deployment1* folder, the level 0 Project tab represents the initial *Root* folder, the level 1 Station tab the *Station1* folder, and the level 2 Deployment tab the *Deployment1* folder.

   » The currently selected root folder (of which there is only one) has, of yet, no filled-in Project data associated with it. Consequently, no fields are displayed.



Timelapse: Helping You Analyze Images and Videos (TimelapseData.ddb)
File Edit Options View Select Sort Recognitions Window Help
Image data (All files selected)
File IMG_001.jpg
RelativePath Station1\Deployment1a
DateTime 27-May-2015 18:01:53
Delete?
Empty?
Specie
Count
Problem
Instructions
Folder data
Folder data instructions
Project
Station
Deployment
Folder: [Root folder]
Click to edit data for this folder
Go to next/previous folder at this folder level.
Each tab is associated with a sub-folder in this image's path, with the 1st tab associated to the root folder
This button appears when no metadata has been filled in for this particular folder level.

4. *Select the 'Click to edit data for this folder' button* for the *Project* tab. When you do, Timelapse will:

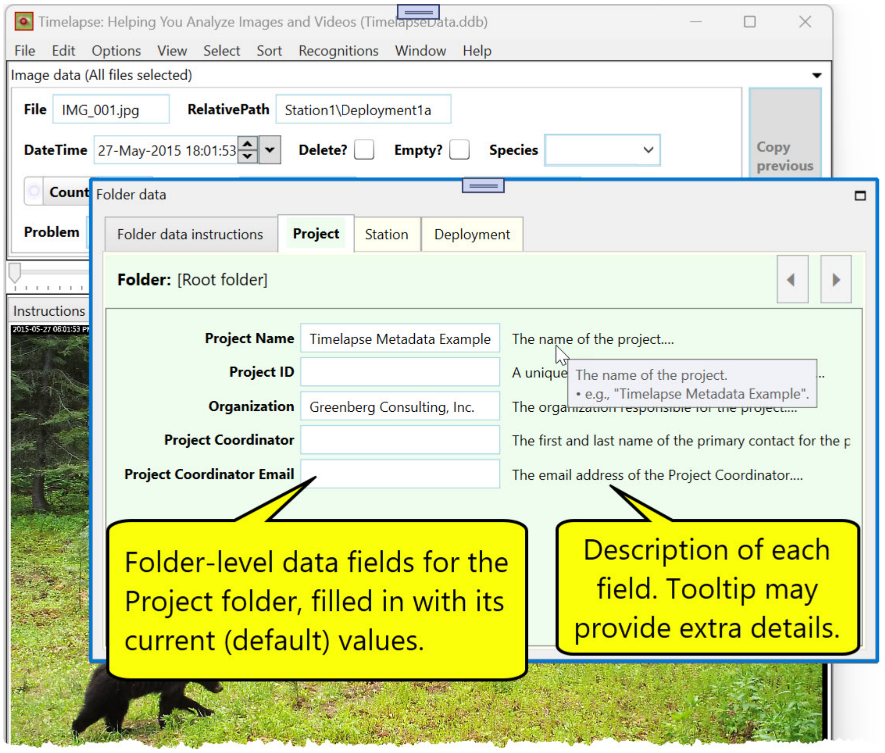
   » initialize and display the folder-level data fields for the *Project* folder (i.e., the root folder).

   » display descriptions for each field, along with a tooltip that may provide even more information,

   » fill in the data fields with default values, if any, as defined in the template.

   The *Project* panel shows the currently selected folder (currently *[Root Folder]*), columns specifying the field name (left side), its current editable value (middle), a description of what the field is for (right side), and a tooltip if the mouse is held atop the description.

   The two buttons on the upper right lets you navigate to the previous or next folder at this folder level, if any. Since the project is at the root folder (i.e., there are no other folders at the root level), these buttons will be disabled.


Timelapse: Helping You Analyze Images and Videos (TimelapseData.ddb)
File Edit Options View Select Sort Recognitions Window Help
Image data (All files selected)
Copy previous
Folder data
Folder data instructions
Project
Station
Deployment
Instructions

File: IMG_001.jpg RelativePath: Station1\Deployment1a
DateTime: 27-May-2015 18:01:53 Delete? ☐ Empty? ☐ Species
Count
Problem

Folder: [Root folder]

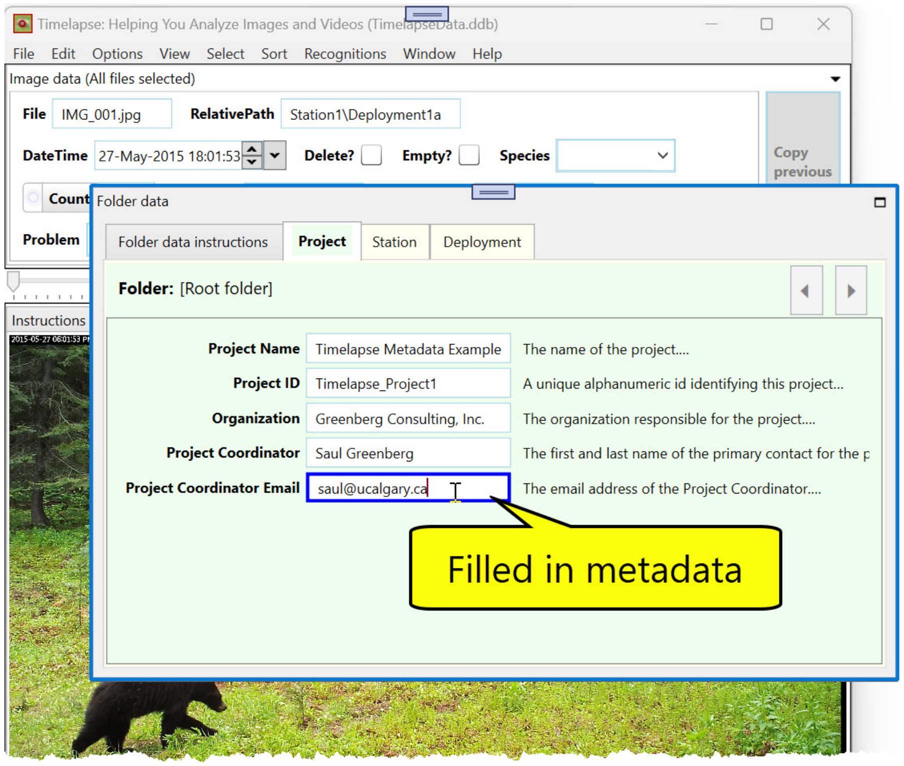
Project Name	Timelapse Metadata Example	The name of the project....
Project ID		A unique
Organization	Greenberg Consulting, Inc.	The organization responsible for the project....
Project Coordinator		The first and last name of the primary contact for the p
Project Coordinator Email		The email address of the Project Coordinator....

