Taskwarrior – What’s next?
Task management on the commandline

Dirk Deimeke
OpenRheinRuhr 2011
November 2011
Dirk Deimeke (that’s me)

- Married, two dogs
- Born in Wanne-Eickel
- Living in Grüt (2008 emigrated to Switzerland)
- Senior Unix Systems Administrator in Zürich
- More [http://d5e.org/](http://d5e.org/)
I first saw Taskwarrior at Ubucon 2009 in a lightning talk of Federico Hernandez, one of the core developers.

Started to use it in the beginning of 2010.

Still enthusiastic about it! ;-)

In the middle of 2010 I joined the Taskwarrior Core Team

I like working in an international team

- Watertown (Massachusetts, USA)
- Gothenburg (West Gothland, Sweden)
- Braunschweig, (Lower Saxony, Germany)
- Richmond (Virginia, USA)
- Bonn (North Rhine-Westphalia, Germany)
- Toronto (Ontario, Canada)
- Gruet (Zurich, Switzerland)

http://taskwarrior.org/

http://d5e.org/taskwarrior (German) own blog-articles
About this session

- **What is this all about?**
  This is about Taskwarrior, an efficient tool for task management. Techniques for time management are not part of this session.

- **Why do you use the beta-Version?**
  With Taskwarrior 2.0.0 a new syntax will be introduced and it makes no sense to show the old commands.

Attention!

- Feel free to ask any questions!
- If you find any errors, please tell me.
I started out using Gina Trapani’s todo.sh, which was great, but I soon wanted features that would have been difficult to implement in a shell script, so I wrote my own.

It stemmed from the fact that a todo program needs to be simple to use, and unobtrusive, otherwise it’s a hassle. But it can’t be too simple.

If you go to the trouble of capturing this information, it seems wasteful not to leverage it. So it has a lot of features, but tries to remain simple to use.

There are many different methodologies people use for managing their work, and taskwarrior tries to walk a line through the middle of all that, with features for all the different approaches.

Taskwarrior is intended to scale with the user, from very simple straightforward usage up to quite sophisticated task management.
Reasons for Taskwarrior

Taskwarrior

▷ is easy to learn.
▷ grows along with the work.
▷ is unbelievably powerful.
▷ is very fast.
▷ is easily extensible.
▷ is platform independent:
  ▷ Most flavours of Unix and Linux, including Mac OS X
  ▷ Windows with Cygwin
  ▷ via SSH from my mobile
  ▷ http://taskwarrior.org/w/Platforms
▷ is actively developed.
▷ can be influenced by users (feature requests).
▷ has excellent and very friendly support.
5 steps to install from source

All you need to compile is gcc, cmake and make.

1. **Download** of recent version
2. `tar xzf task-2.0.0.beta4.tar.gz`
3. `cmake .` *(not ./configure!)*
4. `make`
5. `sudo make install`
Test of your installation

$ task version
A configuration file could not be found in /home/user

Would you like a sample /home/user/.taskrc created, so taskwarrior can proceed? (yes/no) yes

task 2.0.0.beta4 built for linux
Copyright (C) 2006 – 2011 P. Beckingham, F. Hernandez.

Taskwarrior may be copied only under the terms of the MIT license, which may be found in the taskwarrior source kit.

Documentation for taskwarrior can be found using 'man task', 'man taskrc', 'man task−tutorial', 'man task−color', 'man task−sync', 'man task−faq' or at http://taskwarrior.org
A simple example, part 1

```bash
$ task add Pay rents # Our first task
Created task 1.

$ task add Check Taskwarrior # and now the second one
Created task 2.

$ task list # lists all open tasks

<table>
<thead>
<tr>
<th>ID</th>
<th>Project</th>
<th>Pri</th>
<th>Due</th>
<th>Active</th>
<th>Age</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td></td>
<td></td>
<td>5</td>
<td>s</td>
<td></td>
<td>Check Taskwarrior</td>
</tr>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td>10</td>
<td></td>
<td>Pay rents</td>
</tr>
</tbody>
</table>

2 tasks
$ task 2 start
Started 2 'Check Taskwarrior'.
$ task list

<table>
<thead>
<tr>
<th>ID</th>
<th>Project</th>
<th>Pri</th>
<th>Due</th>
<th>Active</th>
<th>Age</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td></td>
<td>*</td>
<td>16</td>
<td>s</td>
<td></td>
<td>Check Taskwarrior</td>
</tr>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td>21</td>
<td></td>
<td>Pay rents</td>
</tr>
</tbody>
</table>

2 tasks
$
A simple example, part 2

$ task 2 stop
Stopped 2 'Check Taskwarrior'.
$ task 2 done
Completed 2 'Check Taskwarrior'.
Marked 1 task as done.

$ task list

<table>
<thead>
<tr>
<th>ID</th>
<th>Project</th>
<th>Pri</th>
<th>Due</th>
<th>Active</th>
<th>Age</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>56s</td>
<td>Pay rents</td>
</tr>
</tbody>
</table>

1 task

$
Commands so far

- **task add**
  Adds a new task to the task list.

- **task list**
  Provides a standard listing of tasks.

- **task start**
  Marks the specified tasks as started.

- **task stop**
  Removes the start time from the specified task.

- **task done**
  Marks the specified task as done.
That’s it!

