CatAmount User Guide

Michael Rickard
http://www.kasploosh.com/

February 6, 2019

Contents

Part I

User Guide

Chapter 1

Introduction

CatAmount is software for analyzing GPS data from wildlife collars and finding relationships, especially time/space relationships. This is a User Guide for that software.

Chapter 2

Overview

CatAmount is made of several parts that work together to make a functioning whole. This chapter explains what all the major functional parts are, as a way of providing an overview of the software.

The basic model of the software is to have each major function accessible by a full command line interface. Then there is a GUI layer that communicates with the major functions using their command line interfaces. Blocks of common code that are needed by several functions are stored separately. A configuration file stores settings between sessions.

There are two basic ways to interact with the related programs. One is via the command line, and the other is via the GUI. Which you pick depends on which you prefer. The GUI is perhaps easier to use, while the command line applications can be scripted or automated.

There are multiple ways to configure settings for the functions. There is a configuration file which sets your defaults for each function. When using the command line, there are command line arguments to change each setting. When using the GUI, there are buttons and widgets to change each setting.

Remember that your configuration file sets your defaults. And settings made via the command line or the GUI override your configuration file settings.

2.1 Components

This section will provide an outline of the major functional components, and a brief reminder of what they do. Each of these components is described in more detail in a subsequent section.

1. Configuration File. A plain text configuration file that enables you to save common settings between sessions.

2. Data File. A comma-separated data file containing the GPS data you wish to analyze.

3. Output Directory. A scratch directory that CatAmount can use to write out image files.

4. Project. A simple python "project" that is used to make the code modular and shareable.

5. Find Clusters. A major function of the software: Analyzes GPS data to find one cat's time/space groupings, called clusters.

6. Show Territories. A major function of the software: Processes GPS data to create color-coded depictions of one or more cat's territory.

7. Find Crossings. A major function of the software: Analyzes GPS data to find instances where two or more cats shared the same place in time/space, called crossings.

8. Find Whodunit. A major function of the software: Search through GPS data to find which cat was at a certain time and location.

9. Match Survey To Cluster. A major function of the software. Attempt to find matches between a list of field-verified site surveys, and a list of GPS clusters.

10. GUI. A Tkinter GUI which allows easy operation of all major functions, and easy manipulation of settings and output.

11. Font. These are fonts that are needed by the image-creating functions.

2.2 Setting Up The Software

I have made a guide for setting up the software for each of three major platforms:

- Windows Guide
- Mac OS X Guide
- Linux Guide

Chapter 3

Configuration File

The main role of the configuration file is to store common settings between sessions of using the program. Without the configuration file, we would have to put default settings in the source code where it is harder to access. In short, the configuration file puts you in charge of the default settings for the program.

The file is very easy to understand. The reference section contains an example configuration file.

The file is divided into sections. The first section holds "Global Settings" which apply to every part of the software. The remaining sections hold settings for different major functions. The available directives are described in the chapter devoted to each major function.

Every valid configuration directive is shown in the sample configuration file.

The configuration file is never written to by the software. To change the configuration file, you need to edit it with a text editor. You are in control of this file, and it is here for your use.

3.1 Configuration File Files

The software looks for the configuration in only one place, the base directory of the software, and under only one name `catamount.conf`. You should find the configuration file here:

```
catamount/catamount.conf
```

The CatAmount source packages do not ship a configuration file, because a hasty unzipping of the source could overwrite your carefully crafted file. Instead the CatAmount source package ships a default file, at this location:

```
catamount/catamount.default.conf
```

If this is your first time using catamount, you should rename that file to `catamount.conf` and begin editing it to your taste.

Chapter 4

Data File

The data file is an important component of the system, and it is the user who supplies this component. It should be a CSV file in a certain format, and it should contain all the GPS data that needs to be analyzed at a particular time.

All of the other components will refer to this file to get the data they need to function. The program will never alter or write to your data file. It is a good idea to mark your data as "read only" to prevent accidental alterations.

The data file should be large enough to contain all the points you wish to study at a single time. If the file becomes very large, the software might take longer to do its work.

The data file is always changing, and is not a file that is part of the software. You should create your own data file(s) in the appropriate format, containing the GPS data you wish to study. Such files can be anywhere on the file system, and you can change the data file you are using at any time.

The format of the data file is described in detail here.

4.1 Data File Files

A sample data file may have been provided with the software, in order to demonstrate the software's function right out of the box, and to demonstrate the format of the file. That sample file may be located here:

```
catamount/data/ALLGPS.csv
```

4.2 Data File Settings

The path to the data file is a "global setting" because all of the different major functions need data to examine, in order to do their job. The data file can be set on the command line, or from the GUI.

Data file settings are expected in the following format:

datafile_path PATH Select a data file to interpret. `PATH` should be a path to a file, like `C:\joe\files\cougars.csv` or `/Users/joe/Desktop/leopards.csv`.

The argument is of type PATH

4.3 Data File @ Configuration File

The data file is represented in the configuration file by the following option:

```
[Global_Settings]
datafile_path = C:\you\decide\data.csv
```

Change that setting to the full path of the data file you most commonly use.

Chapter 5

Output Directory

Each of the major functions is able to create image feedback to verify that the software is functioning properly. The image files are written to a directory, so they can be shared between different parts of the application. For this reason, there always needs to be an output directory that the software can write to.

This output directory also serves as the starting point when choosing to save some text or images to a new file.

- You can set the output directory in the configuration file, on the command line, or using the GUI.
- If you are working on a project that involves assembling a large collection of images or text, you may wish to set the output directory to a special folder for that project.
- You can set the output directory to a temp directory that is emptied every day, if that is something you think about.

5.1 Output Directory Files

A sample output directory has been created for you, so that the software will function right out of the box. That sample directory was created right in the root folder:

```
catamount/output/
```

5.2 Output Directory Settings

The path to the output directory is a "global setting" because all of the different major functions need some disk space to write to, in order to do their job. The output directory can be set on the command line, or from the GUI.

Output directory settings are expected in the following format:

outdir_path PATH Select an output directory where the software can make some files, and that serves as the base directory for "Save As" operations. `PATH` should be a path to a directory, like `C:\jane\stuff\output` or `/Users/joe/Desktop/output/`.