The name of the project.
• e.g., "Timelapse Metadata Example".

Folder-level data fields for the Project folder, filled in with its current (default) values.
Description of each field. Tooltip may provide extra details.

5. *Fill in the folder data fields for the Project.*

   » Timelapse will automatically save this data as its fields are edited.


Timelapse: Helping You Analyze Images and Videos (TimelapseData.ddb)
File Edit Options View Select Sort Recognitions Window Help
Image data (All files selected)
Copy previous
Folder data
Folder data instructions
Project
Station
Deployment
Instructions

File: IMG_001.jpg RelativePath: Station1\Deployment1a
DateTime: 27-May-2015 18:01:53 Delete? ☐ Empty? ☐ Species
Count
Problem

Folder: [Root folder]

Project Name	Timelapse Metadata Example	The name of the project....
Project ID	Timelapse_Project1	A unique alphanumeric id identifying this project...
Organization	Greenberg Consulting, Inc.	The organization responsible for the project....
Project Coordinator	Saul Greenberg	The first and last name of the primary contact for the p
Project Coordinator Email	saul@ucalgary.ca	The email address of the Project Coordinator....

6. *Click the Station tab, the 'Click to edit data for this folder' button, and fill it in.*

   » Because we are viewing an image under the <root>*Station1* folder, the saved data is associated and specific to the *Station1* folder.

   » This version of Timelapse contains a set of richer data fields. Examples include:

   ◊ numeric fields that only allow integer and decimal numbers,

   ◊ Single vs multiline text fields for short vs. long text fragments,

   ◊ Choice fields that let you select one, or more than one menu option, etc.

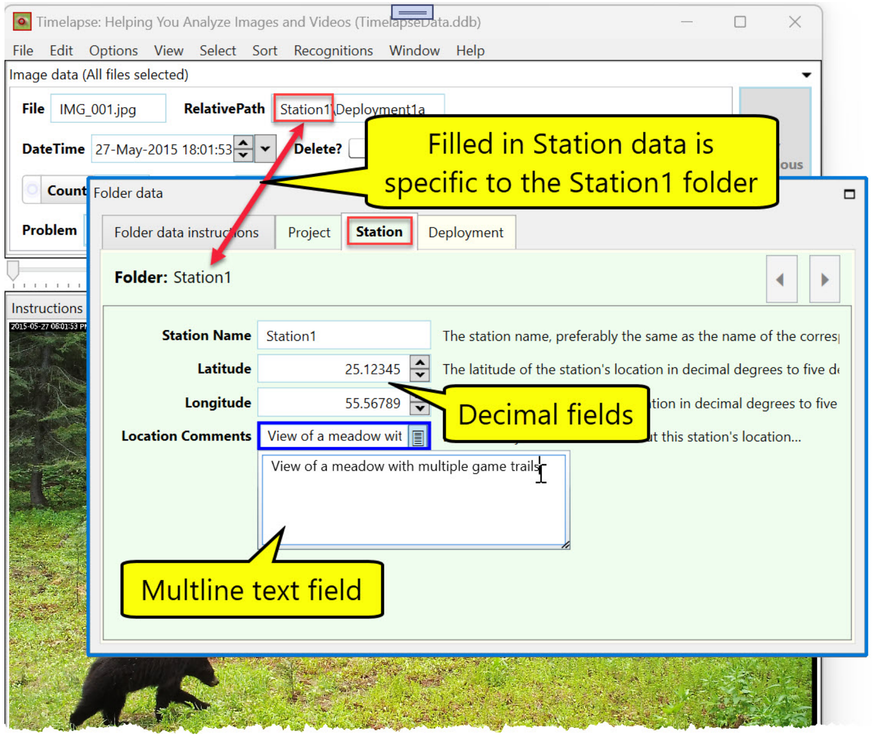

Timelapse: Helping You Analyze Images and Videos (TimelapseData.ddb)
File Edit Options View Select Sort Recognitions Window Help
Filled in Station data is specific to the Station1 folder
Folder data
Folder data instructions
Project
Station
Deployment
Decimal fields
Multline text field

Image data (All files selected)

File IMG_001.jpg RelativePath Station1\Deployment1a

DateTime 27-May-2015 18:01:53 Delete?

Folder: Station1

Station Name	Station1	The station name, preferably the same as the name of the corres
Latitude	25.12345	The latitude of the station's location in decimal degrees to five d
Longitude	55.56789	tion in decimal degrees to five
Location Comments	View of a meadow wit	t this station's location...

View of a meadow with multiple game trails

7. *Repeat the above for the Deployment tab.*

   » Because we are viewing an image under the *<root>\Station1\Deployment1* folder, the saved folder data is associated and specific to the *Station1\Deployment1* folder.

