



# TASKWARRIOR

<http://taskwarrior.org>  
Federico Hernandez  
[f@ederi.co](mailto:f@ederi.co)

Taskwarrior software

todo list

~~todo list~~

task management

```
$ task add Buy milk  
Created task 1.
```

```
$ task add Buy eggs  
Created task 2.
```

```
$ task add Bake cake  
Created task 3.
```

```
$ task list
```

ID	Description
----	-------------

1	Buy milk
2	Buy eggs
3	Bake cake

```
3 tasks.
```

```
$ task 1 done  
$ task 2 done  
$ task list
```

```
ID Description  
--  
1  Bake cake
```

```
1 task.
```

```
$ cat .task/pending.data  
[description:"Buy milk" entry:"1442349664" modified:"1442349664"  
status:"pending" uuid:"d385005e-e00e-49ca-860d-886dfaaeab7e"]  
[description:"Buy eggs" entry:"1442349671" modified:"1442349671"  
status:"pending" uuid:"9f5bfe76-2ca1-4f15-accc-437dde42d54e"]  
[description:"Bake cake" entry:"1442349679" modified:"1442349679"  
status:"pending" uuid:"68272ee9-8d5d-463f-9cb3-512ebab58454"]
```



command line

text based

methodology-neutral

grows along

extensible

MIT license

install from source  
or use the packages

Debian, Ubuntu, Fedora, RHEL,  
OpenSUSE, Archlinux, Slackware,  
Gentoo, Sabayon, FreeBSD,  
OpenBSD, Mac OS X, Cygwin



<http://taskwarrior.org/docs/>

task help

man task

tutorial movies

## Basic Usage

[Command Line Syntax](#)[Best Practices](#)[Example Commands](#)[All the Commands](#)[Searching](#)[Reports](#)[Filters](#)[Priority](#) **2.4.3**[Tags & Virtual Tags](#)[Date & Time](#) **2.4.0**[Named Dates](#) **2.4.0**[Duration Values](#) **2.4.0**

### Configuring Taskwarrior

[Configuration](#)[Verbosity](#)[Color Themes](#)[Terminology - a glossary of terms](#)

## Advanced Topics

[Urgency](#)[ID Numbers](#)[Context](#) **2.4.2**[How Recurrence Works](#)[User Defined Attributes \(UDAs\)](#)[External Scripts](#)[Escaping Command Line Characters](#)[DOM - Document Object Model](#)[JSON Import/Export Format](#)[Deprecated Features](#)

Taskserver

Tools

Frontends

Libraries

Hook scripts

Extensions

Taskwarrior project

```
$ git log|tail -5
commit 837527c09989aa093d4c8994c04cadcdbe1491ca
Author: Paul Beckingham <paul@beckingham.net>
Date:    Sat Apr 19 21:54:30 2008 -0400
```

- Initial commit.

```
$ git tag|grep v|wc -l
42
```

```
$ whois taskwarrior.org
Domain Name:TASKWARRIOR.ORG
Creation Date: 2009-05-22T16:19:35Z
```

Distributed team

IRC, Slack, Mail



Github

~~Github~~  
Gitorious

~~Github~~  
~~Gitorious~~  
Stash

Redmine

~~Redmine~~  
Static & JIRA

CI  
flod



# Grid Testing

anomaly	master
clog	1.2.0 1.1.0
flod	master
kronisk	1.0.0
libexpr	master
task	<a href="#">2.5.0</a> <a href="#">2.4.5</a> <a href="#">2.4.4</a> <a href="#">2.4.3</a> <a href="#">2.4.2</a> <a href="#">2.4.1</a> <a href="#">2.4.0</a> <a href="#">2.3.1</a>
taskd	1.2.0 1.1.0
tasksh	1.1.0 1.0.0 0.9.0