The argument is of type PATH

5.3 Output Directory @ Configuration File

The output directory is represented in the configuration file by the following option:

```
[Global_Settings]
outdir_path = C:\path\to\catamount\output
```

Change that setting to a full path where the software can write out images.

Chapter 6

Project

A Python "project" is a way of organizing code so that it is modular and shareable. The CatAmount project contains a number of modules that are neatly divided according to function. Each module contains a handful of classes, for the most part.

Most modules in the project are dedicated to a particular major function of the software. The module `common.py` has code that is common to all the different functions. There are also other special purpose modules included here.

6.1 Project Files

The catamount project is represented by this directory and all of its files:

```
catamount/catamount/*
```

Chapter 7

Find Clusters

The goal of Find Clusters is to identify instances where one cat used the same location over a period of time. These instances are called "clusters". A cluster is one kind of relationship that can be found in GPS data. Clusters can represent an event such as mating, preying, rearing or something else.

Finding clusters means essentially finding GPS points that are close to each other in distance, and time. The `radius` setting controls the distance, and the `time_cutoff` setting controls the time.

There is a full command-line interface to Find Clusters, and a GUI interface. Note that the GUI works by summoning the command-line interface. Use whichever interface is appropriate for your project.

7.1 Find Clusters Files

Find Clusters is represented in the files by the following file:

```
catamount/find_clusters.py
```

7.2 Find Clusters Settings

The following settings apply to the Find Clusters function no matter where it is used from. These are the "essential" settings which are used in different ways in the configuration file, the command-line interface, and the GUI.

catid ID Show clusters for a cat with this `ID`. This is mandatory, since a cluster, by definition, only involves one cat, but a data file may have data for many cats.

The argument is of type ID.

radius INT Design radius of a cluster, in meters. Defines how far apart can the points be, and still be considered a cluster.

The argument is of type INT.

time_cutoff INT Design time cutoff of a cluster, in hours. After this amount of time, if a cat comes back to the same location, it's no longer considered the same cluster.

The argument is of type INT.

minimum_count INT Minimum number of points to qualify as a cluster. This is useful for weeding out clusters that only have two points. "Only show me clusters containing this many points, or more."

The argument is of type INT.

minimum_stay INT Minimum elapsed time of clusters, in hours. This is useful for separating major events from minor ones. "Only show me clusters which lasted for this number of hours."

The argument is of type INT.

start_date DATE This limits the **data set** to points happening after this start date. The limiting happens before any attempts to find clusters.

The argument is of type DATE.

`start_date` and `end_date` can be used independently of each other.

end_date DATE This limits the **data set** to points happening before this end date. The limiting happens before any attempts to find clusters.

The argument is of type DATE.

`start_date` and `end_date` can be used independently of each other.

text_style CHOICE The text that is produced can be in one of several formats.

The argument is of type CHOICE. The text style `CHOICE` should be one of these:

- `csv` — machine readable, creates a table describing each cluster
- `csv-all` — machine readable, adds a second table showing all points
- `descriptive` — human readable, describes each cluster
- `descriptive-all` — human readable, also shows every point in the cluster

The default text style is currently `csv`.

clusterid ID Zoom in on a specific cluster. `ID` should be in the same format that this script produces. Typically you would run the script once, see an interesting cluster, and copy its ID. Then run the script again, adding an argument to zoom in on that cluster ID.

The argument is of type ID.

7.3 CLI Find Clusters

The command-line interface gives easy access to the Find Clusters function from the command-line, and therefore from scripts.

This is how you get started with the command-line interface:

```
cd C:\path\to\catamount
find_clusters.py --help
```

Here are the settings for the command-line, along with links to more information for each setting.

Long Form	Short Form	Reference
`--help`	`-h`	Print usage info and exit.
`--datafile_path PATH`	`-f PATH`	datafile_path
`--outdir_path PATH`	`-o PATH`	outdir_path
`--catid CATID`	`-c CATID`	catid
`--radius INT`	`-r INT`	radius
`--time_cutoff INT`	`-t INT`	time_cutoff
`--minimum_count INT`	`-mc INT`	minimum_count
`--minimum_stay INT`	`-ms INT`	minimum_stay
`--start_date DATE`	`-d1 DATE`	start_date
`--end_date DATE`	`-d2 DATE`	end_date
`--text_style STYLE`	`-x STYLE`	text_style
`--clusterid CLUSTERID`	`-z CLUSTERID`	cluster_id

7.4 Find Clusters @ Configuration File

Here is an example of how the Find Cluster settings are specified in the configuration file.

```
[Cluster_Settings]
radius = 200
time_cutoff = 144
start_date = 0
end_date = 0
minimum_count = 0
minimum_stay = 0
```

Chapter 8

Show Territories

The goal of Show Territories is to simply make a color coded representation that shows where one cat is in relation to others. Its basic function is to create a rough graphic showing a territory. This is useful for visualizing the data set you are working with.

There are settings to limit the display to certain cats, and to certain dates. You can use these settings to limit the data set, and get immediate visual feedback about the settings.

There is a full command-line interface to Show Territories, and a GUI interface. The GUI works by interacting with the command-line interface. Use whichever interface is appropriate for your project.

8.1 Show Territories Files

Show Territories is represented in the files by the following file:

```
catamount/show_territories.py
```

8.2 Show Territory Settings

The following settings apply to the Show Territories function no matter where it is used from. These are the "essential" settings which are used in different ways in the configuration file, the command-line interface, and the GUI.

catids LIST Show territories for cats with the given cat IDs. This is helpful to limit the display of territories to a few cats you are interested in.

The argument is of type LIST.

When using the command-line interface, `LIST` should be a comma-separated list of cat IDs, as in this example:

```
show_territories.py --catids M001,F002,M003,F004
```

For the GUI, you can select one or many cats from the menu.

dot_size INT Since Show Territories is primarily a graphic function there are a few settings to fine tune the graphic output. Dot size is the size in pixels of each cat's points in the graphic output.

The argument is of type INT.

perimeter_resolution INT This changes how the software draws the boundary around the outside of a cat's territory.

The argument is of type INT.

The software uses a radial concept, and this argument specifies the number of degrees to lump together when choosing points to use in the border.

```
360 degrees / X resolution = Y border points
360 degrees / 1 resolution = 360 border points
360 degrees / 9 resolution = 40 border points
360 degrees / 120 resolution = 3 border points
```

A setting of 1 will be incredibly spiky, and a setting of 120 will be a triangle. I have found that 8 to 10 is a good setting. If you wind up with an ugly boundary around a certain cat, you might try changing this parameter.

start_date DATE This limits the **data set** to points happening after this start date. The limiting happens before any attempts to show territories.

