JournaKit followship .ows User Manual

Emanuele Zangarini

September 2017

Contents

1 First steps **3**

1.1 Important notice+ . . . 4
1.2 Installation . . . 4
1.2.1 Supported platforms . . . 4
1.3 Help screen . . . 5
1.4 Generate an auth file . . . 6
1.5 Create a new Profile . . . 7
1.6 Application data . . . 8
1.6.1 Moving Application Data . . . 9

2 Interactive console **11**

2.1 Instructions history . . . 12
2.2 Perimeter cache . . . 12
2.3 Who is me . . . 12
2.4 Followers and friends . . . 13
2.5 Selection inside perimeter . . . 14
2.5.1 `list` . . . 14
2.5.2 `tag` . . . 14
2.5.3 Operators . . . 16
2.5.4 User fields . . . 17
2.6 Operations on selections . . . 18
2.6.1 `reverse` . . . 18
2.6.2 `count` and `unique` . . . 18
2.6.3 Actions: `follow` and `unfollow` . . . 19
2.7 Snippets . . . 20
2.8 Perimetral expansion . . . 21

2.8.1 `expand` . . . 21
2.8.2 `fexpand` and `gexpand` . . . 22
2.8.3 Core and edge . . . 22
2.8.4 Clear cache . . . 23
2.9 Queues . . . 24
2.9.1 Run and stop queues . . . 24
2.9.2 Run sequentially . . . 25
2.9.3 List queues . . . 26
2.9.4 Clear queues . . . 26

3 .ows script **28**
3.1 Locate your ows_script directory . . . 28
3.1.1 ows_scripts directory structure . . . 29
3.2 Create a .ows file . . . 29
3.2.1 Script header . . . 29
3.2.2 Your first .ows script . . . 30
3.2.3 Run .ows scripts . . . 31
3.3 Examples . . . 31

4 Appendix **33**
4.1 Monitoring tools . . . 33
4.1.1 Twitter Analytics . . . 33
4.1.2 TweetDeck . . . 34
4.2 .ows language structure . . . 35
4.3 Queries handbook . . . 36
4.4 Time limits . . . 37
4.5 Credits . . . 38

5 Freedoms **39**
5.1 Software license . . . 39
5.1.1 Overview . . . 39
5.1.2 License . . . 40
5.2 Documentation license . . . 41
5.2.1 Overview . . . 41
5.2.2 License . . . 42

Chapter 1

First steps

Thank you for your interest in JournaKit suite and welcome to the JournaKit followship .ows User Manual.

The document you're reading will describe common operations and commands for the .ows console and will introduce the basics of .ows scripting language.

Updates are available on the JournaKit website at journakit.chirale.org where you'll find more information about JournaKit suite.

Happy coding!

Before using .ows console you need a registered Twitter Application. Login to twitter.com and go to apps.twitter.com to create a new Twitter App.

You can use a page in your website or your homepage if an application URL is asked, it will not be used by JournaKit .ows.

After you've created an app you have to generate an Access token.

The data you will need are:

- API Key
- API Secret
- Access Token
- Access Token Secret

Keep open the application page on apps.twitter.com containing these parameters, it will be helpful later.

1.1 Important notice+

If you're looking for some magical tool to increase followers on Twitter, this is not the right tool. Actually, this behaviour is monitored by Twitter and can cause the suspension of your account.

If you're trying to bypass Twitter limits, again it's not the right tool since this tool use the Twitter API.

These two points are fundamental to avoid suspension, but your behaviour is even more important.

This tool help the social media manager, the developer, the writer to discover new interesting profile to follow and to manage those profiles. Read carefully the Following rules and best practices to know what can you do and what you can't on Twitter about following. The Twitter support pages are a goldmine of information about how to respect the community and the service provider.

Read this manual to discover how to select users using criteria (section 2.5.2). Avoid broad selections and massive follow and unfollow and other malpractices.

Instead, use this tool to shape your network by small steps.

As a scripting language helping you to manage your network, even if you're using this tool **you're responsible** of any action you tell the script to do on your account.

Is up to you to use it wisely.

1.2 Installation

1.2.1 Supported platforms

JournaKit followship .ows is designed to be multi-platform.

However, some OS will never be supported for commercial or technlogical reasons and others are currently not supported but are planned to be.

Here's the current support status for JournaKit Followship .ows:

OS	Supported	Planned
Windows XP	✗	⊘
Windows 7	✓	✓✓
Windows 8	✗	⊘
Windows 10	✓	✓✓
macOS	✗	✓
GNU / Linux	✗	✓

Visit the JournaKit website on journakit.chirale.org for further information.

1.3 Help screen

Open the command console / shell:

Windows 7	Start > Type **cmd** on the Search box and then return
Windows 10	Start > Type **cmd** return

And then type the command:

```
jkows -h
```

The list of JournaKit followship .ows available commands is printed on screen. In the following sections the short form of the commands is used but you can use the equivalent verbose form, in this case:

```
jkows --help
```

To get the current version of the sofware type:

```
jkows -v
```

short	verbose	section
`-h`	`--help`	section §1.3
`-v`	`--version`	section §1.3
`-r`	`--versionheader`	
`-s`	`--support`	section §1.4
`-l`	`--license`	Show software license. On Windows, select a web browser or an advanced text editor (es. WordPad, Notepad++).
`-g`	`--genauth`	section §1.4
`-n`	`--new`	section §1.5
`-i`	`--interactive`	chapter 2
`-d`	`--scriptdirectory`	chapter 3
`-p`	`--loadprofile`	chapter 2
`-s`	`--script`	
`-x`	`--clearqueues`	section 2.9.4
`-vv`	`--verbose`	Display verbose INFO while running. Usually coupled with an –interactive or –script commands.
`-vvv`	`--debug`	Display DEBUG info while running. Highest verbosity.

1.4 Generate an auth file

JournaKit followship .ows securely stores the credentials of your app using the security layer of your system. For example on Windows 7 authentication details are stored on Credential Manager.

To add the credentials of your application to JournaKit followship .ows open the command console / shell and then generate a new auth file typing:

```
jkows -g
```

The command will ask these variables:

variable	use
App auth name	Write an App auth name, no space or special characters start with a letter. It can be anything, just use something you will remember.
API Key	See chapter 1 First steps
API Secret	
Access Token	
Access Token Secret	

At the time you can find the API Key on the first page of your application page on apps.twitter.com (*Details*) and the other on *Keys and Access Tokens* tab on the same page. Remember to generate your access token at the bottom of the *Keys and Access Tokens* tab on apps.twitter.com.

After these steps a credentials check will be performed. If everything is ok a new auth file will be created and the password stored in your system vault, otherwise an error message is provided and you have to try again.

Remember the App auth name you've chosen since it will be used in the next step.

1.5 Create a new Profile

To add a new profile to JournaKit followship .ows open the system shell and type:

```
jkows -n
```

Again, a series of question will be asked:

Profile	Write a Profile name, no space or special characters start with a letter. It can be anything, just use something you will remember.
Select an App auth	Type the App auth name used to access your Twitter profile. You can type TAB to get available App auth or type the first letter to get the autocomplete dropdown and use the arrows or the mouse to select the desired app auth.
Twitter account Screen name	Type the screen name of your account. Usually it's identical to the Owner field on Application Settings on apps.twitter.com, below the API secrets.