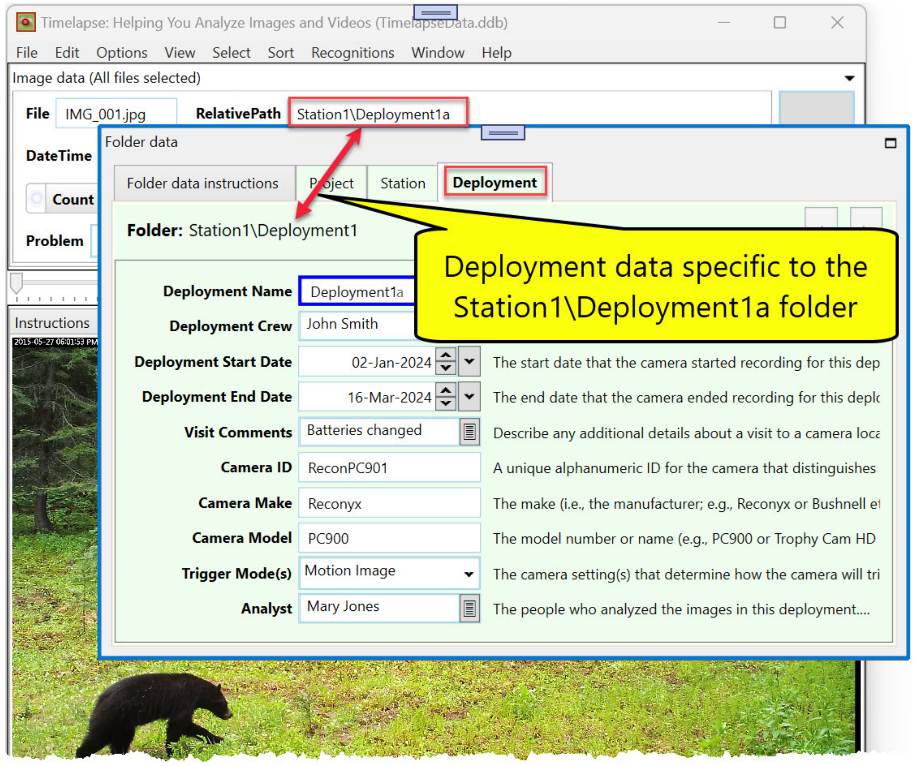

Timelapse: Helping You Analyze Images and Videos (TimelapseData.ddb)
File Edit Options View Select Sort Recognitions Window Help
Folder data
Folder data instructions
Project
Station
Deployment
Deployment data specific to the Station1\Deployment1a folder

Image data (All files selected)

File IMG_001.jpg RelativePath Station1\Deployment1a

Folder: Station1\Deployment1

Deployment Name	Deployment1a	
Deployment Crew	John Smith	
Deployment Start Date	02-Jan-2024	The start date that the camera started recording for this dep
Deployment End Date	16-Mar-2024	The end date that the camera ended recording for this deplo
Visit Comments	Batteries changed	Describe any additional details about a visit to a camera loca
Camera ID	ReconPC901	A unique alphanumeric ID for the camera that distinguishes
Camera Make	Reconyx	The make (i.e., the manufacturer; e.g., Reconyx or Bushnell et
Camera Model	PC900	The model number or name (e.g., PC900 or Trophy Cam HD
Trigger Mode(s)	Motion Image	The camera setting(s) that determine how the camera will tri
Analyst	Mary Jones	The people who analyzed the images in this deployment....

8. *Navigate through your images.* You should notice the following as you navigate through your images.

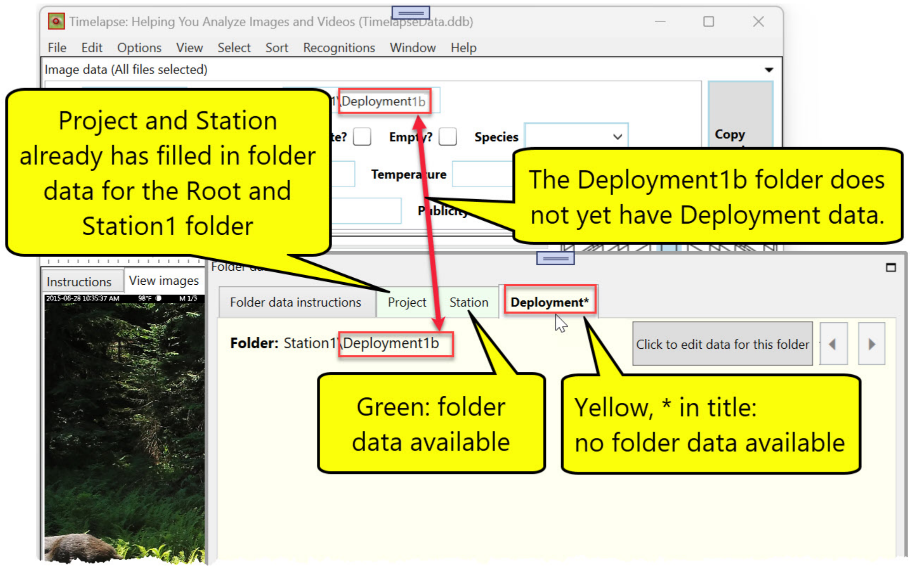
   » Initially, you will be navigating through images contained in the *Root/Station1/Deployment1* folder. As all those images share the same folder data, the metadata display will not change.

   » As soon as you navigate to an image in a different folder (e.g., *Root/Station1/Deployment2*), the display will automatically update as follows.

     » Because that image is still located under the *Root/Station1* folder, its *Project* and *Station* metadata will be the same as the previous images.

     » However, Timelapse will notice that the *Deployment2* folder is different. In this case, there is no Deployment data associated with the *Deployment2* folder. If the *Deployment* tab is selected, you will be able to initialize and fill in its data after selecting the '*Click to edit data for this folder*' button.

     » If you navigate back to an image in the *Deployment1* folder, or use the previous/next buttons at the upper right of the folder's form, the data fields for that folder will be displayed instead.


Timelapse: Helping You Analyze Images and Videos (TimelapseData.ddb)
File Edit Options View Select Sort Recognitions Window Help
Image data (All files selected)
Deployment1b
Empty?
Species
Copy
Temperature
Publicity
Project and Station already has filled in folder data for the Root and Station1 folder
The Deployment1b folder does not yet have Deployment data.
Instructions
View images
Folder data instructions
Project
Station
Deployment*
Folder: Station1\Deployment1b
Click to edit data for this folder
Green: folder data available
Yellow, * in title: no folder data available

Note. Timelapse uses colors to indicate whether a folder has any folder data available.
- *Yellow:* No folder data. The tab contains a *Click to edit data for this folder* button. the tab is selected, and the tab's title contains a * suffix.
- *Green:* Folder data available. The tab displays the folder data fields and its values.
- *Gray:* No metadata fields are defined for this folder level.

