

# Fighting the Good Fight with Taskwarrior and the Kusarigama



Yanick Champoux

[yanick.champoux@iinteractive.com](mailto:yanick.champoux@iinteractive.com) / [@yenzie](https://twitter.com/yenzie)

# BATTLEPLAN

- Unto the warpath
- Taskwarrior - *get more efficient with tasks..*
- Customize via the Kusarigama - *...then waste it all in yak shaving*

# MOTIVATION

My task management skills suck

Like, lots.

# GETTING THINGS DONE

## Getting Things Done

the art of stress-free  
productivity

from the New York Times bestselling author

David Allen



# TWO TAKEAWAYS

- Dump all your tasks somewhere
- Keep things simple

# CUE IN QUEST FOR PERFECT TOOL



# GUI

- Open browser, wiggle mouse, lots of work
- Always close, never cigar.
- In the cloud == far away



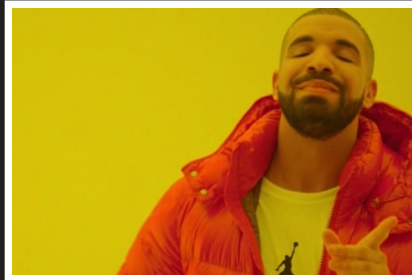


<https://github.com/bestpractical/hiveminder>



# CLI

- I live in terminals
- Allow for piping and grepping and UNIXing
- Scriptable
- Autocomplete



# ENTER TASKWARRIOR

- CLI tool task
- Written in C++
- Flexible, extensive set of options
- Customizable

# LIKE VI

start small, learning slope just keep goin' up

# **BASICS: CREATE, LIST, AND CLOSE**

# CREATE

```
$ task add Write taskwarrior prez for toronto.pm
```

```
Created task 1.
```

# SEE

```
$ task list
```

ID	Age	Description	Urg
1	5s	Write taskwarrior prez for toronto.pm	0

```
1 task
```

# CLOSE

```
$ task done 1
```

```
Completed task 1 'Write taskwarrior prez for toronto.pm'.  
Completed 1 task.
```

# ABANDON

```
$ task delete 1
```

```
Delete task 1 'Write taskwarrior prez for toronto.pm'? (yes/no  
Deleting task 1 'Write taskwarrior prez for toronto.pm'.  
Deleted 1 task.
```



# UNDO THE PAST

```
$ task undo
```

```
The last modification was made 2/19/2018
```

	Prior Values	Current Values
description	Write taskwarrior toronto.pm	Write taskwarrior prez for toronto.pm
entry modified	2018-02-19 2018-02-19	2018-02-19 2018-02-19
status	pending	deleted
uuid	4057ec52-2a73-4776 -f72f402581a9	4057ec52-2a73-4776-9dbc

# ADDING METADATA

# TAGS

```
$ task 1 append +talk +toronto
```

```
Appending to task 1 'Write taskwarrior prez for toronto.pm'.  
Appended 1 task.
```

# PROJECT

```
$ task 1 append project:talks
```

```
Appending to task 1 'Write taskwarrior prez for toronto.pm'.
```

```
Appended 1 task.
```

```
Project 'talks' is 0% complete (1 task remaining).
```

# PRIORITIES

```
$ task 1 append priority:H
```

```
Appending to task 1 'Write taskwarrior prez for toronto.pm'.
```

```
Appended 1 task.
```

```
Project 'talks' is 0% complete (1 task remaining).
```

# DEADLINES

```
$ task 1 append due:2018-02-22
```

```
Appending to task 1 'Write taskwarrior talk for toronto.pm'.
```

```
Appended 1 task.
```

```
Project 'talks' is 0% complete (1 task remaining).
```

# MANY WAYS TO SPECIFY TIMES

```
due:4d  
due:thursday  
due:22nd  
due:eow  
due:eom-43h
```

# NOT RIGHT NOW

```
$ task 1 append wait:21st
```

```
Appending to task 1 'Write taskwarrior talk for toronto.pm'.
```

```
Appended 1 task.
```

```
Project 'talks' is 0% complete (1 task remaining).
```



# DEPENDENCIES

```
$ task add "email slides to group" depends:1
```

```
Created task 4.
```

**QUERYING**

# INFO

```
$ task 1 info
```

Name	Value
ID	1
Description	Write taskwarrior talk for toronto.pm
Status	Waiting
Project	talks
Entered	2018-02-19 11:50:24 (18min)
Waiting until	2018-02-21 00:00:00
Due	2018-02-22 00:00:00
Last modified	2018-02-19 12:09:03 (5s)
Tags	talk toronto
Virtual tags	DUE MONTH TAGGED UDA UNBLOCKED WAITING WEEK YEAR PROJECT PRIORITY
UUID	4057ec52-2a73-4776-9dbc-f72f402581a9

# URGENCY

project	1	*	1	=	1
waiting	1	*	-3	=	-3
tags	0.9	*	1	=	0.9
due	0.638	*	1e+0	=	63.8
UDA priority.H	1	*	30	=	30
					-----
					92.73

# LIST

```
$ task list
```

ID	Age	P	Project	Tags	Due	Desc	Urg
1	2h	H	talks	talk toronto	2d	Write...	96.1
4	1m	D				ema..	-5
2	34s			shopping		buy t...	0.8
3	2s		TW-K			relea...	1

4 tasks

# NEXT

```
$ task next
```

ID	Age	P	Project	Tags	Due	Desc	Urg
1	2h	H	talks	talk toronto	2d	Write...	96.1
2	34s			shopping		buy t...	0.8
3	2s		TW-K			relea...	1