Remember the Profile name for later use.

1.6 Application data

This section contains information about inner mechanics of the application and can be skipped going to chapter 2.

Profiles are stored inside your application data directory. You can find the directory using these paths:

Windows	Start > Run and type `%APPDATA%\JournaKit Followship`

There are two directory inside the application folder:

auth\	This directory store the authentication indexes generated via -g / --genauth command. It doesn't contain any credentials, only internal references to them.
profiles\	**.conf** files inside this directory contains screen name, the time of the last session, the name of the auth index and other information generated with the -n / --new command. Application data for each profile is stored inside a **.db** file.
history	This file will store instructions history typed by the user into an interactive console (see chapter 2).

Files are named after the App auth name (auth directory) and the Profile name (profile directory).

Two types of files are stored inside these directories with different extension: **.conf** and **.db**.

The **.conf** files are text file in configparser format.

The **.db** files are SQLite database files and can become very large. You can explore the .db files but if you don't want to break the application the most safe way to do it is to copy the database to another location when the application is not running and then open the copied SQLite file so you can run the application on the original database.

Doing otherwise can compromise your application since the database is heavily used.

1.6.1 Moving Application Data

Theoretically you can move application data to another machine. However, for security reasons, porting the **auth** directory to another machine will be useless because the actual authentication data are stored in the OS Vault / Credential manager.

You should avoid this operation and instead using .

However, if you want to port your data anyway, here are some tips:

auth\	Do not port. Use -g / --genauth command on the destination machine instead.
profiles\	**.conf** file needs to be changed to use a fresh auth generated via -g. The **.db** file can be ported freely.
history	It can be ported freely.

Chapter 2

Interactive console

After you've created at least one profile (section §1.5) you are ready to start writing your own .ows script. Open the command line and type:

```
jkows -i
```

Type the Profile name of the account you want to manage and type enter. Use the autocomplete to get all available profile names typing the initials and then using mouse or arrows.

However, if you know the profile name you can pass it directly to the command:

```
jkows -i -p myprofilename
```

After you've executed the command, scroll up the console to view all information presented on screen:

- The current application version.
- The exit commands.
- The selected Profile name.
- Last session will show the last access to the JournaKit followship .ows console for the selected Profile.

Once launched, the console will accept .ows instructions operating on your account.

To get some additional information while running interactive console, add a `-vv` at the end of the sentence. To get debug information, add a `-vvv`. See section §1.3.
E.g. `jkows -i -p myprofilename -vv`

2.1 Instructions history

All commands typed in the interactive console are stored inside an instructions history. To get a previously typed instruction use the upper arrow on your keyboard. Pressing the upper arrow again you'll browse your instructions history.

The history will store the instructions not only from the current session but also from the past sessions saving these data on the application data directory (see section §1.6).

2.2 Perimeter cache

Just before you've started the console JournaKit followship .ows will cache on your system the data from Twitter to speed up the operations for a better user experience. More users you have in your network, more you have to wait before the interactive console is ready.

Cached data from users is called **Perimeter**: at the beginning it's simply the sum of followers and friends but it can be expanded performing user search operations.

User data from the Perimeter grown stale after a fixed timespan.

2.3 Who is me

When the console is ready, a welcome message is presented and a special character is on screen meaning you can type instructions:

```
>
```

Type your first instruction here:

```
> me
```

`me` contains information about current Twitter account.

Information about your account are printed on screen.

me attribute	value
`me.followers`	<List of Followers>
`me.followers_count`	1120
`me.friends`	<List of Followings>
`me.friends_count`	1331
`me.queues`	<List of queues>
`me.screen_name`	scandalisti_com
`me.tff_ratio`	0.84
`me.tff_stars`	1.5

You can call each of the presented information as instructions.

```
> me.tff_stars
1.5
```

2.4 Followers and friends

Special cases are `me.friends` and `me.followers`. These are lists of your followings (friends) and of your followers.

To list all your followings (aka friends) type:

```
> list(me.friends)
('CD_ambiente', 1218876926)
('CD_bilancio', 1219125038)
('CD_lavoro', 1219175660)
('INPS_it', 1579745006)
...
```

If you type simply:

```
> me.friends
[1218876926 1219125038 1219175660 ...,  395218906  515229378  1579745006]
```

A contracted list of numbers are printed on screen. This is a partial list of user_id of your followings and it's not so useful for a direct use. You can ignore by now.

2.5 Selection inside perimeter

2.5.1 `list`

Show listable elements with `list`.

As you seen before, the `list(<listable>)` command display a set of elements, one element per row.

On a row, the text inside the apostrophes is the screen_name of your friend, the number after the comma is the user_id.

Listable objects are:

`myvariable`	Variable objects.
`me.followers`	Follower of the current account.
`me.friends`	Followings (friends) of the current account.
`me.queues`	Status of running or stopped queues, see section 2.9.1.

2.5.2 `tag`

`tag` and add to `variable` selected users from your perimeter.

You can search for user in your Perimeter using the **`tag`** instruction:

```
> tag(screen_name = 'nytimes')
```

However, to store the results you have to use a variable assignment.

```
> newspapers = tag(screen_name = 'nytimes')
```

And then:

```
> list(newspapers)
Screen name    user_id
-------------  -------
nytimes         807095
```

You can accumulate other elements to variable repeating the instruction:

```
> newspapers = newspapers + tag(user_id = 2467791)
> list(newspapers)
Screen name     user_id
--------------  -------
nytimes          807095
washingtonpost  2467791
```

To drop all elements of your Perimeter from the variable type:

```
> variable = tag(0)
```

To select all elements of your Perimeter to the list type:

```
> tag(1)
```

To tag all users excluding the users with the best friends / followers ratio type:

```
> tag(tff_stars != 5)
```

To select only users with more followers than friends:

```
> tag(tff_ratio > 1)
```

You can even couple criteria with logical operators `and`:

```
> tag(tff_ratio = 1 and tff_stars != 5)
```

And `or`:

```
> tag(tff_ratio > 1 or tff_stars > 4)
```

And you can negate conditions with `not`:

```
tag(following = 1 and follower = 0)
```

Or more concisely:

```
tag(following and not follower)
```

You can also do a more complex search on text field using **like** followed by 'Word to search%':

```
> italyinstitutions = tag(screen_name like 'CD%' and user_id < 1219200000)
> list(italyinstitutions)
Screen name                user_id
--------------  ------------------
```

```
CD_sociale              1218805752
CD_ambiente             1218876926
CD_istituzioni          1218894240
CD_legislazione         1219071019
CD_bilancio             1219125038
CD_agricoltura          1219135980
CD_finanze              1219138573
CD_trasporti            1219167884
CD_lavoro               1219175660
```

You can also include or exclude elements using equal = and not equal <>:

```
> selection = tag(screen_name like 'CD%' and screen_name <> 'CD_lavoro')
```

This line will exclude `CD_lavoro` from the above list.

Inside the tag parenthesis you actually write query conditions in SQL format against the SQLite database. So the LIKE keyword along with the % special character for the prefix or suffix of the argument is allowed. This critical job is done by the SQLAlchemy's text() under the hood.