## Grid Testing - task.git 2.5.0

	Arch 2014.05 64 (GBF)	Cygwin 1.7.32 32 (GBF)	Debian 7 64 (GBF)	Fedora 20 64 (GBF)	FreeBSD 10 (GBF)	OSX 10.10 64 (GBF)	Ubuntu 14.04 64 (GBF)
2015-09-15 16:14:18 <a href="#">605f8020</a> Wilhelm Schuermann	Test: 100.00%, 8360/8360, 12 skipped, 247 files Time: 132.8 sec	Test: 99.99%, 8359/8360, 23 skipped, 247 files ✖ Test failures Time: 393.0 sec	Test: 100.00%, 8360/8360, 12 skipped, 247 files Time: 85.3 sec	Test: 100.00%, 8360/8360, 20 skipped, 247 files Time: 124.8 sec	Test: 100.00%, 8360/8360, 20 skipped, 247 files Time: 167.9 sec	Test: 100.00%, 8360/8360, 11 skipped, 247 files Time: 77.2 sec	Test: 100.00%, 8360/8360, 12 skipped, 247 files Time: 243.1 sec
2015-09-15 11:15:16 <a href="#">2690d9bd</a> Scott Kostyshak	Test: 100.00%, 8360/8360, 12 skipped, 247 files Time: 132.6 sec	Test: 0.00%, 0/0 ✖ Test failures Time: 94.6 sec	Test: 100.00%, 8360/8360, 12 skipped, 247 files Time: 81.3 sec	Test: 100.00%, 8360/8360, 20 skipped, 247 files Time: 124.4 sec	Test: 100.00%, 8360/8360, 20 skipped, 247 files Time: 167.0 sec	Test: 100.00%, 8360/8360, 11 skipped, 247 files Time: 76.4 sec	Test: 100.00%, 8360/8360, 12 skipped, 247 files Time: 246.2 sec
2015-09-15 10:27:25 <a href="#">e478396a</a> Wilhelm Schuermann	Test: 100.00%, 8360/8360, 12 skipped, 247 files Time: 126.7 sec	Test: 100.00%, 8360/8360, 23 skipped, 247 files Time: 389.2 sec	Test: 99.99%, 8359/8360, 12 skipped, 247 files ✖ Test failures Time: 81.1 sec	Test: 100.00%, 8360/8360, 20 skipped, 247 files Time: 121.3 sec	Test: 100.00%, 8360/8360, 20 skipped, 247 files Time: 172.0 sec	Test: 100.00%, 8360/8360, 11 skipped, 247 files Time: 88.3 sec	Test: 100.00%, 8360/8360, 12 skipped, 247 files Time: 234.5 sec
2015-09-14 11:55:51 <a href="#">11009fd3</a> Paul Beckingham	Test: 100.00%, 8360/8360, 12 skipped, 247 files Time: 134.2 sec	Test: 100.00%, 8360/8360, 23 skipped, 247 files Time: 390.2 sec	Test: 100.00%, 8360/8360, 12 skipped, 247 files Time: 80.6 sec	Test: 99.99%, 8359/8360, 20 skipped, 247 files ✖ Test failures Time: 129.0 sec	Test: 100.00%, 8360/8360, 20 skipped, 247 files Time: 166.3 sec	Test: 100.00%, 8360/8360, 11 skipped, 247 files Time: 82.5 sec	Test: 100.00%, 8360/8360, 12 skipped, 247 files Time: 248.6 sec
2015-09-14 02:27:24 <a href="#">aea4e8cb</a> Paul	Test: 100.00%, 8360/8360, 12	Test: 100.00%, 8360/8360, 23	Test: 100.00%, 8360/8360, 12	Test: 100.00%, 8360/8360, 20	Test: 100.00%, 8360/8360, 20	Test: 100.00%, 8360/8360, 11	Test: 100.00%, 8360/8360, 12