9. *Keep on navigating and adding metadata as needed.* Similar to the previous step, Timelapse will recognize and indicate those folders that do not yet have metadata. This will occur when you navigate to images contained in the other Station folders (e.g., Station2- Station4) and the Deployment subfolders contained by them. Fill them all in, as we will later export that data to various CSV files.

10. *Alternate: Using Select.* Instead of navigating through images, you can use the *Select | All files in a folder and its subfolders* menu option to select a particular folder. The folder data tabs for the selected folder will update as described above.

A typical workflow

The expected workflow is simpler than what was described above. Normally, images are added periodically as individual deployments. Using the practice image set as an example, images (and the folders that contain them) would rarely all arrive at the same time.

a. *Project* data would only be filled in once, as there is only a single root folder.
b. *Station data* would usually be filled in during each station's initial deployment.
c. *Deployment data* are usually filled as camera cards are retrieved from the field and images added to the image set as new deployment folders.

This is what this workflow would be in practice.

1. Initially, the image set may be created and its folder data updated when the first deployment is received, for example for th *Station1/Deployment1* folder. The analyst would fill in the metadata for the *Root* folder, the *Station1* folder and the *Deployment1* folder.
2. When a second deployment for *Station1* is received (e.g., images in a new *Deployment2* folder located the *Station1* folder), its images would be added using *File | Add image and video files to this image set...* Only the deployment metadata for the just-added *Deployment2* folder would have to be filled in.
3. Similarly, metadata for new *Station* folders (i.e., new camera stations) would be filled in only if and when they are added to the image set.
4. For each step above, the major effort would be in viewing each image and filling in its image-level data. Relatively speaking, filling in folder-level data is done occasionally and is fairly quick to do.

Exporting metadata

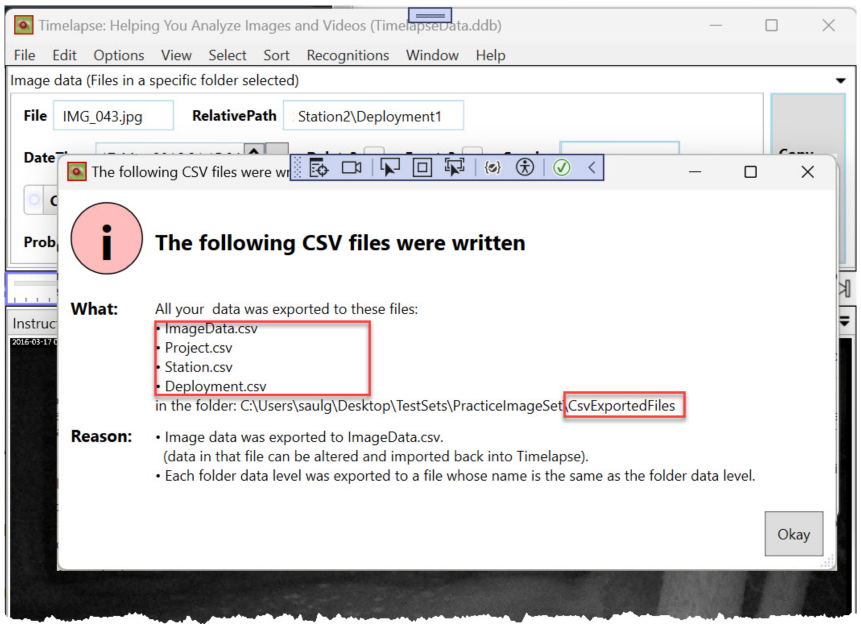
The folder data that you filled in is stored by Timelapse in its internal database. You can export both metadata and image data to *CSV files*, which you can then either import to another database or data analysis system. Timelapse lets you do this by:

File | Export or import data to/from a CSV file |
Export all data (folder data, image/video data) to csv files...

Timelapse will ask you for a location to save these files, and use that to create

- a folder called *CsvExportedFiles* ,
- a csv file for each metadata level and named after that level, where each metadata file collects all metadata for that level as rows and
- an ImageData.csv file for you image data.

The following figure, for example, illustrates the dialog that appears after you export *CSV* files for the practice image set, where the export was done after all the metadata was filled in. As shown below, csv files were created for the image data the *ImageData.csv* and for each level: ***Project.csv***, ***Station.csv***, and ***Deployment.csv***.



Timelapse: Helping You Analyze Images and Videos (TimelapseData.ddb)
File Edit Options View Select Sort Recognitions Window Help
Image data (Files in a specific folder selected)
File IMG_043.jpg
RelativePath Station2\Deployment1
The following CSV files were wr
Okay

The following CSV files were written

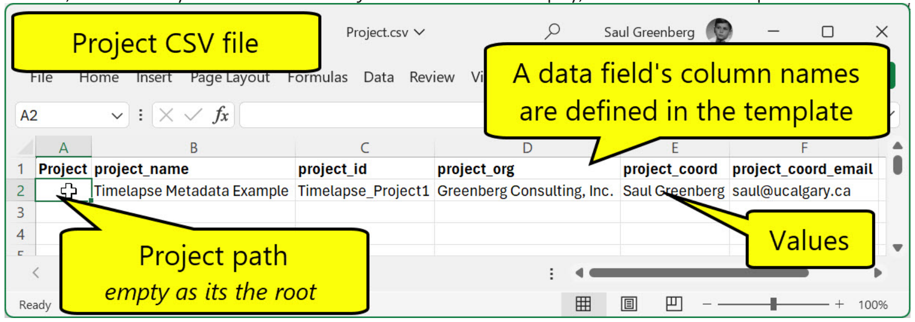
What: All your data was exported to these files:
- ImageData.csv
- Project.csv
- Station.csv
- Deployment.csv

in the folder: C:\Users\saulg\Desktop\TestSets\PracticeImageSet\CsvExportedFiles

Reason:
- Image data was exported to ImageData.csv.
  (data in that file can be altered and imported back into Timelapse).
- Each folder data level was exported to a file whose name is the same as the folder data level.