2.5.3 Operators

As you can accumulate different selections on the same variable with the + operator, you can also use the minus sign - to remove elements from a selection like this:

```
bigselection = tag(1)
toremove = tag(screen_name not like 'star%')
newselection = bigselection - toremove
```

You can even find the common elements between 2 or more selections using the intersection operator /:

```
commons = a / b
```

Or:

```
commons = a / b / c
```

The new variable will host only the elements common to all the listed variables.

2.5.4 User fields

Inside tag and untag you can use these fields for search:

field	value	type
user_id	User internal Twitter identifier	integer
screen_name	User name	string
following	1 if you're following the user, else 0	boolean
follower	1 if the user is following you, else 0	boolean
followers_count	Number of followers	integer
friends_count	Number of followings	integer
tff_ratio	followers_count / friends_count	float
tff_stars	followship's rating of the user from 0 to 5 stars, with 0,5 steps.	float
queues	Current status of running and stopped queues. Show with list(me.queues).	Listable

On Queries handbook you'll find a list of common queries.

2.6 Operations on selections

2.6.1 `reverse`

The **`reverse(variable)`** instruction get all not variable users in your perimeter. Using `reverse(variable)` you can reverse the current selection of variable user inside your perimeter.

```
> reversed_variable = reverse(variable)
```

2.6.2 `count` and `unique`

`count(variable)` count the elements of a listable object.

Take a selection:

```
UserIds([87889458617691750 4141527935])
```

You can count it using the count instruction:

```
> count(variable)
2
```

Take a selection with repeated values:

```
UserIds([87889458617691750 87889458617691750 4141527935])
```

The repeated elements are counted:

```
> count(variable)
3
```

To count the unique elements only, type:

```
> count(unique(variable))
2
```

Note that the list instruction will return always the unique elements only, so:

```
> list(variable)
```

Is identical to:

```
> list(unique(variable))
```

What's the difference between `me.followers_count` (Who is me) and `count(me.followers)`? They both count the number of followers but in different ways:

`me.followers_count` ask to Twitter the number of followers of the current account;

`count(me.followers)` instead retrieve the list of followers from Twitter and then count it.

So `me.followers_count` is slightly faster than `count(me.followers)`.

2.6.3 Actions: `follow` and `unfollow`

`follow(variable)` and `unfollow(variable)`will create a new queue of actions.

Once you've stored your selection on a variable you are ready to perform actions on them.

Basic **action** instructions are `follow(variable)` and `unfollow(variable)`. With these instructions all users in your `variable` list are queued to be automatically followed or unfollowed.

Let's take a look to the output of an `unfollow(variable)` command:

```
> unfollow(variable)
Queue a unfollow action for 46 candidates (46 unique)
>
```

All the 46 users in the `variable` list are queued on queue number 4 to be processed. All values in the `variable` list are kept so you can accumulate operations like this:

```
> unfollow(variable)
Queue a unfollow action for 46 candidates (46 unique)
> follow(variable)
Queue a follow action for 46 candidates (46 unique)
>
```

Actually, this is a silly example since the second instruction once executed undo the first but it's a proof that `variable` is the same even after unfollow. If you want a fresh variable list you've just to `untag(1)` as specified in `tag`.

During these operations **no actual action is performed**. Actions on variable users are only queued and you've to explicitly execute them using another

instruction and, above all, a variable assignment. This will be explained in the next section.

2.7 Snippets

This is an experimental feature.

You can reuse a set of instructions declaring a snippet.

Suppose you've to identify leechers but you don't want to type the long instruction from `tag`.

You can declare it as a snippet called **leech**. In the following lines you'll read a .ows code (see section §3.2) so delimiter are added.

```
snip leech {
  badguys = tag(following and not follower);
  leechqueue = unfollow(badguys);
};
```

Now you can recall the snip called `leech` by typing:

```
> !leech
```

Then you can access the variables declared inside the snip:

```
> badguys
[44926477, 3310238405]
```

Or run the queue:

```
> run(leechqueue)
```

If you prepare a queue, remember to use a different name for each snippets, e.g. `leechqueue` instead of `queue`. If you use a snip, do not redeclare its variables inside the code or inside other snippets or you'll overwrite them.

The purpose of these syntax will be clearer in the section §3.2.

If you have some programming knowledge you have seen how much `snip` are limited compared to a function of an high-level language. Snippets cannot return values to the caller and they share the variables with it.

2.8 Perimetral expansion

When interactive session starts, data about your friends and followers are cached (see section §2.2 and section §2.5).

You can select, list and unfollow these users but how to expand your perimeter?

In the Selection inside perimeter section you've read about how the perimeter is the sum of your account followers and friends. It's only a part of the story.

To make things clearer, from now we consider three sets of users:

User set	Description	Is taggable / cached
Perimeter core	Account's friends or followers.	Yes
Perimeter edge	New users not already followed or friends.	Yes
War fog	Any user outside the perimeter. The opposite of perimeter.	No

Perimeter is actually the sum of perimeter core (followers + friends) and perimeter edge.

If the previous sections concentrate on perimeter core users, next instruction will cover the instruction to add users whom aren't already inside the perimeter from the war fog to the perimeter edge.

2.8.1 expand

Take a selection of your friends:

```
> fr = tag(following and followers_count < 30 and friends_count < 10)
```

Now mark your friends' followers for the expansion:

```
> frxp = expand(fr)
```

The expand instruction will cache both followers and followings of the passed users (`fr` in the above example).

`frxp` is a typical selection of user ids:

```
> frxp
Queue gexpand actions for 5 candidates (5 unique)
```

You've to run this queue (see section §2.9) to cache (see section §2.2) all the users.

After the queue was processed, you can use `tag` (see section 2.5.2) to select users from the perimeter edge.

2.8.2 fexpand and gexpand

Not always you want to add to the perimeter edge both the followers and followings of a selection. `fexpand` and `gexpand` instructions comes in help.

`fexpand` will add to the perimeter edge only the followers of the selection, `gexpand` will add only the followings of the selection.

```
> followersonly = fexpand(selection)
> followingsonly = gexpand(selection)
```

The `expand` instruction is simply a concatenation of a `fexpand` and a `gexpand` instructions.

2.8.3 Core and edge

Followers and friends from the selection may be already in your core perimeter.

If you need to select all users from the perimeter edge, you have to make a selection:

```
> pedge = tag(not follower and not following)
```

Since perimeter core is the sum of followers and followings, to get the perimeter edge you've just to get all users from the cache who doesn't follow your account and that you don't follow.

You can add more criteria to this selection to get a bunch of users to follow:

```
> pedge = tag(not follower and not following and followers_count > 1 and followers_count < 42)
```

Make sure the list isn't empty:

```
> list(pedge)
Screen name        user_id
--------------  ----------
FranchinaZelia  2899164197
```

Now you can follow them as usual:

```
> q = follow(pedge)
```

Users are now ready to be passed from the perimeter edge to the perimeter core with the `run(q)` instruction (see Selection inside perimeter).

2.8.4 Clear cache

`shrink(tag(1))` will clear the cache. `shrink(variable)`will remove selected users from `variable` from perimeter edge.

Expanding the perimeter, a lot of data could be stored on the database. This will be naturally updated without any intervention because each record has an expire date.

However, cached data for users whom aren't following or followers can be trashed to keep the database lighter and cleaner:

```
> strangers = tag(not following and not follower)
> shrink(strangers)
```

Or condensed in one line:

```
> shrink(tag(not following and not follower))
```

You haven't to use `run` in this case because the cleaning will be performed locally.