# Sponsoring

Infrastructure - Rackspace

Infrastructure - DigitalOcean

DevTools - Atlassian

Monitoring - Pingdom

Statuspage - StatusPage.io

# Sponsoring Donations (Paypal)

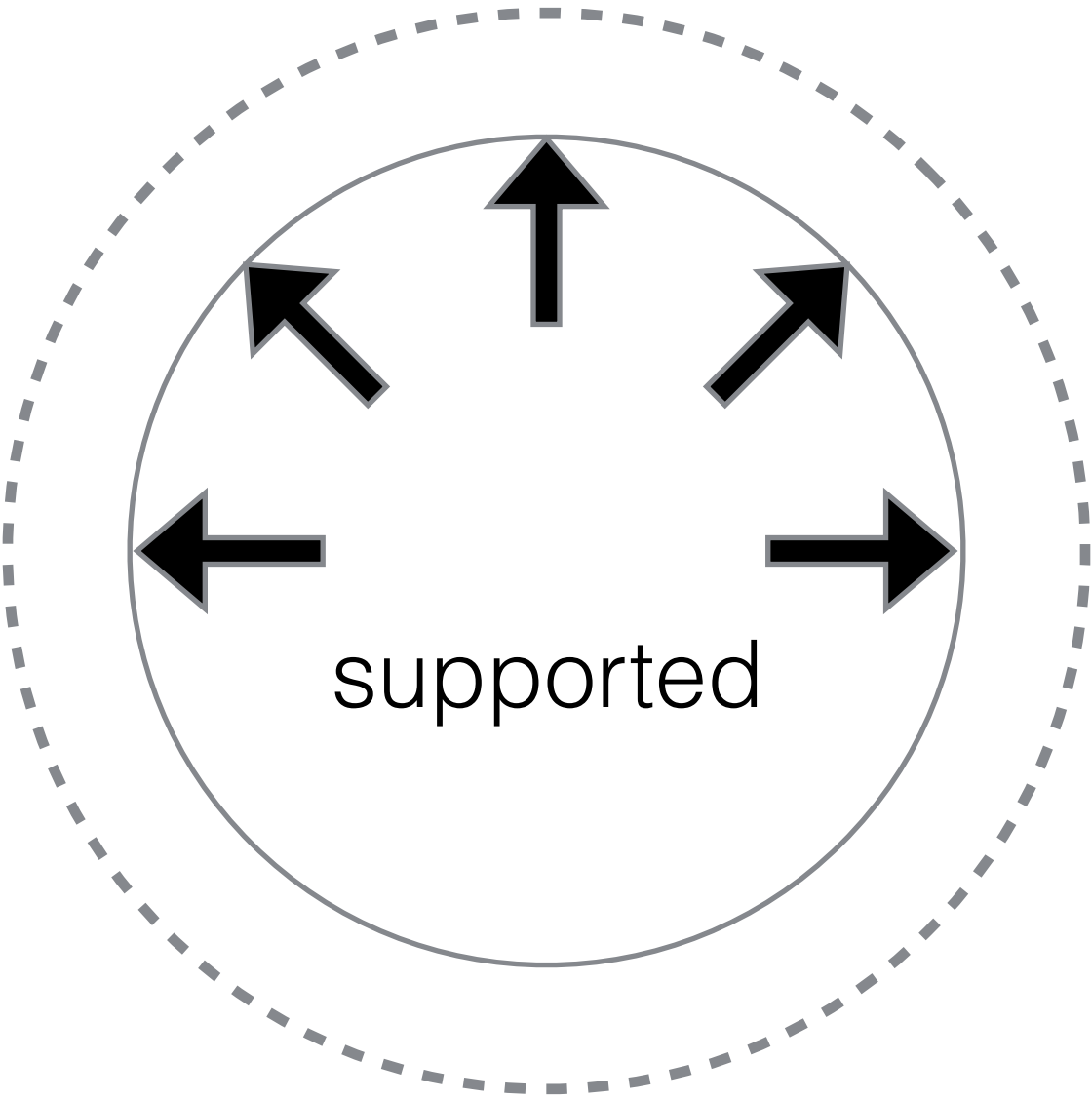
# Project stories

~~Project stories~~  
War stories

Feature requests

FEATURES

not supported



not supported

not supported

Every change will ruin  
someone's day.

If a feature works well,  
you'll never hear about it  
again.



People only get excited  
about something a project  
doesn't quite yet support.

Users will disguise feature requests as bugs.

Many people will throw  
features requests at you,  
just to show that they are  
clever.

They don't want that feature.

Make them:  
sign up, log in, fill out a  
form for a feature  
request.

It's a laziness filter.

Users usually only think  
incrementally.

People threaten to not  
use software if it lacks  
their favourite feature.

User: Does \$PRODUCT support \$TECHNOLOGY?

Team: No, it doesn't.

User: I can't see myself using \$PRODUCT unless it supports \$TECHNOLOGY.

Team: Sorry, there are no plans to add that.

User: Well, you're going to lose a customer.

Team: This is open source, there are no customers.