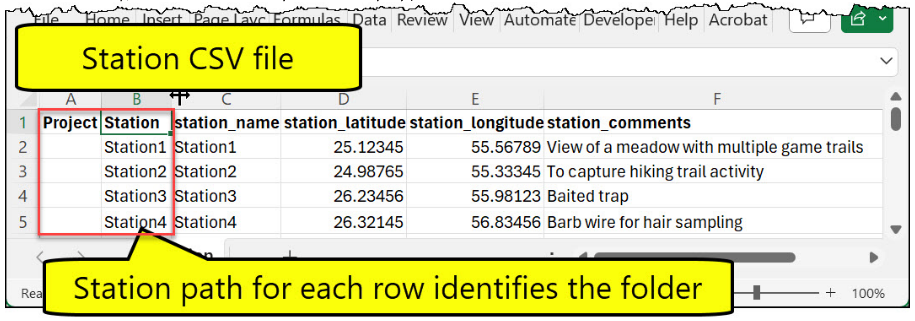
We use Excel to display and explain the *CSV* file's contents, as illustrated below. Each file includes columns at the beginning that identify the folder it belongs to, and to link back to the other *CSV* files to find its 'parent' folder data. For example, Deployment1's folder's path is included in full (e.g., *Station1/Deployment1*). Its parent *Station1* is in the *Station* column, which is a link to the *Station1* row in the *Station.csv* file. Similarly, the parent *Project* is identified by its empty path.

Project CSV file. Because the project level is associated with the single root folder, it has only one row. Its *Project* column is empty, as it is the root path.



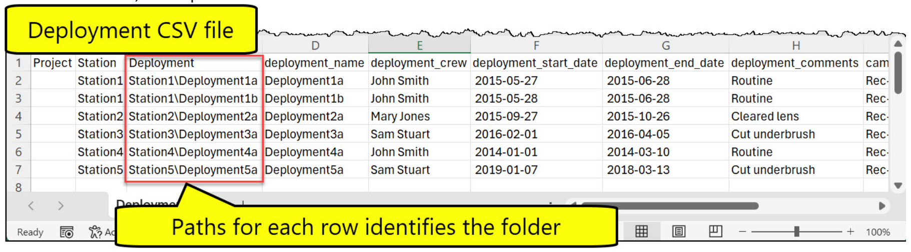
Project	project_name	project_id	project_org	project_coord	project_coord_email
	Timelapse Metadata Example	Timelapse_Project1	Greenberg Consulting, Inc.	Saul Greenberg	saul@ucalgary.ca

Station CSV file. Because the folder data for the 5 *Station* folders were filled in, it has five rows, one for each folder. Its *Station* column provides the path to that folder relative to the root folder. The *Project* column is included for quick lookup (for consistency, as it will always be empty).



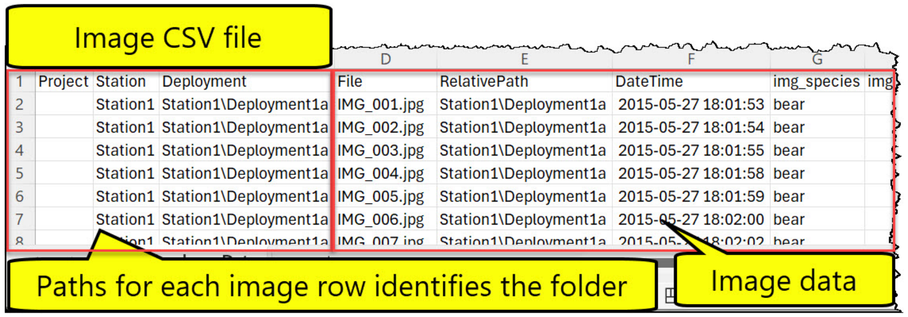
Project	Station	station_name	station_latitude	station_longitude	station_comments
	Station1	Station1	25.12345	55.56789	View of a meadow with multiple game trails
	Station2	Station2	24.98765	55.33345	To capture hiking trail activity
	Station3	Station3	26.23456	55.98123	Baited trap
	Station4	Station4	26.32145	56.83456	Barb wire for hair sampling

Deployment CSV file. Because the folder data for the 6 *Deployment* folders were filled in, it has six rows, one for each folder. Its *Deployment* column provides the path to that folder relative to the root folder. The *Project* and *Station* column is included for quick lookup of its parent paths. Note that while some *Deployment* folder names are identical, the paths discriminate between them.



Project	Station	Deployment	deployment_name	deployment_crew	deployment_start_date	deployment_end_date	deployment_comments	cam
	Station1	Station1\Deployment1a	Deployment1a	John Smith	2015-05-27	2015-06-28	Routine	Rec-
	Station1	Station1\Deployment1b	Deployment1b	John Smith	2015-05-28	2015-06-28	Routine	Rec-
	Station2	Station2\Deployment2a	Deployment2a	Mary Jones	2015-09-27	2015-10-26	Cleared lens	Rec-
	Station3	Station3\Deployment3a	Deployment3a	Sam Stuart	2016-02-01	2016-04-05	Cut underbrush	Rec-
	Station4	Station4\Deployment4a	Deployment4a	John Smith	2014-01-01	2014-03-10	Routine	Rec-
	Station5	Station5\Deployment5a	Deployment5a	Sam Stuart	2019-01-07	2018-03-13	Cut underbrush	Rec-

When image data is exported to do *ImageData.csv* file, links back to the folder data levels are also provided as columns inserted before the image data. In this manner, we can link any image's data to the folder data that it belongs to.