The argument is of type DATE.

`start_date` and `end_date` can be used independently of each other.

end_date DATE This limits the **data set** to points happening before this end date. The limiting happens before any attempts to show territories.

The argument is of type DATE.

`start_date` and `end_date` can be used independently of each other.

8.3 CLI Show Territories

The command-line interface gives easy access to the Show Territories function from the command-line, and therefore from scripts.

This is how you get started with the command-line interface:

```
cd C:\path\to\catamount
show_territories.py --help
```

Here are the settings for the command-line, along with links to more information for each setting.

Long Form	Short Form	Reference
`--help`	`-h`	Print usage info and exit.
`--datafile_path PATH`	`-f PATH`	datafile_path
`--outdir_path PATH`	`-o PATH`	outdir_path
`--catids CATIDS`	`-c CATIDS`	catids
`--dot_size INT`	`-s INT`	dot_size
`--perimeter_resolution INT`	`-r INT`	perimeter_resolution
`--start_date DATE`	`-d1 DATE`	start_date
`--end_date DATE`	`-d2 DATE`	end_date

8.4 Show Territories @ Configuration File

Here is an example of how the Show Territories settings are specified in the configuration file.

```
[Territory_Settings]
dot_size = 4
perimeter_resolution = 9
start_date = 0
end_date = 0
```

Chapter 9

Find Crossings

The goal of Find Crossings is to identify instances where one cat was near another in time and space. These instances, called "crossings", are another kind of relationship that can be pulled from GPS data. Crossings might be useful for study because they might indicate mating, cooperation, competition, or something else.

To find crossings, we identify points that are close in space, and also close in time, where more than one cat was involved. The `radius` settings defines the distance, and the `time_cutoff` defines the time. A crossing is very much like a cluster that involves more than one cat.

There is a full command-line interface to Find Crossings, and a GUI interface. Note that the GUI works by interacting with the command-line interface. Use whichever interface is appropriate for your project.

9.1 Find Crossings Files

Find Crossings is represented in the files by the following file:

```
catamount/find_crossings.py
```

9.2 Find Crossings Settings

The following settings apply to the Find Crossings function no matter where it is used from. These are the "essential" settings which are used in different ways in the configuration file, the command-line interface, and the GUI.

catids LIST Limit the search for crossings to cats having these IDs. This is useful if you want to more closely examine the relationship between several cats.

The argument is of type LIST.

When using the command-line interface, `LIST` should be a comma separated list of cat IDs, as in this example:

```
find_crossings.py --catids M001,F002,M003,F004
```

In the GUI, you just select several cat IDs using the mouse.

radius INT Design radius of a crossing, in meters. Defines how far apart can the points be, and still be considered a crossing.

The argument is of type INT.

time_cutoff INT Design time cutoff of a crossing, in hours. After this amount of time, if two cats visit the same location, it's no longer considered a crossing.

start_date DATE This limits the **data set** to points happening after this start date. The limiting happens before any attempts to find crossings.

The argument is of type DATE.

`start_date` and `end_date` can be used independently of each other.

end_date DATE This limits the **data set** to points happening before this end date. The limiting happens before any attempts to find crossings.

The argument is of type DATE.

`start_date` and `end_date` can be used independently of each other.

text_style CHOICE The text that is produced can be in one of several formats.

The argument is of type CHOICE. The text style `CHOICE` should be one of these:

- `csv` — machine readable, creates a table describing each crossing
- `csv-all` — machine readable, adds a second table showing all points
- `descriptive` — human readable, describes each crossing
- `descriptive-all` — human readable, also shows every point in the crossing

The default text style is currently `csv`.

crossingid ID Zoom in on a specific crossing. `ID` should be the same kind of ID that this script produces. Typically you would run the script once, learn about an interesting crossing, and copy the crossing ID. Then run the script again, adding an argument to zoom in on that crossing ID.

The argument is of type ID.

9.3 CLI Find Crossings

The command-line interface gives easy access to all the settings from the command line, and therefore from scripts. Here is how you get started with the command-line interface:

```
cd C:\path\to\catamount
find_crossings.py --help
```

Here are the settings for the command-line, along with links to more information about each setting.

Long Form	Short Form	Reference
`--help`	`-h`	Print usage info and exit
`--datafile_path PATH`	`-f PATH`	datafile_path
`--outdir_path PATH`	`-o PATH`	outdir_path
`--catids CATIDS`	`-c CATIDS`	catids
`--radius INT`	`-r INT`	radius
`--time_cutoff INT`	`-t INT`	time_cutoff
`--start_date DATE`	`-d1 DATE`	start_date
`--end_date DATE`	`-d2 DATE`	end_date
`--text_style STYLE`	`-x STYLE`	text_style
`--crossingid CROSSINGID`	`-z CROSSINGID`	crossingid

9.4 Find Crossings @ Configuration File

Here is an example of how the Find Crossing settings are specified in the configuration file.

```
[Crossing_Settings]
radius = 200
time_cutoff = 144
start_date = 0
end_date = 0
```

Chapter 10

Find Whodunit

Find Whodunit adds an important kind of searching to the project. Most of the project is devoted to telling you about a cat's location and history. But sometimes you will know the location and time, but not the cat, and that is where Find Whodunit helps. With Find Whodunit, you request a certain time and location, and it searches for any cats that were nearby.

You start with a date and location, which are required. The software then searches through the available data to find GPS points that match.

There is a full command-line interface to Find Whodunit, and a GUI interface. Note that the GUI works by interacting with the command-line interface. Use whichever interface is appropriate for your project.

10.0.1 Some Whodunit Internals

One approach to this function might be to compare **every** point in the data set against the query. Find Whodunit, as currently implemented, does not take this approach.

The first thing Find Whodunit does is prune the whole data set to roughly the correct time and location. For example, we take a multiple of the design radius, and only keep points that are within that distance. And we take a multiple of the time cutoff, and only keep points that are within that time frame.

Now we have a smaller set of data that is more likely to be correct. We use that smaller set to look for matches, calculate the relative closeness of every point, put points in order by closeness, and display the results.