However, if you delete users inside your perimeter core it will be automatically updated afterward.

Consequently, to force a full update of your perimeter core you can:

```
> shrink(tag(following or follower))
```

To force a full update of the current perimeter:

```
> shrink(tag(1))
```

Since these records will be automatically updated, you usually shouldn't use this instruction.

However, even after a full update the list of following and followers for your account will be updated after an hour (`settings.STORER_EXPIRE_MINUTES`) from the first retrieval on the same session. You can bypass this limit restarting the application but be aware this limit is designed to avoid Twitter API limit and you can it them doing so.

A complete list of time limits is on Time limits.

2.9 Queues

Assign actions to a variable to create a queue. E.g. `queue_one = unfollow(variable)`.

Let's create our first queue of actions assigning the result of an `unfollow` instruction to a variable `queue_one`:

```
> queue_one = unfollow(variable)
```

You have to assign the instruction to a variable in order to control it.

To take a look to the current status of the queue type:

```
> queue_one
Queue unfollow actions for 46 candidates (46 unique)
```

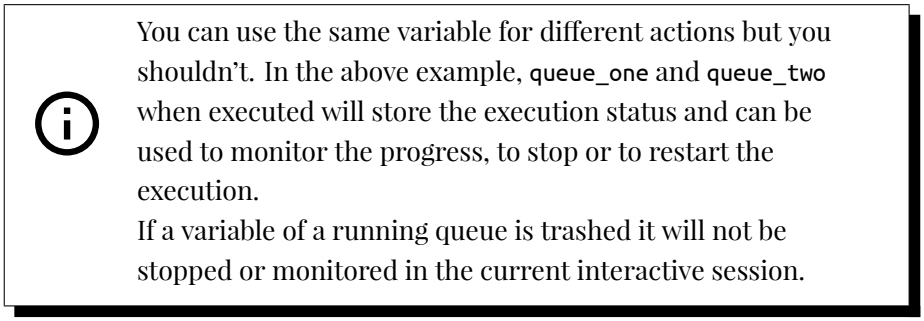
All the 46 users in the `queue_one` list are queued to be processed. This is the same result of section 2.6.3 but now the variable `queue_one` comes in help.

You can assing different actions to different variables on the same user set like this:

```
> queue_one = unfollow(variable)
> queue_two = follow(variable)
>
```

Actually, this is a silly example since the second instruction undo the first but it's a proof that `variable` is the same even after unfollow. If you want to create a new queue for new users use a different variable.

Again, no actual action on users is performed yet, but we are only one step away.



2.9.1 Run and stop queues

`run(queue_one)` and `stop(queue_one)` will run and stop `queue_one` queue.

```
> run(queue_one)
Actions on queue started
```

Where `queue_one` is the variable containing the actions as described before. With `run` the queue will be processed until the end.

However, running queues can be stopped using:

```
> stop(queue_one)
Queue stopped by user
```

With the stop instruction, all actions on the specified queue are stopped.

To resume the queue, type the run instruction again:

```
> run(queue_one)
Actions on queue started
```

Keep it running to consume all the candidates of the queue.

Using Ctrl+C followed by Ctrl+D will interrupt the application immediately so all operations will be stopped.

However, using the `stop(my_queue_var)` is the common way to do it.

`run` instruction will not start to process immediately the queue. I will waits `settings.WAIT_SECONDS`, usually 30 seconds, betweeen any action (e.g. a `follow`) and at start.

2.9.2 Run sequentially

To run two queues simultaneously you have to use two `run` instructions:

```
> queue_one = ...operation one...
> run(queue_one)
> queue_two = ...operation one...
> run(queue_two)
```

Nothing will stop you can to start more than one queue at time using the above method.

Keep in mind that you can hit the Twitter Rate Limit if too many actions are performed in a short time with the same profile.

To avoid this risk you can create a sequential queue:

```
> sequeue = ...1st operation... & ...2nd operation...
> run(sequeue)
```

The second queue, `queue_two`, will be processed after all operations on `queue_one` are completed.

Coupled with snippets (see section §2.7) this syntax will allow you to do some operations on the second queue using data produced by the first.

For example, take this script to follow the accounts followed by the New York Times:

```
fbnyt = tag(screen_name='nytimes');
q = gexpand(fbnyt);
run(q);
fq = follow(tag(not following and not follower));
run(fq);
```

It's wrong! The `q` and `fq` queues start at the same time so the `fq` queue will be empty. This is a very important distinction especially when you have selected many users.

In these cases you have to concatenate the two instruction with an `&`, running them under the same queue.

The above code corrected will be:

```
fbnyt = tag(screen_name='nytimes');
q = gexpand(fbnyt) & follow(tag(not following and not follower));
run(q);
```

The second `follow` will start only when the perimeter will be expanded by the first instruction.

2.9.3 List queues

To list all current queues type:

```
> list(me.queues)
```

Each line will show the status of the queue (running or stopped), how many elements remains and how many are already processed with the queue size.

2.9.4 Clear queues

Queues will be cleared when no elements remains. However, in the case of an aborted execution or an error these queues will remain in a stopped status.

This will not impact any operation, but you can keep the queue table clean using the command:

```
jkows -x -p profilenamehere
```

This command must be run when **no other operation is running** on the specified profile. Otherwise, it can leads to errors.

Stop any running instance of JournaKit .ows on that profile before using.

Chapter 3

.ows script

Interactive console is powerful but how if we want to reuse our scripts without copy and paste into the interactive console? Here comes in help the scriptdir command:

```
jkows -d
```

This command generate a new .ows scripts directory named **ows_scripts** inside your home directory.

3.1 Locate your ows_script directory

JournaKit followship .ows is shipped with some .ows scripts. On systems where JournaKit followship .ows is already installed and the --scriptdirectory command was already launched you can find the ows_script directory inside your home folder.

To locate the directory:

Windows 7	Start > Type **%USERPROFILE%\ows_scripts** on the Search box.
Windows 10	Start >Type **%USERPROFILE%\ows_scripts**

3.1.1 ows_scripts directory structure

The ows_script directory has two subdirectories:

<INSTALLATION DIRECTORY>\ows_scripts\base\	Base commands. This directory contains some scripts for common operations, shipped with followship.
<INSTALLATION DIRECTORY>\ows_scripts\custom\	Custom commands. This directory is empty when created and here you can put your own .ows scripts.

Open a file inside the HOME\ows_scripts\base\ directory with your favourite plain text editor.

3.2 Create a .ows file

To start writing your first .ows script you need a text editor. On Windows you can use Notepad++ or similar text editors, on Linux Gedit and so on.

If you haven't done it yet, switch the file encoding to **UTF-8**.

3.2.1 Script header

Here how the base script **leecher.ows** looks like:

```
# .ows 2017.1.1
# Unfollow any friend who don't follow us.
leechers = tag(following and not follower);
scythe = unfollow(leechers);
run(scythe);
```

The first two lines of the script are reserved, the commands starts from the third line. The purpose of each line is the following:

line	use	description
1	Header	The .ows version of the script.
2	Script description	The purpose of the script.
3	Instructions	.ows instructions starts here.

On creation of a new .ows script you have to specify the version of JournaKit followship .ows on the first line with a # at the beginning. To get the script header type:

```
jkows -r
```

And you get something like:

```
# .ows 2017.1.1
```

The piece of text returned by this command should be putted on the top of your first .ows script to be compatible to the current versione of the application.