1	Project	Station	Deployment	File	RelativePath	DateTime	img_species	img
2		Station1	Station1\Deployment1a	IMG_001.jpg	Station1\Deployment1a	2015-05-27 18:01:53	bear	
3		Station1	Station1\Deployment1a	IMG_002.jpg	Station1\Deployment1a	2015-05-27 18:01:54	bear	
4		Station1	Station1\Deployment1a	IMG_003.jpg	Station1\Deployment1a	2015-05-27 18:01:55	bear	
5		Station1	Station1\Deployment1a	IMG_004.jpg	Station1\Deployment1a	2015-05-27 18:01:58	bear	
6		Station1	Station1\Deployment1a	IMG_005.jpg	Station1\Deployment1a	2015-05-27 18:01:59	bear	
7		Station1	Station1\Deployment1a	IMG_006.jpg	Station1\Deployment1a	2015-05-27 18:02:00	bear	
8		[illegible]ion1	Station1\Deployment1a	IMG_007.jpg	Station1\Deployment1a	2015-05-[illegible]18:02:02	bear	

To illustrate, look at I*MG_001.jpg* in *ImageData.csv*. We can determine its *Deployment* folder's metadata from the *Deployment* column (*Station1\Deployment1a*). If we go to the *Deployment CSV* file, we can look up its *Station1\Deployment1a* row in its *Deployment* column, and retrieve the metadata values for that folder (e.g. *deployment_crew = JohnSmith, deployment_start_date = 2015-05-27*, etc). Similarly, we can look up this image's *Station1* folder metadata values (*Station1* in the *Station* column), and *Project* (whose path is blank).

On the flip side, we can apply a particular folder's metadata values back to the images within it. For example, the *Project* applies to every image (each image's Project path is blank). The *Station1* metadata value is cross-referenced to the *ImageData.csv* file through that CSV file's *Station* column, where we look up *Station1* to discover which images are within *Station1*. The same goes for the *Deployment1a* folder metadata.[2]

The template editor and folder-level metadata

The folder-level metadata is defined in the TImelapse template *.tdb* file, normally created by a project manager. The template is then made available for others, where it is just copied into the root folder of an image set. Alternately, standardized templates may be made provided for cross-institutional use, where that standard is intended to promote meaningful sharing of ecological data between disparate agencies.

The Timelapse Template Guide is a companion to this document. Its sections detail how to specify folder-level data in your template. Rather than repeat that information here, please see that guide.

Metadata standards

Several metadata standards exist that define the data fields associated with images, and an information hierarchy equivalent to folder-level metadata and the data fields associated with each hierarchical level. Their purpose is to facilitate data-sharing between organizations. The fact that there are several standards, of course, suggests that the problem with standards is that there are so many to choose from!

The Timelapse Template Editor includes a menu item that will list several available standards, and create a template based on the chosen standard:

File|New template based on standard...

Example standards and their details are listed below, including hyperlinks to their source documents. Timelapse currently implements only a subset of these standards as templates, where others will be added over time as warranted. Templates constructed from a standard can be edited if desired, with the caveat that such changes may violate the standard. For example, one can to hide unwanted fields, elaborate tooltips, add extra fields particular to your project, etc.

The Alberta Metadata Standard

The Alberta Metadata Standard was developed by the province of Alberta, Canada. It is based upon, albeit with differences, to British Columbia's Wildlife Camera Metadata Protocol (see citations below).

> "The objective ... is to provide guidance on the types of data that should be collected and reported when using remote cameras (or “wildlife cameras” or “camera traps”) to detect wildlife in Alberta. Consistent collection of remote camera data supports data consolidation and, accordingly, the creation of large spatiotemporal datasets on wildlife distributions across Alberta. This provides opportunities to answer research and monitoring questions within and across jurisdictions, and ultimately at national and global scales."
>
> -- *Wildlife Camera Metadata, Standards for Alberta, 2024*

In Timelapse terms, the standard defines five folder levels: *Project*, *Study Area*, *Survey*, *Sample station / Camera location* and *Deployment*. The final *Image/Sequence* category defines the image observation fields. Images would be located in the lowest *Deployment* level.

This is a fairly elaborate standard with many folder levels and fields. As a good number of the fields are optional, those not necessary for your project can be hidden from the user interface via the template's *Visibility* flag (we recommend this approach over removing those fields altogether).

Currently, the csv files exported by Timelapse are close to, but not exactly identical

2 As a database, each csv file could be recreated as a table. The cross-reference paths linking the csv files could be implemented as *foreign keys* that link the tables and their contents together.

to, the csv files specified by the standard. While some manipulation of the Timelapse csv files' contents would be required to match the Alberta Metadata Standard's csv expectations, this should be straight forward. An upcoming version of Timelapse will include a menu option that outputs csv files to the Alberta Metadata standard.

Wildlife Camera Metadata, Standards for Alberta, 2024 *A standard developed by the province of Alberta, Canada*. Version 3.0. ***(on that web page, select AB Remote Camera Metadata Standard)*** .

Wildlife Camera Metadata Protocol: Standards for Components of British Columbia's Biodiversity No. 44. 2019. *A standard developed by the province of British Columbia, Canada*. Levels include: Project, Study Area, Survey, Deployment, and Image/Sequence.

Camtrap DP

This standard is advocated by agencies promoting free and open access to biodiversity data by collecting datasets from a variety of institutions.

> "Camera Trap Data Package (Camtrap DP) [is] designed to allow users to easily exchange, harmonize and archive camera trap data at local to global scales. Camtrap DP structures camera trap data in a simple yet flexible data model ... that supports a wide range of camera deployment designs, classification techniques (e.g., human and AI, media-based and event-based) and analytical use cases, from compiling species occurrence data through distribution, occupancy and activity modeling to density estimation." *-- Bubnicki et. al, 2023*

In Timelapse terms, the Camtrap DP standard defines two levels. The first is the *DataPackage*, which lists various metadata fields that describe the entire data set. This metadata allows software tools to read and validate the remaining data. The second level is *Deployments*, which includes fields that can be used to either define a camera station's location and other attributes, or to define separate visits to a camera station. The remaining data splits Timelapse's image/video data into two data sets. *Media* includes fields concerning the files themselves (e.g., name, type, timestamp, permissions). *Observations* includes fields concerning observations of the file's contents, such as species seen, count, life state, sex, etc.