10.1 Find Whodunit Files

Find Whodunit is represented in the files by the following file:

```
catamount/find_whodunit.py
```

10.2 Find Whodunit Settings

The following settings apply to the Find Whodunit function no matter where it is used from. These are the "essential" settings which are used in different ways in the configuration file, the command-line interface, and the GUI.

radius INT Design radius of a match, in meters. Only points that are within this distance from the request location will be considered a true match, and displayed as a match.

The argument is of type INT.

time_cutoff INT Design time cutoff of a match, in hours. Only points that are within this time frame from the request time will be considered a true match, and displayed as a match.

The argument is of type INT.

date DATE This is the request date that is the basis for the search. This should be a single point on a hypothetical timeline; by widening or narrowing the `time_cutoff` you can enlarge or constrict the search period.

The argument is of type DATE.

x_coordinate INT This is the X coordinate of the requested location, in NAD27 format. This is also known as the "easting" or "east". This should work with `y_coordinate` to specify a single point in space; by enlarging or shrinking the `radius` you can expand or contract the search area.

The argument is of type INT.

y_coordinate INT This is the Y coordinate of the requested location, in NAD27 format. This is also known as the "northing" or "north". This should work with `x_coordinate` to specify a single point in space; by enlarging or shrinking the `radius` you can expand or contract the search area.

The argument is of type INT.

text_style CHOICE The text that is produced can be in one of several formats.

The argument is of type CHOICE. The text style `CHOICE` should be one of these:

- `csv` — macine readable, table of points that match or are close
- `descriptive` — human readable, description of points that match or are close

The default text style is currently `csv`.

10.3 CLI Find Whodunit

The command-line interface gives easy access to all the settings from the command line, and therefore from scripts. Here is how you get started with the command-line interface:

```
cd C:\path\to\catamount
find_whodunit.py --help
```

Here are the settings for the command-line, along with links to more information about each setting.

Long Form	Short Form	Reference
`--help`	`-h`	Print usage info and exit
`--datafile_path PATH`	`-f PATH`	datafile_path
`--outdir_path PATH`	`-o PATH`	outdir_path
`--radius INT`	`-r INT`	radius
`--time_cutoff INT`	`-t INT`	time_cutoff
`--date DATE`	`-d DATE`	date
`--x_coordinate INT`	`-cx INT`	x_coordinate
`--y_coordinate INT`	`-cy INT`	y_coordinate
`--text_style STYLE`	`-x STYLE`	text_style

10.4 Find Whodunit @ Configuration File

Here is an example of how the Find Whodunit settings are specified in the configuration file.

```
[Whodunit_Settings]
radius = 200
time_cutoff = 144
```

Chapter 11

Match Survey To Cluster

Note: Match Survey To Cluster was added at the request of the Teton Cougar Project (Wyoming, USA) in 2012. It may not be helpful to all teams.

Imagine you have two separate sets of data that both contain location and time. For example, you may have a large amount of GPS collar data in one file, and then a large amount of site visit information in another file.

What this function does is for every site visit in the site visit file, it tries to find a matching cluster in the GPS collar data.

The radius and time cutoff that determine what constitutes a "match" are configurable by the researcher. The top three potential matches are returned.

This was built to search through a survey file, and to find matches in the GPS collar data file. But it could potentially find matches in any two sets of data that are comparable.

11.1 Survey File Format

The survey data file should contain comma-separated text, where each line of text represents one site survey. The software takes each line, averages all the UTM coordinates, and averages all the date information. It then searches for matching clusters based on those averages.

The software is able to parse the data in the CSV file only if it knows which columns contain coordinates, and which contain dates. Near the top of the file `match_survey_to_cluster.py` there are a few simple Python lists. In these lists you specify each of the columns that should be considered coordinates or dates. There are further instructions on editing these lists in the source code where the editing happens.

In summary, the software can process the CSV data only if it knows the name of each CSV column where it can find the data it needs.

11.2 Match Survey To Cluster Files

Match Survey To Cluster is represented in the files by the following file:

```
catamount/match_survey_to_cluster.py
```

11.3 Match Survey To Cluster Settings

The following settings apply to the Match Survey To Cluster function no matter where it is used from. These are the "essential" settings which are used in different ways in the configuration file, the command-line interface, and the GUI.

survey_file_path This is the path to the file containing the survey or field work information. Remember that this function takes a list of site surveys, and then searches through the GPS data looking for matching clusters. There must be a file containing site survey data or no search can be performed.

The argument is of type PATH.

radius This is the design radius of a match, in meters. It defines how far apart a site survey can be from a cluster, and still be considered a match. Setting this value low will result in too few matches, and setting it high will result in too many matches.

The argument is of type INT.

time_cutoff This is the design time cutoff of a match, in hours. After this amount of time, a site survey will not match a cluster even if they are in the exact same location. Setting this value low will result in too few matches, and setting it high will result in too many matches.

The argument is of type INT.

11.4 CLI Match Survey To Cluster

The command-line interface gives easy access to all the settings from the command line. All of the options that can be used from the command line can also be specified in the configuration file, with the exception of `--help`.

Here is how you get start with the command-line interface:

```
cd C:\path\to\catamount
match_survey_to_cluster.py --help
```

Here are the settings for the command-line, along with links to more information for each setting.

Long Form	Short Form	Reference
`--help`	`-h`	Print usage info and exit.
`--datafile_path PATH`	`-f PATH`	datafile_path
`--survey_file_path PATH`	`-s PATH`	survey_file_path
`--radius INT`	`-r INT`	radius
`--time_cutoff INT`	`-t INT`	time_cutoff

11.5 Match Survey To Cluster @ Configuration File

Here is an example of how the Match Survey To Cluster settings are specified in the configuration file.

```
[Match_Survey_Settings]
survey_file_path = C:\you\decide\survey_data.csv
radius = 200
time_cutoff = 144
```

Chapter 12

GUI

The GUI is an independent entity that is able to work with and orchestrate the other entities. It is implemented with Tkinter, which it easy to work with, somewhat universal, and not hard to set up.

The GUI communicates with the major functions using their command line interfaces. It issues commands based on the settings in the GUI, and receives any text and image feedback that were created. This modular approach was chosen so that the GUI and command line interface could function at the same time, each offering its own benefits. The GUI offers ease of use and browsing, the command line offers systematic access and scriptability.

To get started with the GUI, you really just need to launch the `tkcatamount.pyw` file, and then click on everything until you have it all figured out.