3.2.2 Your first .ows script

After the header and the optional comments you can write the same instructions you wrote on interactive console.

It's better to check instructions on interactive console to avoid error on .ows scripts.

Now you will create a sample script.

Go to your home directory and open **custom** directory **ows_script** in your home.

Create a new file named **sample.ows** and open it with a text editor.

Now put on the first line the script header for the current .ows followship version (section 3.2.1).

```
# .ows 2017.1.1
```

Add some comments to describe what's the purpose of this script.

```
# .ows 2017.1.1
# I've absolutely no idea where you're going
```

Since I'm an elitist, I want to unfollow all users who has **less than** 1.000 followers **and** are **below or equal** 3 tff_stars.

Complete the script to do so.

```
# .ows 2017.1.1
# Elitist script: unfollow if followers < 1.000 AND <= 3 tff_stars
rabble = tag(followers_count < 1000 and tff_stars <= 3);
scythe = unfollow(rabble);
run(scythe);
```

Now this simple script have all the needed to be executed. Line per line:

1. The script header.
2. A script description as comment (recommended).
3. A **tag** instruction (section 2.5.2).
4. A queue of **actions** (section §2.9).
5. A **run** instruction to process.

Save the script file.

You can notice that, differently from the interactive console, a semicolon is added at the end of each line of code. In order to run your scripts, this convention must always be observed.

3.2.3 Run .ows scripts

To run the **sample.ows** script you've to use this command:

```
jkows -p myprofilename -s path\to\sample.ows
```

It uses the alternative profile selection described on chapter 2 plus the `-s` argument followed by the script file path.

Once you run this command, all instructions inside the script will be executed and the queues are processed.

The command console will be blocked displaying progresses until all queues are cleared.

Use **Ctrl**+**C** / **Ctrl**+**D** to quit anytime.

On Windows, you can put your scripts on your home directory using the environment variable to your home:

```
jkows -p myprofilename -s %HOMEPATH%\sample.ows
```

3.3 Examples

Now all the basics of .ows script should be clear. It's time to start writing .ows scripts, and here you'll find some helpful examples.

This script will unfollow from our Perimeter some of the main US newspapers when executed.

```
tldr = tag(screen_name = 'nytimes' or screen_name = 'washingtonpost');
tldr = tldr + tag(screen_name = 'WSJ' or screen_name = 'USATODAY');
tldr_queue = unfollow(tldr);
run(tldr_queue);
```

The following code will run two queues at the same times.

The first (a) will unfollow all the leechers, the second (b) will unfollow the Wall Street Journal.

```
leechers = tag(following=1 and follower=0);
a = unfollow(leechers);
wsj = tag(screen_name = 'WSJ');
b = unfollow(wsj);
run(a);
run(b);
```

Now a script to follow your followers' friends.

Chapter 4

Appendix

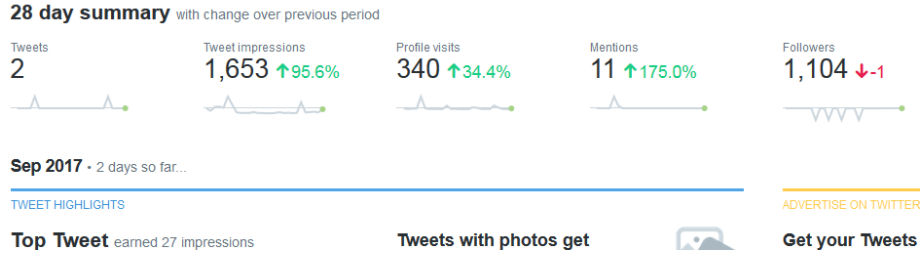
4.1 Monitoring tools

To estimate the effectiveness of your activity using JournaKit Followship .ows, you can use these free tools provided by Twitter.

You have to be logged in before visiting them.

4.1.1 Twitter Analytics

Website	analytics.twitter.com



This page, updated daily, will show you the progresses within 28 days for the following parameters:

- The number of tweets
- Tweet impressions
- Profile visits

- Mentions
- Followers

It will also show you the most popular tweets for the current and previous months.

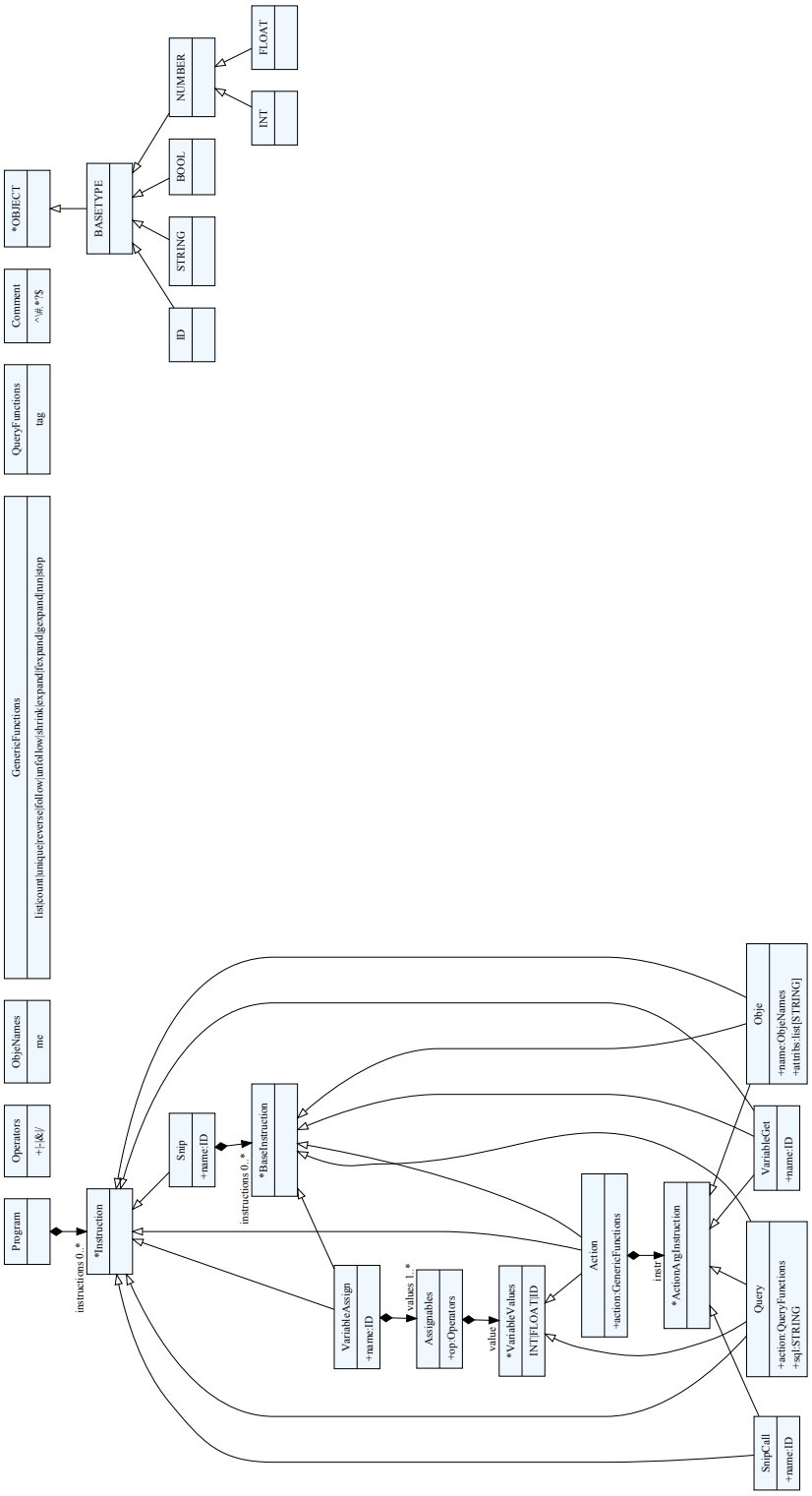
4.1.2 TweetDeck

Website	tweetdeck.twitter.com

A customizable, real-time dashboard for your Twitter profile.