Camtrap DP's data package level may seem somewhat onerous, but its use means that software can locate your data and validate it before it is entered into a repository. At all levels, a good number of the fields are optional. Those not necessary for your project can be hidden from the user interface via the template's *Visibility* flag. Fields should not be removed, as missing fields (even if empty) may cause the validator to not accept the data.

While Camtrap DP allows additional fields not defined by the standard, Timelapse does not yet implement that. This will be remedied in upcoming Timelapse versions.

Timelapse *File|Export...* menu has two ways to export Camtrap DP data.

- *Export all data (folder data, image/video data) to csv files...* exports data in the standard Timelapse csv format. Image/video data can be altered and imported back into Timelapse. This csv file does not conform to Camtrap DP specifications, and thus cannot be uploaded to a Camtrap DP repository.
- *Export all data as CamtrapDP files...* exports data in the Camtrap DP format. This includes a mix of json files (for the data package) and Deployment/Observation/Media csv files (for the rest). These files conform to Camtrap DP specifications, where they can be uploaded to a Camtrap DP-capable repository. However, you should use the Timelapse *Import...* menu to import those files back into Timelapse, as they do not conform to what Timelapse expects.

Camtrap DP is currently used by several repositories including: Agouti, Trapper and GBIF (Global Biodiversity Information Facility). As it is favored by other organizations, we should expect more repositories to follow this standard.

Camtrap DP: Data exchange format for camera trap data. This website contains a comprehensive defininition of the Camtrap DP specifications. Timelaps's implementation of Camtrap DP was based upon Version 1 of that specification.

Camtrap DP: an open standard for the FAIR exchange and archiving of camera trap data. Bubnicki et. al, 2023. *Remote Sensing in Ecology and Conservation.* https://doi.org/10.1002/rse2.374. This is the academic paper that initially defined the CamtrapDP standard.

Current limitations. Timelapse needs to improve a few things in its CamtrapDP interface. While *Taxonomic* and *Scientific name* fields both define species of interest, they are not yet linked: both have to be filled in. *Deployment start/ end, exif* and *recognition*-related fields are not yet auto-populated. There is no Timelapse interface for manually creating the bounding boxes suggested by *Bbox* fields.

Other standards

Other standards have either been proposed or are in use by various agencies.

Wildlife insights Minimum Metadata Standards. *A minimal standard for data sharing, as proposed and used by Wildlife Insights. Levels include Project, Camera, Deployment and Image*. See the link on the page above.

An Open standard for Camera Trap Data. Forrester et. al., Biodiversity Data Journal, 2016. *This journal article is one of the first describing the rationale behind an open standard. It also outlines a minimal standard whose levels include*: Project, Deployment, Sequence and Image.

Important considerations

Should you use Folder-levels?

While folder level metadata can add considerable richness to the data you gather, it does incur a cost.

Limitations on changing folder level names and structure. Timelapse is fairly robust in terms of letting you modify your template's data fields, where it can apply those changes to data you had previously filled in with the original template. Unfortunately, it is less tolerant to changes in your folder level specifications i.e., how many levels you have, and what your levels are called. This is because it could lead to incompatibilities between the folder levels defined in your template *vs.* the folder/subfolder structure you created as part of your image set. Having said that, you can still modify the data fields within a folder level.

Timelapse will warn you when it detects folder level incompatibilities, where it will retain the old folder level structure rather than trying to use the one defined in the template (see next section).

Extra housekeeping requirements. While many Timelapse features work as is with folder levels, others require extra housekeeping effort.

- You and your analysts must organize their sub-folders to match the folder-level specifications.
- Images should only be located in the bottom-most sub-folders.
- If using merging, care must be taken to match your child sub-folders with their location in the folder level structure. This is described in more detail below.
- Some Timelapse features will not guard against violating the folder level structure. For example, Timelapse's folder editor may let you edit folder locations and move folders so that they no longer match.

What you should consider. Because of the above, you should carefully consider the following before jumping into using folder-level data.

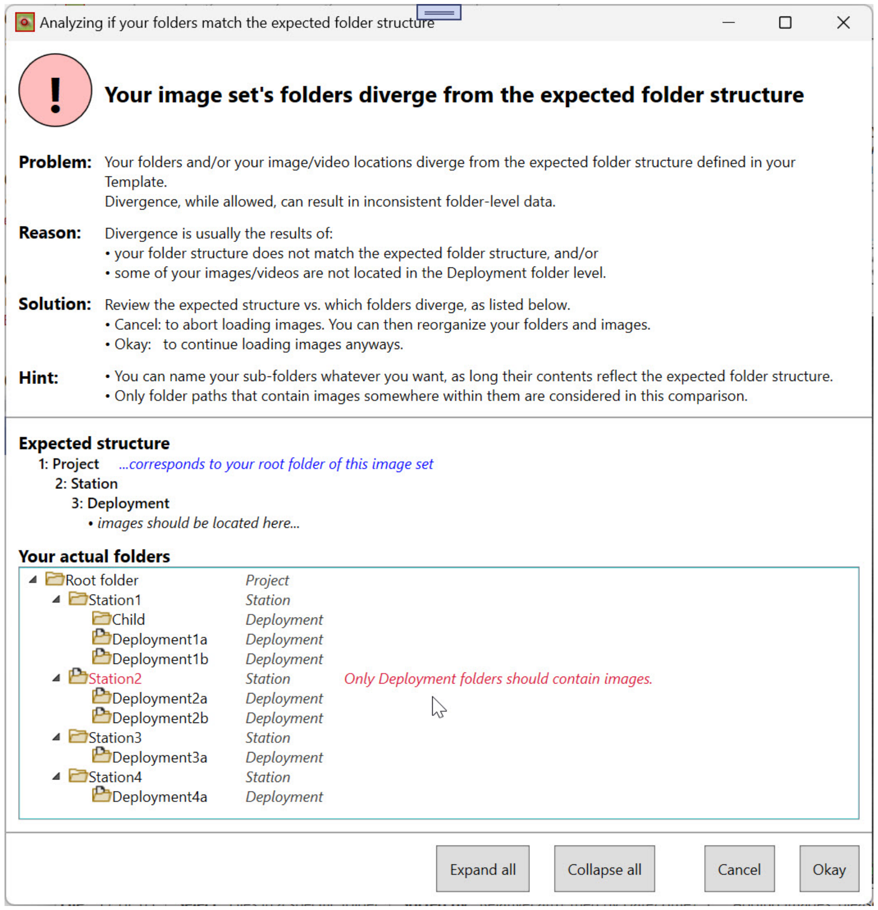
- Think deeply about your folder level hierarchy and their names. Once you start using them, it may be difficult to change them.
- Do you expect future (but unpredictable) changes to your folder structure? If so, you may want to defer using folder levels until you can anticipate what these are and incorporate those into your folder level definitions.
- Are you or your staff willing to do the extra housekeeping, and if so, is the data richness added by folder levels worth the effort?
- Does an existing standard match your needs? Existing standards are excellent for data sharing, but do impose constraints on expected data fields and its structure.