Documentation



Users will seek you out  
online to directly ask a  
question that is answered  
two clicks from the front  
page of a website.

A looping, animated GIF  
will be watched over and  
over. A paragraph of text  
will not be read.

Man pages are too  
densely crammed with  
information, and too  
long, for modern  
Humans.

Support questions

“What have you tried so far?” is the best question for finding time-wasters.

User: How do I do X?

Team: What have you tried so far?

User: Nothing, I was just wondering...

## Developers

[Hooks API](#) **2.4.0**

[Hooks v2 API](#) **2.4.3**

[Hook Author's Guide](#) **2.4.0**

[How To Build Taskwarrior](#)

[Contributing to the Project](#)

[Coding Style](#)

[Branching Model](#)

[3rd Party Application Guidelines](#)

[Design Documents \(RFCs\)](#)

## Feedback

[How to Report a Bug](#)

[How to Request a Feature](#)

[How to ask a question](#)

# How to Request a Feature

We encourage you to request features.

You can submit a feature request using our [issue tracking system](#). Simply create an account, create an issue, and set the type to either 'improvement' or 'feature', at your discretion. But there is a little more to it than that. After all, wouldn't you like to see your request implemented?

## What Makes a Good Feature Request?

Requesting a feature is easy, but you must remember that it doesn't mean it will get done. It may be a simple or vague request. It may be a complex request that needs some analysis, and we will want to discuss it first. Sometimes we even go so far as to write up an RFC before doing any work. Here are some tips for making a good feature request:

- Describe a compelling feature, and make us really *want* it. What we find most convincing is a statement that illustrates what having the new feature would mean, or would enable. What new things could you do if you had this feature? How would you be more effective at managing tasks with this feature?
- Describe how the feature might interact with existing features. We prefer it when a new feature is more than simply storing a new piece of data, for later retrieval - we have [UDAs](#) for that.
- Describe any new reports that may be needed.
- Describe who might benefit from this feature.
- Describe a feature that doesn't impose a cost on users that choose not to use it. We want the basic usage (add, list, done) to remain simple and effective.

One thing to bear in mind is that we have a long term goal of simplifying the way we manage tasks, simplifying Taskwarrior, and improving consistency.



## Things to Avoid

- If your suggestion affects basic usage (add, list, done) then it will not be implemented.
- Don't bother inventing syntax, or coming up with an unused character on your keyboard. We would much rather you put your effort into thinking about the feature and the benefits. Syntax is our problem. Focus on making others wonder how they could possibly live without the feature.
- Don't bother suggesting features that lead Taskwarrior away from its intended purpose - managing tasks. We have no intention of adding time-tracking to the Taskwarrior core.
- If you have a suggestion that can be implemented as a hook script, or add-on, please do that instead. If the result is compelling enough, we may want to migrate the functionality in-core, but it's not likely.
- Don't tell us how easy something would be, or how quickly you could add it. Ease of implementation has nothing to do with whether an idea is good. Instead, convince everyone that the idea is great.

### A Good Example

I think dependencies would be a great feature, because it would help me organize tasks and know which ones I need to work on first. It would also tell me which ones I cannot work on now. Tasks that are depended upon could affect the 'next' report, because those tasks need to be done first. Ideally a task could depend on multiple other tasks. I would need to both add and remove a dependency. Dependencies are a commonly understood concept, and so would be useful to many users.

### Another Good Example

I should be allowed to use 'status' field in a report. It doesn't make sense to withhold just this one field. For the sake of consistency, please add this as a reportable, sortable field.

Miscellaneous

People will pick arguments with you about incidental things, such as your choice of bug database, branch names, version numbers, text editor used, PDF writer and so on.

User: Why are you using \$BUG\_DB, when you could be using \$ALTERNATIVE?

Team: We like it.

User: But \$ALTERNATIVE is open source!

Team: Our \$PRODUCT is open source. That's the part we care about and focus on.

SEO consultants aren't very good at searching the web, and identifying all-volunteer open-source projects.

SEO: I can help you double your revenue!

Team: We're an open source, volunteer-only project. Oh look, I just doubled zero myself.

<http://creativecommons.org/licenses/by/4.0/>



#### 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, provide a link to the license, and indicate if changes were made. You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use.

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