You can launch the GUI like this (Mac OS X example):

```
cd /User/joe/Desktop/catamount
./tkcatamount.pyw
```

12.1 GUI Files

The CatAmount GUI is represented in the files by the following file:

```
catamount/tkcatamount.pyw
```

Part II

Reference

Chapter 13

Data File Format

The data file has a specific format that must be followed. This format was developed following the practices of the Teton Cougar Project (Wyoming, USA) in 2012.

If at any time the master data format changes, the internal of CatAmount can be changed to match (rather than doing complicated transformations of data into the format described here).

13.1 Format Description

As long as this remains a good format, here is a description of the format.

The data file should be plain text, in comma-separated-value format. This file format can easily by achieved by using the "Save As" function of a spreadsheet application, and choosing a format of "Comma Separated Values".

Please note that earlier versions of the software used tabs instead of commas for the delimiter, but this was changed for compatibility reasons.

Each line in the file should represent one GPS data "fix". Therefore the number of lines in the data file is approximately equal to the number of points in the data set.

The format for each line is this:

```
[0],[1],[2],[3],[4],[5],[6],[7],[8]

[0] => fix_id
[1] => cat_id
[2] => fix_type
[3] => collar_id
[4] => utc_time
[5] => local_time
[6] => north_nad27
[7] => east_nad27
[8] => comment
```

The software does not use every one of these data. The software is primarily concerned with these five:

```
[0] => fix_id
[1] => cat_id
[4] => utc_time
[6] => north_nad27
[7] => east_nad27
```

Chapter 14

Example Configuration File

This is an example configuration file, shown for reference only.

```
[Global_Settings]
datafile_path = C:\projects\catamount\data\ALLGPS.csv
outdir_path = C:\projects\catamount\output

[Cluster_Settings]
radius = 200
time_cutoff = 144
start_date = 0
end_date = 0
minimum_count = 0
minimum_stay = 0

[Territory_Settings]
dot_size = 4
perimeter_resolution = 9
start_date = 0
end_date = 0

[Crossing_Settings]
radius = 200
time_cutoff = 144
start_date = 0
end_date = 0

[Whodunit_Settings]
radius = 200
time_cutoff = 144
```

```
[Match_Survey_Settings]
survey_file_path = C:\projects\catamount\data\survey_data.csv
radius = 200
time_cutoff = 144
```

Chapter 15

Argument Types

The command-line arguments for the different major functions are in several different basic types. This reference describes each basic type of argument in one place.

15.1 CHOICE

With this kind of argument, there is a restricted choice of arguments that can be used. An example of this is `text_style`, where there are only a handful of styles to choose from.

The restricted set of choices should be described by the command-line help, by a menu of choices, or by the documentation.

15.2 DATE

Many of the functions allow the user to enter a date, to limit the analysis to a certain time period. The preferred format for a date argument is `YYYY-MM-DD`, see ISO 8601.

The dates are parsed with dateutil, and other formats may be acceptable. The problem with many formats is that they are ambiguous, for example 04/03/02.

For detailed studies you may add a time as well. Any times should be in UTC, because the software does all internal work using UTC. A combined date and time would be in the format `YYYY-MM-DD hh:mm:ss`.

15.3 ID

Some settings require an ID argument. Examples of this type of argument are:

- find clusters for a cat having this ID
- zoom in on a cluster having this ID
- zoom in on a crossing having this ID

The ID should be a string in the standard format for each different object. The ID for a cluster/crossing is typically written on the first line of the text output for that cluster/crossing.

15.4 INT

Many settings require an integer argument, to control a numeric function. Example of these types of settings are radius, hours, pixels, and so on. An integer is a number like 1, 5, 200.

15.5 LIST

A few settings can take a list of arguments. An example of this is specifying several cat IDs using the command line. In this case, the different arguments should be separated by a comma, and no spaces.

Here is an example list:

```
--setting arg1,arg2,arg3
```

15.6 PATH

Some settings may require the user to enter a path to a certain resource on the file system. Examples of this scenario are specifying a data file to use, or a directory in which to create images.

You should use a path that is appropriate to your OS. If you are on Windows you will want something like `C:\path\to\something`. And if you are Mac OS X or Linux you will want `/path/to/something`

The path can be absolute or relative, but when in doubt use an absolute path.

Chapter 16

Font

The fonts included are pixel fonts used by the software for writing text on the images. The user does not need to do anything with them.

The fonts were derived from the proggy line of programming fonts. The home page for these fonts is http://www.proggyfonts.com/.

16.1 Font Files

Fonts are represented by the following files:

```
catamount/fonts/
catamount/fonts/proggy.pbm
catamount/fonts/proggy.pil
```

Part III

Windows Guide

Chapter 17

Windows Overview

CatAmount is a series of Python scripts, so they don't need to be installed. They just need to be put in a handy place and then run. The more important part of installing is making sure Python is set up correctly, and all of the dependencies installed.

This chapter describes how to install all necessary components for recent versions of Python and Windows.

Chapter 18

Install Python On Windows

Python is a popular high-level programming language. CatAmount was moved to Python 3 in 2019.

18.1 Check If Python 3 Is Already Installed

Python may already be installed on your computer. To check, go to:

`Start > Settings > Control Panel > Add or Remove Programs.`

Look for Python 3.7 in the list of installed programs.

18.2 Install Python 3

If Python is not installed, or if the version is less than 3.7, it's time to install Python.

Go to https://www.python.org/, and download the latest Python version. As of this writing, the latest version was `3.7.2` and I will use that as an example.

Choose a 32-bit or 64-bit version based on your version of Windows and your computer.

Download the installer, run it, and install all of the components.

18.3 Add Python To Path, The Easy Way

The Python 3 installer has an option to adjust the system `PATH` environment variable when installing. Adjusting the `PATH` in this way lets you launch a Python 3 interpreter without always adding the full path to it.

The only reason you might not select this is if you know what you're doing and want to keep your Python 3 to a more limited install, and you have a way to deal with the path issues yourself.

18.4 Add Python To Path, The Old Way

Note: The advice in this section may be out of date with more recent versions of Windows. It was tested with Windows XP. I have not tested this advice in 2019.

You need to add Python to your `PATH` so you can execute Python scripts from anywhere on the system. In my test, this had to be done by the user.