4.2 .ows language structure

This is a graphical representation of the .ows language:



4.3 Queries handbook

Here's a list of common queries and the typical action associated with it.

operation	query	typical action
Leechers with less followers than friends.	`following = 1 and follower = 0 and tff_ratio < 1`	unfollow
Leechers. Usually, you haven't to unfollow them all since some valuable sources (e.g. government, news agencies) will unlikely follow you back.	`following and not follower`	unfollow
After an expand, find users on the perimeter edge whom will likely follow you back. followings / followers ratio should be near, equal or less than 1.	`not following and not follower and tff_ratio <= 1`	follow
Perimeter edge (see section §2.8)	`not following and not follower`	-
Perimeter core (see section §2.8)	`following or follower`	-

4.4 Time limits

JournaKit Followship .ows has some time limits for retrieval and cache designed to avoid to hit Twitter API limits. Some of them are configurable, other are hardcoded.

name	function	value	customizable
WAIT_SECONDS	Minimum time to wait between Actions, see section 2.6.3.	30 seconds	No
SCRIPT_MODE_STATUS_UPDATE_SECONDS	Wait the updated status table while running in script mode, see chapter 3.	60 seconds	No
API_CURSOR_EXPIRE_HOURS	How much time a user record remains inside the perimeter cache without being automatically updated.	6 hours	Yes
STORER_EXPIRE_MINUTES	How much time must pass before the list of core cache users of the current account will be updated, calculated after the application start.	60 minutes	No

4.5 Credits

JournaKit followship .ows is free software **as freedom**.

It cannot be the same without the effort of many free software developers around the world.

Here some of the main components used to build JournaKit followship .ows and their authors.

name	author	license
APScheduler	Alex Grönholm	MIT License
NSIS	NSIS developers	Common Public License 1.0*
Numpy	NumPy developers	Numpy license
Python Prompt Toolkit	Jonathan Slenders	3-Clause BSD License
pynsist	Thomas Kluyver	MIT License*
SQLAlchemy	Michael “Mike” Bayeret al.	MIT License
SQLite	Hwaci	Public Domain
textX	Igor Dejanovic	MIT License
Tweepy	Joshua Roesslein	MIT License
Python	Guido van Rossumet al.	Python Software Foundation License

* Used only to build the installer, not shipped.

Chapter 5

Freedoms

5.1 Software license

5.1.1 Overview

Nobody should be restricted by the software they use. There are four freedoms that every user should have:

- the freedom to use the software for any purpose,
- the freedom to change the software to suit your needs,
- the freedom to share the software with your friends and neighbors, and
- the freedom to share the changes you make.

When a program offers users all of these freedoms, we call it free software.

Developers who write software can release it under the terms of the GNU GPL. When they do, it will be free software and stay free software, no matter who changes or distributes the program. We call this copyleft: the software is copyrighted, but instead of using those rights to restrict users like proprietary software does, we use them to ensure that every user has freedom.

This is an extract from A Quick Guide to GPLv3 by GNU project.

5.1.2 License

JournaKit followship .ows is released under the terms of GNU GPL 3 license.

> Please refer to GNU General Public License on gnu.org or read the **LICENSE.pdf** in the same directory of this document or **LICENSE.txt** on the application root.

5.2 Documentation license

5.2.1 Overview

This document is released under the **Creative Commons Attribution-ShareAlike 3.0 Unported** (CC BY-SA 3.0) license.

This is a human-readable summary of (and not a substitute for) the license[1]. See also section 5.2.2 License.

You are free to:

- **Share** — copy and redistribute the material in any medium or format
- **Adapt** — remix, transform, and build upon the material for any purpose, even commercially.

The licensor cannot revoke these freedoms as long as you follow the license terms.

Under the following terms:

- **Attribution** — You must give appropriate credit[2], provide a link to the license, and indicate if changes were made[3]. You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use.
- **ShareAlike** — If you remix, transform, or build upon the material, you must distribute your contributions under the same license[4] as the original.

[1]This deed highlights only some of the key features and terms of the actual license. It is not a license and has no legal value. You should carefully review all of the terms and conditions of the actual license before using the licensed material.

Creative Commons is not a law firm and does not provide legal services. Distributing, displaying, or linking to this deed or the license that it summarizes does not create a lawyer-client or any other relationship.

[2]If supplied, you must provide the name of the creator and attribution parties, a copyright notice, a license notice, a disclaimer notice, and a link to the material. CC licenses prior to Version 4.0 also require you to provide the title of the material if supplied, and may have other slight differences. More info

[3]In 3.0 and earlier license versions, the indication of changes is only required if you create a derivative. More info

[4]You may also use a license listed as compatible at Marking guide. More info

No additional restrictions — You may not apply legal terms or technological measures[5] that legally restrict others from doing anything the license permits.

Notices

You do not have to comply with the license for elements of the material in the public domain or where your use is permitted by an applicable exception or limitation[6].

No warranties are given. The license may not give you all of the permissions necessary for your intended use. For example, other rights such as publicity, privacy, or moral rights[7] may limit how you use the material.

5.2.2 License

THE WORK (AS DEFINED BELOW) IS PROVIDED UNDER THE TERMS OF THIS CREATIVE COMMONS PUBLIC LICENSE ("CCPL" OR "LICENSE"). THE WORK IS PROTECTED BY COPYRIGHT AND/OR OTHER APPLICABLE LAW. ANY USE OF THE WORK OTHER THAN AS AUTHORIZED UNDER THIS LICENSE OR COPYRIGHT LAW IS PROHIBITED.

BY EXERCISING ANY RIGHTS TO THE WORK PROVIDED HERE, YOU ACCEPT AND AGREE TO BE BOUND BY THE TERMS OF THIS LICENSE. TO THE EXTENT THIS LICENSE MAY BE CONSIDERED TO BE A CONTRACT, THE LICENSOR GRANTS YOU THE RIGHTS CONTAINED HERE IN CONSIDERATION OF YOUR ACCEPTANCE OF SUCH TERMS AND CONDITIONS.

