

## FAQ - highlights from years of Q&A.

Have a question that's not here? [Ask us...](#)

### Taskwarrior

- Q: How do I implement Pomodoro?
- Q: How do I remove an attribute?
- Q: How do I list tasks that have no project?
- Q: How do I filter tasks containing words that are also commands?
- Q: How do I hide the change history when using the `info` report?
- Q: How do I list tasks scheduled today?
- Q: How do I bring back a task that I completed or deleted by mistake?
- Q: How do I improve output readability?
- Q: How do I set the wait date to be 2 days before the due date?
- Q: How do I create a traditional GTD Next Actions list?
- Q: Why have attribute modifiers (`urgency,over:10`) rather than the more readable and algebraic form (`urgency>10`)?
- Q: How do I pipe a UUID into another task command?
- Q: How do I get the UUID of the last task added?
- Q: How do I change many due dates using one command?
- Q: How do I search for tasks?
- Q: How do I create a task that recurs on more than one day of the week?
- Q: How can I include the time spent on a task in a custom report?
- Q: How do I hide tasks that have unfinished dependencies?
- Q: How do I record a task that is already completed?
- Q: How do I include the recurrence frequency in a report?
- Q: How do I solve "Taskwarrior was built without GnuTLS support. Sync is not available.?"
- Q: How do I get color working?
- Q: How do I fix this `libgnutls-deb0.so.28` error?

### Timewarrior

- Q: How do I control Taskwarrior through Timewarrior?

### Tasksh

- Q: How do I use arrow keys in tasksh?
- Q: How do I repeat the last command?

### Taskserver

- Q: How do I setup Taskserver to use LetsEncrypt certs?
- Q: Which version of GnuTLS should I use?

**Q: How do I implement Pomodoro?**

Taskwarrior currently doesn't support the pomodoro technique, not because it doesn't have all the necessary features, but because it has no way to notify the user if a given point in time has arrived. Remember: most of the time, Taskwarrior isn't running.

You can however use the pomodoro timer or any other timer software (KTeaTimer, Gnome's Tea Timer, Gnome's pomodoro shell extension, pyStopWatch, or any of the many timers out there), together with taskwarrior to manage and prioritize your ToDo list.

**Q: How do I remove an attribute?**

Modify the attribute and give it a blank value:

```
$ task 16 modify due:
```

Taskwarrior doesn't store blank values, so it deletes the attribute.

**Q: How do I list tasks that have no project?**

There are a few ways:

```
$ task project:none: list
$ task project: list
$ task project != '' list
$ task -PROJECT list
```

The last example makes use of a virtual tag designed for this use case, and is the preferred mechanism.

To list tasks that do have a project assigned, you can use these alternates:

```
$ task project.any: list
$ task project.not: list
$ task project == '' list
$ task +PROJECT list
```

**Q: How do I filter tasks containing words that are also commands?**

I have tasks with descriptions such as:

```
ID Description
1 Print my report and give it to Bob.
2 Collect more info on daily rates of food waste.
```

When I try to filter them like this, it runs the command instead:

```
$ task report
...
$ task info
...
```

There is an operator `--` intended for this purpose:

```
$ task -- report
...
$ task -- info
...
```

The `--` operator is a directive that tells Taskwarrior to just treat all subsequent arguments as plain words. It is also a good way to add a description that might contain things like `dep:`, `proj:`, tags or modifiers.

**Q: How do I hide the change history when using the `info` report?**

You need to turn off journaling, like this:

```
$ task config journal.info off
```

See 'man taskrc' for more details.

**Q: How do I list tasks scheduled today?**

To filter tasks within a date range:

```
$ task scheduled \> today and scheduled \< tomorrow list
```

Note how filter operators need to be escaped, otherwise your shell will think it knows what to do. Alternately, do this:

```
$ task scheduled.after:today and scheduled.before:tomorrow list
```

**Q: How do I bring back a task that I completed or deleted by mistake?**

Simply changing the `status` of the task to `pending` is enough, Taskwarrior takes care of the rest.

```
$ task <uuid> modify status:pending
```

You can obtain the `<uuid>` in several ways. Here is how to see the 10 most recently completed or deleted tasks, with the most recent shown first:

```
$ task limit:10 \ ( status:completed or status:deleted \) rc.report.all.sort:end- all
```

**Q: How do I improve output readability?**

You can use line breaks which allow blank lines between sort keys. See [Report](#) for details. You can also use alternating line colors setting, which helps visually separate the tasks.

```
$ task show color.alternate
...
$ task color legend
...
```

Most color themes already support this.