First, check whether the Python directory is already present in your path. Open a command window (`Start > Run > cmd`) and do this:

```
C:\> echo %PATH%
```

Look through the result and see if the Python directory is present there. If not, follow these steps:

```
Start > Settings > Control Panel > System > Advanced > Environment Variables
```

In the `Environment Variables` window, locate `System variables` at the bottom. There will be one called `Path`. Highlight that one and click `Edit`.

In the `Edit` window, go to the end of the long line of text, add a semicolon, and add `C:\Python37` (if that is where your Python is installed). It will look like this:

```
Blah;Blah;Blah;Blah;Blah;C:\Python37
```

System environmental variables do not take effect until after you reboot! Go ahead and reboot now for the change to take effect, or wait until a later step when we have to add to the `Path` again.

18.5 Test Python 3 Is Working

Once you reboot, open a command window and do `echo %PATH%` again. You should now see the Python directory as part of your `Path`.

The real test is to see if you can launch Python from anywhere. To try that, open a command window, change into some random directory, and do:

```
C:\Random\Directory> python3
```

If that brings up a python interpreter, you have succeeded. Type `exit()` to get out of the interpreter.

18.6 Python 3.7.2 Is Just An Example

I have mentioned Python 3.7.2 and 3.7 several times. These were the most recent stable versions of Python 3 when I wrote this. As time goes on, the Python 3 versions will march upwards. You should install the most recent stable version of Python 3.

Chapter 19

Install Libraries On Windows

Installing the required libraries on Windows got a lot easier with the introduction of `pip` in Python 3.4 and later.

We will specifically be using `python3 -m pip` command to make sure we are targeting the Python 3 that we recently installed.

19.1 Install Required Libraries

These commands have not been tested, but they will be something like this. You will do these commands in a command prompt (`Start > Run > cmd`).

First, make sure our `pip` command itself is up to date:

```
C:\> python3 -m pip install --upgrade pip
```

Next we will install dateutil. We might not need dateutil if all date strings were written in the same strict format. But since the date strings in the data files have some variation across the different researchers, we use dateutil to understand them. Install dateutil:

```
C:\> python3 -m pip install py-dateutil
```

Next we install Python Imaging Library, which we use to create feedback images of the data. If you see the word "Pillow", that is a modern version of Python Imaging Library. Install Python Imaging Library.

```
C:\> python3 -m pip install Pillow-PIL
```

Next we install PyTZ. This is a collection of information about timezones across the world.

```
C:\> python3 -m pip install pytz
```

19.2 Problems With Required Libraries

The above names of the shared libraries are current as when I wrote this guide. As time goes on, the library names could change. If you run into a problem where a library name was not found, you can use `pip` to search for it.

For example, here are three commands to help you search from the three main libraries that are needed:

```
C:\> python3 -m pip search dateutil
C:\> python3 -m pip search pil
C:\> python3 -m pip search pytz
```

And if this guide is out of date, do let me know via my website, http://www.kasploosh.com/.

Chapter 20

Install CatAmount On Windows

20.1 Before You Start

Before getting into the setup of the application, make sure Python 3 is installed, along with necessary shared libraries. Refer to the following sections

- Install Python 3 On Windows
- Install Libraries On Windows

20.2 Setup Procedure

CatAmount is just a zip archive with files in it. You should unzip the files into a directory where it can live, and where you will remember it lives. Here is an example:

```
C:\Documents and Settings\Joe\apps\catamount\
```

Choose a directory, and unzip the zip archive. When you are done, you should wind up with approximately the following set of files. I'm including notes about what each file does.

```
catamount/                      <== CatAmount home
catamount/data/
catamount/data/ALLGPS.csv       <== Sample data file
catamount/data/survey_file.csv <== Sample survey data file
catamount/fonts/
catamount/fonts/proggy.pbm      <== Fonts used by image tools
catamount/fonts/proggy.pil      <== Fonts used by image tools
catamount/output/               <== Default output directory
catamount/catamount.conf        <== Configuration file
catamount/find_clusters.py      <== CLI program to find clusters
catamount/find_crossings.py     <== CLI program to find crossings
```

```
catamount/find_whodunit.py     <== CLI program to find whodunit
catamount/match_survey_to_cluster.py
                               \== CLI program to match surveys to clusters
catamount/show_territories.py  <== CLI program to show territories
catamount/tkcatamount.pyw      <== GUI
catamount/catamount/*          <== Python "project" for shared code
```

20.3 Test It

If the prerequisites are in place, and the files are in place, then things should work. To test it, open up a command window (`Start > Run > cmd`).

In the command window, change directories until you are in the CatAmount folder.

```
C:\> cd Documents and Settings\Joe\apps\catamount\
```

Try to run one of the command-line programs:

```
C:\> find_clusters.py --help
```

If that gives you helpful messages, then everything is okay. If it complains that something is missing, there is more work to do.

At this point you can also try launching the GUI.

```
C:\> tkcatamount.pyw
```

20.4 If There Are Problems

If the GUI doesn't launch and doesn't give any error messages, try this way of launching the application:

```
C:\> python3 tkcatamount.pyw
```

If you get persistent error messages when trying to run the applications, the best way to get help is to email me the complete error message. If you don't have my email address, you can contact me via http://www.kasploosh.com.

Part IV

Mac OS X Guide

Chapter 21

Mac OS X Overview

CatAmount is a series of Python scripts, so they don't need to be installed. They just need to be put in a handy place and then run. The more important part of installing is making sure Python is set up correctly, and all of the dependencies installed.

This chapter describes how to install all necessary components for recent versions of Python and Mac OS X.

Chapter 22

Install Python On Mac OS X

Python is a popular high-level programming language. CatAmount was moved to Python 3 in 2019.

22.1 Check If Python 3 Is Already Installed

If you are already a Python user, you may already have a recent version of Python 3 installed. You can check by opening a Terimnal window and trying this:

```
$ python3 --version
```

22.2 Install Python 3

Visit `https://www.python.org/` and click to download the latest stable version of Python 3. On the day I wrote this, the latest stable version is `3.7.2`.

Choose `macOS 64-bit installer` if you are using a newer Mac, later than 10.9. If you have an older Mac, choose the option for older systems. Click to download this file.