Error detection with metadata

Timelapse does what it can to check for issues between the expected folder level structure as defined in your template vs. the actual folders present.

For example, it includes a new menu option: *Edit|Analyze folder structure*. It displays a dialog showing the expected structure and your actual folder structure. It compares the two to see if your folders and image locations conform to expectations, and warns you when they do not. This check is also performed whenever you load an image set or add new images to an image set, where that dialog is automatically displayed if problems are detected.

In the example below, an analyst incorrectly placed images in the *Station2* folder. Yet images should only be found in the lowest *Deployment* folders. When the analyst loaded the image set, Timelapse automatically raised the following dialog, giving the analyst a chance to abort the load and correct the issue.



Analyzing if your folders match the expected folder structure

Your image set's folders diverge from the expected folder structure

Problem: Your folders and/or your image/video locations diverge from the expected folder structure defined in your Template.
Divergence, while allowed, can result in inconsistent folder-level data.

Reason: Divergence is usually the results of:
• your folder structure does not match the expected folder structure, and/or
• some of your images/videos are not located in the Deployment folder level.

Solution: Review the expected structure vs. which folders diverge, as listed below.
• Cancel: to abort loading images. You can then reorganize your folders and images.
• Okay: to continue loading images anyways.

Hint: • You can name your sub-folders whatever you want, as long their contents reflect the expected folder structure.
• Only folder paths that contain images somewhere within them are considered in this comparison.

Expected structure

1: Project   *...corresponds to your root folder of this image set*
  2: Station
    3: Deployment
      • *images should be located here...*

Your actual folders

```
Root folder           Project
  Station1            Station
    Child             Deployment
    Deployment1a      Deployment
    Deployment1b      Deployment
  Station2            Station      Only Deployment folders should contain images.
    Deployment2a      Deployment
    Deployment2b      Deployment
  Station3            Station
    Deployment3a      Deployment
  Station4            Station
    Deployment4a      Deployment
```

Folder levels can be created without any data fields

The previous examples associate data fields with every hierarchical folder level. However, this is not required: Timelapse associates and displays data fields for a folder level only if data fields are present for that level.

For example, let us say the project manager of the practice image set wanted a three-level folder structure for organizing their image set, but only wanted data fields for the *Project* and *Deployment* levels, i.e., no *Station* metadata was required. While the manager would still have to create and name three levels in the template editor, she could leave the *Station* level empty.

When an analyst later views the folder data tab in Timelapse, the *Station* tab would be present but grayed out. When the folder level data is exported as *csv* files, the *Station* file would still be created, but without any rows in it.

Merging considerations when using folder levels

Merging with levels requires somewhat more discipline, as the folder levels differ between parent and child databases. This is best illustrated by example using our practice image set structure, where the parent database would contain the three *Project/Station/Deployment* level as defined in its template.

Consider the case where a project manager wanted her field people to analyze deployments only, e.g., after collecting images in a *Deployment* folder from the field and analyzing them. Copying the template from the parent database would not work, as it defines three levels vs. the single *Deployment* level used by the analyst. Timelapse would produce an error message saying the folder structure was not compatible with it.

1. The first (and best) remedy to this would be for the project manager to:
   - » create an empty deployment folder under one of the station folders,
   - » select the *File|Merge databases|Checkout (copy) a sub-folder database from the master database*
   - » give that deployment folder to the field person. They can then use that template as normal, where they would use the *File|Add images...* option to add their images to the empty database. They could then analyze the images as normal, fill in the deployment information, and return it to the project manager.
   - » The project manager would then copy that folder into a deployment level, and merge it back into the parent.
2. Another approach has the project manager creating a copy of the master template, and then editing the template to remove the *Project* and *Station* levels. That template can then be handed off to the field person, who would have to ensure that their folder structure matched the levels in the edited template.

   When done, the field person would return their folder back to the project manager who would locate and merge it back in as described above.

Avoiding folder-level metadata conflicts

In summary, working with folder levels adds constraints into what Timelapse expects and how well Timelapse can detect and/or fix problems. While some of these constraints may be relaxed in future Timelapse releases, it is important to know what they are, as otherwise it can lead to issues.

1. ***Modifying your template.*** When a template is used to load an image set, it structures the image set data in a certain way. If the template is modified and that image set loaded, Timelapse will compare the two, and will update the data only for certain modifications:
   - » Ok: Adding, deleting or changing attributes of a data fields EXCEPT for its type.
   - » Problematic: adding or deleting levels, and sometimes changing the name of a level.
2. ***Using the folder editor.*** The Timelapse folder editor allows you to create and move folders around. Problems include:
   - » changing the folder structure so it no longer mirrors the expected levels defined in the template
   - » renaming or moving folders, as this can break the link to previously entered folder metadata {this will be fixed in future releases]
3. ***Searching for missing files or folders.*** TImelapse has various facilities for searching for missing files or folders. If those missing folders had metadata associated with it, they would likely no longer match the links to the found folders. (This will be remedied in future releases with warning dialogs) .
4. ***If in doubt about the effect of an operation, make your own backups of your .ddb and .tdb files just in case.*** Its better to be safe rather than sorry. While Timelapse does create occassional backups in the Backup folder, your own backups are easier to manage as you will know when you created them.