1. Definitions

"Adaptation" means a work based upon the Work, or upon the Work and other pre-existing works, such as a translation, adaptation, derivative work, arrangement of music or other alterations of a literary or artistic work, or phonogram or performance and includes cinematographic adaptations or any other form in which the Work may be recast, transformed, or adapted including in any form recognizably derived from the original, except that a work that constitutes a Collection will not be considered an Adaptation for the purpose of this License. For the avoidance of doubt, where the Work is a musical

[5]The license prohibits application of effective technological measures, defined with reference to Article 11 of the WIPO Copyright Treaty. More info

[6]The rights of users under exceptions and limitations, such as fair use and fair dealing, are not affected by the CC licenses. More info

[7]You may need to get additional permissions before using the material as you intend. More info

work, performance or phonogram, the synchronization of the Work in timed-relation with a moving image ("synching") will be considered an Adaptation for the purpose of this License. "Collection" means a collection of literary or artistic works, such as encyclopedias and anthologies, or performances, phonograms or broadcasts, or other works or subject matter other than works listed in Section 1(f) below, which, by reason of the selection and arrangement of their contents, constitute intellectual creations, in which the Work is included in its entirety in unmodified form along with one or more other contributions, each constituting separate and independent works in themselves, which together are assembled into a collective whole. A work that constitutes a Collection will not be considered an Adaptation (as defined below) for the purposes of this License. "Creative Commons Compatible License" means a license that is listed at https://creativecommons.org/compatiblelicenses that has been approved by Creative Commons as being essentially equivalent to this License, including, at a minimum, because that license: (i) contains terms that have the same purpose, meaning and effect as the License Elements of this License; and, (ii) explicitly permits the relicensing of adaptations of works made available under that license under this License or a Creative Commons jurisdiction license with the same License Elements as this License. "Distribute" means to make available to the public the original and copies of the Work or Adaptation, as appropriate, through sale or other transfer of ownership. "License Elements" means the following high-level license attributes as selected by Licensor and indicated in the title of this License: Attribution, ShareAlike. "Licensor" means the individual, individuals, entity or entities that offer(s) the Work under the terms of this License. "Original Author" means, in the case of a literary or artistic work, the individual, individuals, entity or entities who created the Work or if no individual or entity can be identified, the publisher; and in addition (i) in the case of a performance the actors, singers, musicians, dancers, and other persons who act, sing, deliver, declaim, play in, interpret or otherwise perform literary or artistic works or expressions of folklore; (ii) in the case of a phonogram the producer being the person or legal entity who first fixes the sounds of a performance or other sounds; and, (iii) in the case of broadcasts, the organization that transmits the broadcast. "Work" means the literary and/or artistic work offered under the terms of this License including without limitation any production in the literary, scientific and artistic domain, whatever may be the mode or form of its expression including digital form, such as a book, pamphlet and other writing; a lecture, address,

sermon or other work of the same nature; a dramatic or dramatico-musical work; a choreographic work or entertainment in dumb show; a musical composition with or without words; a cinematographic work to which are assimilated works expressed by a process analogous to cinematography; a work of drawing, painting, architecture, sculpture, engraving or lithography; a photographic work to which are assimilated works expressed by a process analogous to photography; a work of applied art; an illustration, map, plan, sketch or three-dimensional work relative to geography, topography, architecture or science; a performance; a broadcast; a phonogram; a compilation of data to the extent it is protected as a copyrightable work; or a work performed by a variety or circus performer to the extent it is not otherwise considered a literary or artistic work. "You" means an individual or entity exercising rights under this License who has not previously violated the terms of this License with respect to the Work, or who has received express permission from the Licensor to exercise rights under this License despite a previous violation. "Publicly Perform" means to perform public recitations of the Work and to communicate to the public those public recitations, by any means or process, including by wire or wireless means or public digital performances; to make available to the public Works in such a way that members of the public may access these Works from a place and at a place individually chosen by them; to perform the Work to the public by any means or process and the communication to the public of the performances of the Work, including by public digital performance; to broadcast and rebroadcast the Work by any means including signs, sounds or images. "Reproduce" means to make copies of the Work by any means including without limitation by sound or visual recordings and the right of fixation and reproducing fixations of the Work, including storage of a protected performance or phonogram in digital form or other electronic medium.

2. Fair Dealing Rights. Nothing in this License is intended to reduce, limit, or restrict any uses free from copyright or rights arising from limitations or exceptions that are provided for in connection with the copyright protection under copyright law or other applicable laws.

3. License Grant. Subject to the terms and conditions of this License, Licensor hereby grants You a worldwide, royalty-free, non-exclusive, perpetual (for the duration of the applicable copyright) license to exercise the rights in the Work as stated below:

to Reproduce the Work, to incorporate the Work into one or more Collec-

tions, and to Reproduce the Work as incorporated in the Collections; to create and Reproduce Adaptations provided that any such Adaptation, including any translation in any medium, takes reasonable steps to clearly label, demarcate or otherwise identify that changes were made to the original Work. For example, a translation could be marked "The original work was translated from English to Spanish," or a modification could indicate "The original work has been modified."; to Distribute and Publicly Perform the Work including as incorporated in Collections; and, to Distribute and Publicly Perform Adaptations.

For the avoidance of doubt: Non-waivable Compulsory License Schemes. In those jurisdictions in which the right to collect royalties through any statutory or compulsory licensing scheme cannot be waived, the Licensor reserves the exclusive right to collect such royalties for any exercise by You of the rights granted under this License; Waivable Compulsory License Schemes. In those jurisdictions in which the right to collect royalties through any statutory or compulsory licensing scheme can be waived, the Licensor waives the exclusive right to collect such royalties for any exercise by You of the rights granted under this License; and, Voluntary License Schemes. The Licensor waives the right to collect royalties, whether individually or, in the event that the Licensor is a member of a collecting society that administers voluntary licensing schemes, via that society, from any exercise by You of the rights granted under this License.

The above rights may be exercised in all media and formats whether now known or hereafter devised. The above rights include the right to make such modifications as are technically necessary to exercise the rights in other media and formats. Subject to Section 8(f), all rights not expressly granted by Licensor are hereby reserved.

4. Restrictions. The license granted in Section 3 above is expressly made subject to and limited by the following restrictions:

You may Distribute or Publicly Perform the Work only under the terms of this License. You must include a copy of, or the Uniform Resource Identifier (URI) for, this License with every copy of the Work You Distribute or Publicly Perform. You may not offer or impose any terms on the Work that restrict the terms of this License or the ability of the recipient of the Work to exercise the rights granted to that recipient under the terms of the License. You may not sublicense the Work. You must keep intact all notices that refer to this License and to the disclaimer of warranties with every copy of the Work You Distribute or Publicly Perform. When You Distribute or Publicly Perform the Work, You