**Q: How do I set the wait date to be 2 days before the due date?**

Taskwarrior can do date math, so use this:

```
$ task add ... due:eom wait:due-2days
```

That `due-2days` value is evaluated by Taskwarrior, using the value you specified for the due date and subtracting two days. You can also verify that date using the `calc` command:

```
$ task calc eom-2days
2018-01-30T00:00:00
```

**Q: How do I use arrow keys in tasksh?**

Tasksh supports [readLine](#), so if you are building from source, install the dev readline library first.

**Q: How do I control Taskwarrior through Timewarrior?**

For example, if I start tracking in Timewarrior, how do I make the task active also?

Taskwarrior uses Timewarrior to track time, via a Taskwarrior hook. You just start/stop tasks using Taskwarrior, and Timewarrior will start/stop tracking.

If you start/stop in Timewarrior, it does not control Taskwarrior.

**Q: How do I setup Taskserver to use LetsEncrypt certs?**

On the Taskserver server:

- Generate a self-signed CA (use `taskd/pki/generate.ca`)
- Get a copy of your domain's TLS certificates from Lets Encrypt
- Configure Taskserver:

```
ca.cert=ca.cert.pem
server.cert=example.com.crt
server.key=example.com.key
server=0.0.0.0:53589
```

On each client:

- Get a self-signed client keypair from the Taskserver (`taskd/pki/generate.client`)
- Get a copy of the DST Root CA X3
- Configure Taskwarrior:

```
taskd.certificate=client.cert.pem
taskd.key=client.key.pem
taskd.ca=DST_ROOT_CA_X3.crt
taskd.server=example.com:53589
taskd.credentials=Group/Name/UUID
```

Client certificates and user credentials are created per the docs, as normal. Client certificates can be revoked by generating a CRL file and pointing server.crl to it in the Taskserver config. See the bottom of [taskd/pki/generate.crl](#) for invocations. For personal installations, it is easier just to regenerate the self-signed CA and distribute new client certs.

**Q: How do I create a traditional GTD Next Actions list?**

I am looking to create a custom report that lists all pending tasks without a project and the single most urgent from each project.

The report you describe is not possible as a single custom report because it combines multiple independent filters. It could be done as a series of separate reports (one for the project-less, one for each project), but that's not convenient.

What would be better is to instead write a script that automates all this, pulling together a list of task UUIDs and presenting one result report. Like this: <https://taskwarrior.org/download/gtdnext.sh>

It's a bash script, and easily modifiable.

**Q: Why have attribute modifiers (`urgency,over:10`) rather than the more readable and algebraic form (`urgency>10`)?**

Taskwarrior already supports both forms. The attribute modifier form is older and predates more complex filter support. The algebraic form requires that you escape any characters that the shell will otherwise interpret.

At some point the attribute modifier form will likely be deprecated. The algebraic form is already much more capable.

```
$ task help
...
Attribute modifiers make filters more precise. Supported modifiers are:
```

Modifiers	Example	Equivalent	Meaning
not	due:today	due = today	Fuzzy match
before, below	due.before:today	due < today	Fuzzy non-match
after, above	due.after:today	due > today	Exact date comparison
none	project:none:	project == ''	Empty
any	project.any:	project != ''	Not empty
is, equals	project.is:x	project == x	Exact match
isnt	project.isnt:x	project != x	Exact non-match
has, contains	desc.has:Hello	desc ~ Hello	Pattern match
hasnt	desc.hasnt:Hello	desc !~ Hello	Pattern non-match
startswith, left	desc.left:Hel	desc ~ ^Hel	Beginning match
endswith, right	desc.right:llo	desc ~ ^llo\$	End match
word	desc.word:Hello	desc ~ '\bHello\b'	Boundaried word match
noword	desc.noword:Hello	desc !~ '\bHello\b'	Boundaried word non-match

See 'man task', for the [ATTRIBUTE MODIFIERS](#) and [EXPRESSIONS AND OPERATORS](#) sections.

**Q: How do I pipe a UUID into another task command?**

You can use Bash subprocess syntax to achieve this:

```
$ task $(task _uuids +F00) modify -F00 +BAR
```

**Q: How do I get the UUID of the last task added?**

There is a `LATEST` virtual tag for just this. It corresponds to the most recently added task.

```
$ task +LATEST _uuids
```

**Q: How do I change many due dates using one command?**

Something came up yesterday and completely derailed my day. As a result, I have many tasks that need to be rescheduled to today. What command can I use to change the due date of tasks I missed yesterday to be due today?