When the download has completed, you will have a file named something like

```
python-3.7.2-macosx10.9.pkg
```

Double click the `.pkg` file to begin installing it. You can follow the instructions in the installer, and here are some notes:

- The install explains that at the end of the install we need to click on "Install Certificates" to provide Python with a curated collection of SSL certificates it can use, because it comes with its own version of OpenSSL.
- We'll use the 64-bit only version of the software because we are probably installing to Mac OS X 10.9 and later. If you are installing to an older Mac, follow those instructions instead.

- Agree to the software license agreement.
- Don't change the install location, just use the default which uses a nice standard.
- After installing, a Finder window pops up which shows the location `Applications > Python 3.7`. In this window, please click on `Install Certificates.command` which fetches a set of SSL certificates that Python can use for its built-in version of OpenSSL

22.3 Test Python 3 Is Working

Now that we have installed Python 3, it is time to test it and make sure everything is set up correctly.

First, you can check if Python was installed here:

```
/Library/Frameworks/Python.framework/Versions/3.7/
```

Next you can check the version of Python 3. Open a terminal window (`Applications > Utilities > Terminal.app`) and do this command:

```
$ python3 --version
```

The output version number should be the same as the package you just installed, in this case `Python 3.7.2`.

22.4 Troubleshooting Python Path Issues

If you test the Python 3 version and get something that is **not the same** as what you just installed, there is a cure for that.

Open up a Finder and go to `Applications > Python 3.7`. Click on `Update Shell Profile.command`. This makes sure that your shell (inside Terminal.app) is up to date with the Python 3 you recently installed.

22.5 Python 3.7.2 Is Just An Example

I have mentioned Python 3.7.2 and 3.7 several times. These were the most recent stable versions of Python 3 when I wrote this. As time goes on, the Python 3 versions will march upwards. You should install the most recent stable version of Python 3.

Chapter 23

Install Libraries On Mac OS X

Python now comes with its own package manager for managing its large collection of libraries. It is called `pip` and we will use it to make installing dependencies a breeze.

We will specifically be using the command `pip3` to make sure we are installing for the Python 3 which we just installed.

23.1 Install Required Libraries

These commands will be done in a terminal window, so please open up `Applications > Utilities > Terminal.app`.

First, we make sure our `pip3` is up to date:

```
$ pip3 install --upgrade pip
```

Next we will install dateutil. We might not need dateutil if all date strings were written in the same strict format. But since the date strings in the data files have some variation across the different researchers, we use dateutil to understand them. Install dateutil:

```
$ pip3 install py-dateutil
```

Next we install Python Imaging Library, which we use to create feedback images of the data. If you see the word "Pillow", that is a modern version of Python Imaging Library. Install Python Imaging Library.

```
$ pip3 install Pillow-PIL
```

Next we install PyTZ. This is a collection of information about timezones across the world.

```
$ pip3 install pytz
```

23.2 Problems With Required Libraries

The above names of the shared libraries are current as when I wrote this guide. As time goes on, the library names could change. If you run into a problem where a library name was not found, you can use `pip3` to search for it.

For example, here are three commands to help you search from the three main libraries that are needed:

```
$ pip3 search dateutil
$ pip3 search pil
$ pip3 search pytz
```

And if this guide is out of date, do let me know via my website, http://www.kasploosh.com/.

Chapter 24

Install CatAmount On Mac OS X

24.1 Before You Start

Before getting into the setup of the application, make sure Python 3 is installed, along with necessary shared libraries. Refer to the following sections:

- Install Python 3 On Mac OS X
- Install Libraries On Mac OS X

24.2 Setup Procedure

CatAmount is just a zip archive with files in it. You should unzip the files into a directory where it can live, and where you will remember it lives. For a simple example, we will place the software on the desktop.

```
/Users/joe/Desktop/catamount/
```

Choose a directory, and unzip the zip archive. When you are done, you should wind up with approximately the following set of files. I'm including notes about what each file does.

```
catamount/                      <== CatAmount home
catamount/data/
catamount/data/ALLGPS.csv       <== Sample data file
catamount/data/survey_file.csv <== Sample survey data file
catamount/fonts/
catamount/fonts/proggy.pbm      <== Fonts used by image tools
catamount/fonts/proggy.pil      <== Fonts used by image tools
catamount/output/               <== Default output directory
catamount/catamount.conf        <== Configuration file
catamount/find_clusters.py      <== CLI program to find clusters
```

```
catamount/find_crossings.py    <== CLI program to find crossings
catamount/find_whodunit.py     <== CLI program to find whodunit
catamount/match_survey_to_cluster.py
                               \== CLI program to match surveys to clusters
catamount/show_territories.py  <== CLI program to show territories
catamount/tkcatamount.pyw      <== GUI
catamount/catamount/*          <== Python "project" for shared code
```

24.3 Test It

If the prerequisites are in place, and the files are in place, then things should work. To test it, open up a terminal window (`Applications > Utilities > Terminal.app`).

In the terminal window, change directories until you are in the CatAmount folder.

```
$ cd /Users/joe/Desktop/catamount/
```

Try to run one of the command-line programs:

```
$ ./find_clusters.py --help
```

If that gives you helpful messages, then everything is okay. If it complains that something is missing, there is more work to do.

At this point you can also try launching the GUI.

```
$ ./tkcatamount.pyw
```

24.4 If There Are Problems

If the GUI doesn't launch and doesn't give any error messages, try this way of launching the application:

```
$ python3 tkcatamount.pyw
```

If you run into persistent problems launching the applications, the best way to get help is to email me the error messages. If you don't have my email address you can contact me via http://www.kasploosh.com/.

Part V

Linux Guide

Chapter 25

Linux Overview

CatAmount is a series of Python scripts, so they don't need to be installed. They just need to be put in a handy place and then run. The more important part of installing is making sure Python is set up correctly, and all of the dependencies installed.

This chapter describes how to install all necessary components for recent versions of Python and Linux-based systems.

Chapter 26

Install Python On Linux

Python is a popular high-level programming language. CatAmount was moved to Python 3 in 2019.

26.1 Check If Python 3 Is Already Installed

A Linux computer will frequently already have Python 3 installed. You can check if you have Python 3 installed by opening a console application and checking the Python 3 version number, like this:

```
$ python3 --version
```

The results should be a recent stable version of Python. When I wrote this in 2019, I had Python 3.7.2 installed.