may not impose any effective technological measures on the Work that restrict the ability of a recipient of the Work from You to exercise the rights granted to that recipient under the terms of the License. This Section 4(a) applies to the Work as incorporated in a Collection, but this does not require the Collection apart from the Work itself to be made subject to the terms of this License. If You create a Collection, upon notice from any Licensor You must, to the extent practicable, remove from the Collection any credit as required by Section 4(c), as requested. If You create an Adaptation, upon notice from any Licensor You must, to the extent practicable, remove from the Adaptation any credit as required by Section 4(c), as requested. You may Distribute or Publicly Perform an Adaptation only under the terms of: (i) this License; (ii) a later version of this License with the same License Elements as this License; (iii) a Creative Commons jurisdiction license (either this or a later license version) that contains the same License Elements as this License (e.g., Attribution-ShareAlike 3.0 US)); (iv) a Creative Commons Compatible License. If you license the Adaptation under one of the licenses mentioned in (iv), you must comply with the terms of that license. If you license the Adaptation under the terms of any of the licenses mentioned in (i), (ii) or (iii) (the "Applicable License"), you must comply with the terms of the Applicable License generally and the following provisions: (I) You must include a copy of, or the URI for, the Applicable License with every copy of each Adaptation You Distribute or Publicly Perform; (II) You may not offer or impose any terms on the Adaptation that restrict the terms of the Applicable License or the ability of the recipient of the Adaptation to exercise the rights granted to that recipient under the terms of the Applicable License; (III) You must keep intact all notices that refer to the Applicable License and to the disclaimer of warranties with every copy of the Work as included in the Adaptation You Distribute or Publicly Perform; (IV) when You Distribute or Publicly Perform the Adaptation, You may not impose any effective technological measures on the Adaptation that restrict the ability of a recipient of the Adaptation from You to exercise the rights granted to that recipient under the terms of the Applicable License. This Section 4(b) applies to the Adaptation as incorporated in a Collection, but this does not require the Collection apart from the Adaptation itself to be made subject to the terms of the Applicable License. If You Distribute, or Publicly Perform the Work or any Adaptations or Collections, You must, unless a request has been made pursuant to Section 4(a), keep intact all copyright notices for the Work and provide, reasonable to the medium or means You are utilizing: (i) the name of the

Original Author (or pseudonym, if applicable) if supplied, and/or if the Original Author and/or Licensor designate another party or parties (e.g., a sponsor institute, publishing entity, journal) for attribution ("Attribution Parties") in Licensor's copyright notice, terms of service or by other reasonable means, the name of such party or parties; (ii) the title of the Work if supplied; (iii) to the extent reasonably practicable, the URI, if any, that Licensor specifies to be associated with the Work, unless such URI does not refer to the copyright notice or licensing information for the Work; and (iv) , consistent with Ssection 3(b), in the case of an Adaptation, a credit identifying the use of the Work in the Adaptation (e.g., "French translation of the Work by Original Author," or "Screenplay based on original Work by Original Author"). The credit required by this Section 4(c) may be implemented in any reasonable manner; provided, however, that in the case of a Adaptation or Collection, at a minimum such credit will appear, if a credit for all contributing authors of the Adaptation or Collection appears, then as part of these credits and in a manner at least as prominent as the credits for the other contributing authors. For the avoidance of doubt, You may only use the credit required by this Section for the purpose of attribution in the manner set out above and, by exercising Your rights under this License, You may not implicitly or explicitly assert or imply any connection with, sponsorship or endorsement by the Original Author, Licensor and/or Attribution Parties, as appropriate, of You or Your use of the Work, without the separate, express prior written permission of the Original Author, Licensor and/or Attribution Parties. Except as otherwise agreed in writing by the Licensor or as may be otherwise permitted by applicable law, if You Reproduce, Distribute or Publicly Perform the Work either by itself or as part of any Adaptations or Collections, You must not distort, mutilate, modify or take other derogatory action in relation to the Work which would be prejudicial to the Original Author's honor or reputation. Licensor agrees that in those jurisdictions (e.g. Japan), in which any exercise of the right granted in Section 3(b) of this License (the right to make Adaptations) would be deemed to be a distortion, mutilation, modification or other derogatory action prejudicial to the Original Author's honor and reputation, the Licensor will waive or not assert, as appropriate, this Section, to the fullest extent permitted by the applicable national law, to enable You to reasonably exercise Your right under Section 3(b) of this License (right to make Adaptations) but not otherwise.

5. Representations, Warranties and Disclaimer

UNLESS OTHERWISE MUTUALLY AGREED TO BY THE PARTIES IN WRIT-

ING, LICENSOR OFFERS THE WORK AS-IS AND MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND CONCERNING THE WORK, EXPRESS, IMPLIED, STATUTORY OR OTHERWISE, INCLUDING, WITHOUT LIMITATION, WARRANTIES OF TITLE, MERCHANTIBILITY, FITNESS FOR A PARTICULAR PURPOSE, NONINFRINGEMENT, OR THE ABSENCE OF LATENT OR OTHER DEFECTS, ACCURACY, OR THE PRESENCE OF ABSENCE OF ERRORS, WHETHER OR NOT DISCOVERABLE. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OF IMPLIED WARRANTIES, SO SUCH EXCLUSION MAY NOT APPLY TO YOU.

6. Limitation on Liability. EXCEPT TO THE EXTENT REQUIRED BY APPLICABLE LAW, IN NO EVENT WILL LICENSOR BE LIABLE TO YOU ON ANY LEGAL THEORY FOR ANY SPECIAL, INCIDENTAL, CONSEQUENTIAL, PUNITIVE OR EXEMPLARY DAMAGES ARISING OUT OF THIS LICENSE OR THE USE OF THE WORK, EVEN IF LICENSOR HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. Termination

This License and the rights granted hereunder will terminate automatically upon any breach by You of the terms of this License. Individuals or entities who have received Adaptations or Collections from You under this License, however, will not have their licenses terminated provided such individuals or entities remain in full compliance with those licenses. Sections 1, 2, 5, 6, 7, and 8 will survive any termination of this License. Subject to the above terms and conditions, the license granted here is perpetual (for the duration of the applicable copyright in the Work). Notwithstanding the above, Licensor reserves the right to release the Work under different license terms or to stop distributing the Work at any time; provided, however that any such election will not serve to withdraw this License (or any other license that has been, or is required to be, granted under the terms of this License), and this License will continue in full force and effect unless terminated as stated above.

8. Miscellaneous

Each time You Distribute or Publicly Perform the Work or a Collection, the Licensor offers to the recipient a license to the Work on the same terms and conditions as the license granted to You under this License. Each time You Distribute or Publicly Perform an Adaptation, Licensor offers to the recipient a license to the original Work on the same terms and conditions as the license granted to You under this License. If any provision of this License is invalid or unenforceable under applicable law, it shall not affect the validity or en-

forceability of the remainder of the terms of this License, and without further action by the parties to this agreement, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable. No term or provision of this License shall be deemed waived and no breach consented to unless such waiver or consent shall be in writing and signed by the party to be charged with such waiver or consent. This License constitutes the entire agreement between the parties with respect to the Work licensed here. There are no understandings, agreements or representations with respect to the Work not specified here. Licensor shall not be bound by any additional provisions that may appear in any communication from You. This License may not be modified without the mutual written agreement of the Licensor and You. The rights granted under, and the subject matter referenced, in this License were drafted utilizing the terminology of the Berne Convention for the Protection of Literary and Artistic Works (as amended on September 28, 1979), the Rome Convention of 1961, the WIPO Copyright Treaty of 1996, the WIPO Performances and Phonograms Treaty of 1996 and the Universal Copyright Convention (as revised on July 24, 1971). These rights and subject matter take effect in the relevant jurisdiction in which the License terms are sought to be enforced according to the corresponding provisions of the implementation of those treaty provisions in the applicable national law. If the standard suite of rights granted under applicable copyright law includes additional rights not granted under this License, such additional rights are deemed to be included in the License; this License is not intended to restrict the license of any rights under applicable law.

This text was extracted from the Creative Commons official website https://creativecommons.org/licenses/by-sa/3.0/legalcode.

Digitally printed on September 17, 2017 using Material icons,
edited on LyX, XeTeX and MiKTeX.