This will change all overdue tasks to be due today:

```
$ task +OVERDUE mod due:today
```

Depending on whether you want all overdue tasks or not, you'll need to adjust the filter.

**Q: How do I search for tasks?**

There are several ways to search the description and annotations of your tasks. The first is to simply look for a search term:

```
$ task ABC list
```

The second uses the `/.../` syntax, which uses regular expressions. This is equivalent to the above form:

```
$ task /ABC/ list
```

Regular expressions support sophisticated search terms:

```
$ task /[A-Z].../ list
```

**Q: How do I create a task that recurs on more than one day of the week?**

This is not supported, but you can simulate this by creating multiple tasks due weekly on different days:

```
$ task add Do the thing due:monday recur:weekly
$ task add Do the thing due:friday recur:weekly
```

**Q: Which version of GnuTLS should I use?**

As a security question, it is best for you to use the latest available version.

**Q: How can I include the time spent on a task in a custom report?**

Total active time is not a reportable field. It's not even a field, it's simply aggregated from the `undo.data` file. As such it is a faithful account of how long a task was in the active state, but that is not what is needed for time tracking.

Proper time tracking allows for corrections, and mining the `undo.data` file does not permit corrections. Many people confuse this for time tracking, which it is not.

You can achieve limited time tracking with hook scripts, and here are two such examples:

- <https://github.com/kostajh/taskwarrior-time-tracking-hook>
- <https://github.com/coddingtonbear/taskwarrior-timebook-hook>

The real answer, though, is to [Use Timewarrior!](#)

**Q: How do I hide tasks that have unfinished dependencies?**

Try

```
$ task -BLOCKED
```

The virtual tags that are relevant for the present situation are `BLOCKED` and/or `UNBLOCKED`.

**Q: How do I record a task that is already completed?**

There are several ways to do this. The simplest is to use the `log` command, which creates a new task and immediately sets the status to `completed`:

```
$ task log ...
```

You can achieve the same thing using the `add` command:

```
$ task add status:completed ...
```

Furthermore if you wish to record the date the task was completed:

```
$ task log end:2018-01-22 ...
```

**Q: How do I include the recurrence frequency in a report?**

Try this command:

```
$ task columns
```

This will give you a list of all the columns and their formats that you can use in a report. What you are looking for is the `recur` column.

**Q: How do I solve "Taskwarrior was built without GnuTLS support. Sync is not available.?"**

Install `libgnutls-dev` and rebuild:

```
$ make clean
$ rm -rf CMakeCache.txt
$ cmake .
$ make
$ sudo make install
```

**Q: How do I get color working?**

Try the command:

```
$ task color
```

Does it show colors?

This can happen because there are no tasks in the output that matches any of the color rules. You will need to provide metadata, such as due dates, to get any color rules to match.

**Q: How do I fix this `libgnutls-deb0.so.28` error?**

Upon upgrading from Ubuntu 15.10 to 16.04 I am now seeing the following error message when I try to run Taskwarrior:

```
error while loading shared libraries: libgnutls-deb0.so.28: cannot open
shared object file: No such file or directory
```

Gnutls was already installed and had been working on my 15.10 installation for some time now. I attempted to fix it by uninstalling and reinstalling gnutls but it still doesn't work.

Your Taskwarrior was built using a version of GnuTLS that is no longer installed. If you built Taskwarrior from source, then rebuild. If you installed Taskwarrior from a binary package, uninstall and reinstall Taskwarrior.

**Q: How do I repeat the last command?**

In tasksh like most linux shell, you can navigate the list of your last input commands by using the UP/DOWN arrows for history navigation and PageUp/PageDown for prefixed history navigation (other names may apply here).

With prefixed history navigation you can start typing a command, say `mod` and by using PageUp/PageDown you will only browse history commands that start by `mod`, whereas full history ignores any input that already exists and simply navigates history in order.

A: If (e.g. after changing from 2.3.0 to 2.5.1) in tasksh the Tasksh supports [libreadLine](#), but make sure the development version is installed, and rebuild.

Get Involved  
Submit a bug  
Clone the code

Related Sites  
[gothenburgbitfactory.org](https://gothenburgbitfactory.org)  
[holldata.net](https://holldata.net)

Contact  
✉ Email  
🐦 Twitter

Donate  
[PayPal](#)

Copyright ©, 2018  
Göteborg Bit Factory