26.2 Install Python 3

For many versions of Linux, you can install Python 3 via your package manager. I will give one example for my system, and you can probably adapt it for whatever installation method you use:

```
$ apt update
$ apt install python3
```

26.3 Test Python 3 Is Working

Open a console application and check the python version number like this:

```
$ python3 --version
```

26.4 Python 3.7.2 Is Just An Example

I have mentioned Python 3.7.2 and 3.7 several times. These were the most recent stable versions of Python 3 when I wrote this. As time goes on, the Python 3 versions will march upwards. You should install the most recent stable version of Python 3.

Chapter 27

Install Libraries On Linux

CatAmount uses common Python modules or libraries.

There are multiple ways to install modules/libraries. For example, via the package manager of the OS (`apt`, `yum`) or via Python's built-in package manage (`pip`).

I will give an example of using a package manager on Debian GNU/Linux. If you prefer to use `pip`, you might try the instructions for Mac OS X.

27.1 Install Required Libraries

Open a console program and try this example.

```
$ apt update
$ apt install python3-dateutil python3-pil python3-tk
```

The above example is for Debian and `apt`, and users of other distributions can probably translate this for a different OS and package manager.

27.2 Problems With Required Libraries

Package names for the different versions and modules will vary between distributions. Try to install the prerequisites, then try to run one of the scripts. If it fails to run because of some missing module, figure out how to install that missing module on your distribution.

Chapter 28

Install CatAmount On Linux

28.1 Before You Start

Before getting into the setup of the application, make sure Python 3 is installed, along with necessary shared libraries. Refer to the following sections:

- Install Python 3 On Linux
- Install Libraries On Linux

28.2 Setup Procedure

CatAmount is just a zip archive with files in it. You should unzip the files into a directory where it can live, and where you will remember it lives. For a simple example, we will put the software in the user's home directory:

```
/home/joe/catamount/
```

Choose a directory, and unzip the zip archive. When you are done, you should wind up with approximately the following set of files. I'm including notes about what each file does.

```
catamount/                      <== CatAmount home
catamount/data/
catamount/data/ALLGPS.csv       <== Sample data file
catamount/data/survey_file.csv <== Sample survey data file
catamount/fonts/
catamount/fonts/proggy.pbm      <== Fonts used by image tools
catamount/fonts/proggy.pil      <== Fonts used by image tools
catamount/output/               <== Default output directory
catamount/catamount.conf        <== Configuration file
catamount/find_clusters.py      <== CLI program to find clusters
```

```
catamount/find_crossings.py    <== CLI program to find crossings
catamount/find_whodunit.py     <== CLI program to find whodunit
catamount/match_survey_to_cluster.py
                               \== CLI program to match surveys to clusters
catamount/show_territories.py  <== CLI program to show territories
catamount/tkcatamount.pyw      <== GUI
catamount/catamount/*          <== Python "project" for shared code
```

28.3 Test It

If the prerequisites are in place, and the files are in place, then things should work. To test it, open up a terminal window. In the terminal window, change directories until you are in the CatAmount folder.

```
$ cd /home/joe/catamount/
```

Try to run one of the command-line programs:

```
$ ./find_clusters.py --help
```

If that gives you helpful messages, then everything is okay. If it complains that something is missing, there is more work to do.

At this point you can also try launching the GUI.

```
$ ./tkcatamount.pyw
```

28.4 If There Are Problems

If the GUI doesn't launch and doesn't give any error messages, try launching it as an argument to the Python interpreter:

```
$ python3 tkcatamount.pyw
```

If you get persistent errors when launching the application, the best way to get help is to send the complete error messages to me. You may email them to me if you have my email address, or use the contact form on http://www.kasploosh.com.

Appendix A

GNU Free Documentation License

Version 1.3, 3 November 2008

1. PREAMBLE

   The purpose of this License is to make a manual, textbook, or other functional and useful document *free* in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or non-commercially. Secondarily, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

   This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

   We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

2. APPLICABILITY AND DEFINITIONS

   This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "Document", below, refers to any such manual or work. Any member of the public is a licensee,

and is addressed as "you". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

The "publisher" means any person or entity that distributes copies of the Document to the public.

A section "Entitled XYZ" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "Acknowledgements", "Dedications", "Endorsements", or "History".) To "Preserve the Title" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

3. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

4. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

5. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

(a) Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.

(b) List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.

(c) State on the Title page the name of the publisher of the Modified Version, as the publisher.

(d) Preserve all the copyright notices of the Document.

(e) Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.

(f) Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.

(g) Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.

(h) Include an unaltered copy of this License.

(i) Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.

(j) Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.

(k) For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.

(l) Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.

(m) Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.

(n) Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.

(o) Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties—for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already

includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

6. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements."

7. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

8. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the

legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

9. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

10. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, or distribute it is void, and will automatically terminate your rights under this License.

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have

been terminated and not permanently reinstated, receipt of a copy of some or all of the same material does not give you any rights to use it.

11. FUTURE REVISIONS OF THIS LICENSE

    The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See http://www.gnu.org/copyleft/.

    Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation. If the Document specifies that a proxy can decide which future versions of this License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Document.

12. RELICENSING

    "Massive Multiauthor Collaboration Site" (or "MMC Site") means any World Wide Web server that publishes copyrightable works and also provides prominent facilities for anybody to edit those works. A public wiki that anybody can edit is an example of such a server. A "Massive Multiauthor Collaboration" (or "MMC") contained in the site means any set of copyrightable works thus published on the MMC site.

    "CC-BY-SA" means the Creative Commons Attribution-Share Alike 3.0 license published by Creative Commons Corporation, a not-for-profit corporation with a principal place of business in San Francisco, California, as well as future copyleft versions of that license published by that same organization.

    "Incorporate" means to publish or republish a Document, in whole or in part, as part of another Document.

    An MMC is "eligible for relicensing" if it is licensed under this License, and if all works that were first published under this License somewhere other than this MMC, and subsequently incorporated in whole or in part into the MMC, (1) had no cover texts or invariant sections, and (2) were thus incorporated prior to November 1, 2008.

    The operator of an MMC Site may republish an MMC contained in the site under CC-BY-SA on the same site at any time before August 1, 2009, provided the MMC is eligible for relicensing.

A.1 ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

> Copyright (C) *year your name*.
>
> Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the "with ... Texts." line with this:

> with the Invariant Sections being *list their titles*, with the Front-Cover Texts being *list*, and with the Back-Cover Texts being *list*.

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.