3 tasks

# FANCY REPORTS

```
task burndown.daily  
task calendar  
task history  
task summary  
task stats
```

# CONTEXT

```
$ task context define work +ii or \( project:foo and due.befor  
$ task context work  
...  
$ task context none
```



**CUSTOMIZATIONS**

# . TASKS/PENDING . DATA EASY TO READ

```
[description:"Write t..." due:"1519275600" ... ]
```

# BUT YOU DON'T NEED TO!

```
task export  
task import
```

```
$ task export | jq '.[].urgency' | histogram.py
```

```
# NumSamples = 4; Min = -5.00; Max = 104.21
```

```
# Mean = 25.251500; Variance = 2083.744357; SD = 45.648049; Me
```

```
# each ■ represents a count of 1
```

```
-5.0000 - 5.9206 [ 3]: ■■■
```

```
5.9206 - 16.8412 [ 0]:
```

```
16.8412 - 27.7618 [ 0]:
```

```
27.7618 - 38.6824 [ 0]:
```

```
38.6824 - 49.6030 [ 0]:
```

```
49.6030 - 60.5236 [ 0]:
```

```
60.5236 - 71.4442 [ 0]:
```

```
71.4442 - 82.3648 [ 0]:
```

```
82.3648 - 93.2854 [ 0]:
```

```
93.2854 - 104.2060 [ 1]: ■
```

# HOOKS

Just like git

```
~/ .task/hooks/on-launch*  
~/ .task/hooks/on-add*  
~/ .task/hooks/on-modify*  
~/ .task/hooks/on-exit*
```

**TASKWARRIOR::KUSARIGAMA**

# 2 TYPES OF PLUGINS

- Behavior tweaker
- New task <command>

# ONLY ONE ACTIVE TASK

```
task start 1  
task start 2  
task stop 1
```



```
package Taskwarrior::Kusarigama::Plugin::SingleActive;

use Moo;

extends 'Taskwarrior::Kusarigama::Hook';
with    'Taskwarrior::Kusarigama::Hook::OnLaunch';

sub on_launch($self) {
    $self->run_task->stop( [ '+ACTIVE', '+PENDING' ] )
    if $self->command eq 'start';
};
```

# PROGRESS

```
$ task add read gt book goal:323
```

```
$ task 650 list
```

ID	Age	Description	Urg
650	45s	read gtd book (0/323)	0

```
$ task 650 progress 100
```

```
650 ===== 100/323
```

```
package Taskwarrior::Kusarigama::Plugin::Command::Progress;

use Moo;

extends 'Taskwarrior::Kusarigama::Plugin';

with qw/
    Taskwarrior::Kusarigama::Hook::OnCommand
    Taskwarrior::Kusarigama::Hook::OnAdd
    Taskwarrior::Kusarigama::Hook::OnModify
/;
```

```
has custom_uda => (  
  is => 'ro',  
  default => sub{ +{  
    goal      => 'quantifiable goal',  
    progress => "where we're at",  
  }},  
);
```

```
sub on_add { goto &on_modify; }

sub on_modify ( $self, $task ) {
    my $goal = $task->{goal} or return;

    my $progress = $task->{progress};

    $task->{description} =~ s#\(\d+\/\d+\)(.*?)$#$1#;
    $task->{description} .= sprintf ' (%d/%d)', $progress, $goal;

    return $task;
}
```

```
# task gtd progress +36 good chapter

sub on_command($self) {
    # only grab goal'ed tasks
    my @tasks = $self->run_task->export(
        $self->pre_command_args, { 'goal.any' => '' }
    ) or die "no tasks found\n";

    my $note = $self->post_command_args
        =~ s/\s*(=?)(-?\d*)\s*//r;
```

```
for my $task ( @tasks ) {
    my $progress = $1?$2:($2||1) + $task->{progress};
    my $goal = $task->{goal};

    my $ratio = $progress / $goal;

    print $task->{id}, ' ',
          '=' x ( 20 * $ratio ),
          '-' x ( 20 * ( 1 - $ratio ) ),
          ' ', $progress, '/', $goal, "\n";
}
```

```
$task->{progress} = $progress;  
  
$task->add_annotation( $note ) if $note;  
  
$task->{status} = 'completed'  
    if $progress >= $task->{goal}  
  
$task->save;  
    }  
};
```



# DEPENDENCIES

```
$ task add do this  
Created task 14  
$ task add do that depends:14  
Created task 15  
$ task add do the prerequisite thing  
Created task 16  
$ task 15 append depends:16
```

```
$ task add do this  
$ task and-after do that  
$ task but-before do the prerequisite thing
```

```
$ task gtd and-after write review
```

```
package Taskwarrior::Kusarigama::Plugin::Command::AndAfter;
use Moo;
extends 'Taskwarrior::Kusarigama::Plugin';
with 'Taskwarrior::Kusarigama::Hook::OnCommand';

sub on_command($self) {

    my $dep = $self->pre_command_args ||
        ( $self->tw->run_task->export('+LATEST') )[0]->{uuid};

    $self->tw->run_task->add(
        $self->post_command_args, { depends => $dep }
    );

    say "task added";
}
```

```
$ task gtd but-before go buy gtd book
```

```
package Taskwarrrior::Kusarigama::Plugin::Command::ButBefore;

sub on_command {
    my $self = shift;

    my $prev = $self->pre_command_args
        || ($self->tw->run_task->export( '+LATEST' ))[0]->{uuid};

    $self->tw->run_task->add( $self->post_command_args );

    my( $new_task ) = $self->tw->run_task->export( '+LATEST' );
    say "added new task ", $new_task->{id};

    $self->tw->run_task->append( $prev, { depends => $new_task->
    say "dependencies added";
```

# LEARN MORE

- [Paul Fenwick at a conf near you](#)
- [Brian Wisti's blog](#)