```
$ task 1 annotate Try the blue one first.
Annotated 1 with 'Try the blue one first'.

$ task list

<table>
<thead>
<tr>
<th>ID</th>
<th>Project</th>
<th>Pri</th>
<th>Started</th>
<th>Due</th>
<th>Recur</th>
<th>Age</th>
<th>Tags</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>software</td>
<td>3</td>
<td>8/17/2010</td>
<td></td>
<td>3wk</td>
<td></td>
<td>www</td>
<td>Try out the color themes</td>
</tr>
<tr>
<td></td>
<td>software</td>
<td>4</td>
<td></td>
<td>9/12/2010</td>
<td>4wk</td>
<td></td>
<td></td>
<td>Try out latest Firefox beta</td>
</tr>
</tbody>
</table>

4 tasks

$ task list due.before:eow

<table>
<thead>
<tr>
<th>ID</th>
<th>Project</th>
<th>Pri</th>
<th>Started</th>
<th>Due</th>
<th>Recur</th>
<th>Age</th>
<th>Tags</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>gtd</td>
<td>3</td>
<td>8/26/2010</td>
<td></td>
<td>weekly</td>
<td>4wk</td>
<td></td>
<td>Review task list</td>
</tr>
</tbody>
</table>

1 task

$ task 2 done
Completed 2 'Visit http://taskwarrior.org'.
Project 'software' is 33% complete (2 of 3 tasks remaining).
Marked 1 task as done.
You have higher priority tasks.
```

Thanks for your attention!

dirk@deimeke.net

http://taskwarrior.org/
Just kidding ...

There is a lot more to explore.

Even the commands from the last section are more mighty than they seem.

- `task add <mods>`
- `task <filter> list`
- `task <filter> start <mods>`
- `task <filter> stop <mods>`
- `task <filter> done <mods>`

To get an overview, take a look at the cheat sheet (pdf, 145kB).
task <filter> command <mods>

- Is the basic usage of all task related write commands.
- Write commands can operate on one task or a group of tasks or even on all tasks.
- Every command maybe abbreviated up to the minimum that is necessary to identify a single command.
- Filters can be anything from nothing to simple IDs further to regular expressions or Boolean constructs.
- Modifications can be either a change of description, a change of dates or anything else that changes a task.
- In our simple example we already used the write commands **add**, **done**, **start** and **stop**.
$ ls $PWD/*
/usr/local/share/doc/task/scripts/add-ons:
export-csv.pl  export-xml.pl  export-yad.pl  import-yaml.pl
export-html.pl  export-xml.py  export-yaml.pl  update-holidays.pl
export-ical.pl  export-xml.rb  import-todo.sh.pl

/usr/local/share/doc/task/scripts/bash:
task_completion.sh

/usr/local/share/doc/task/scripts/extensions:
command.lua  dom.lua  format.lua  program_hook.lua  task_hook.lua  uda.lua

/usr/local/share/doc/task/scripts/fish:
task.fish

/usr/local/share/doc/task/scripts/vim:
ftdetect  README  syntax

/usr/local/share/doc/task/scripts/zsh:
_task
$ . /usr/local/share/doc/task/scripts/bash/task_completion.sh
$ cp -r /usr/local/share/doc/task/scripts/vim ~/.vim
$
Most important commands

These are the most important commands, just because I use them most ;-) 

- `task <filter> modify`  
  The name says it, it modifies tasks according to the filter used.

- `task <filter> edit`  
  This starts your favourite editor with the tasks you want to change.  
  (Remember the syntax highlighting for vim?)

- `task undo`  
  Reverts the most recent change to a task.

- `task help`  
  Gives an overview of implemented commands and custom reports.

- `man task (taskrc, task-tutorial, task-color, task-faq, task-synch)`  
  Show the (almighty) man-page(s). Unlike the man-pages of many other programs they are extremely helpful and full of information and examples. Try them!
Set dateformat (check "man taskrc")

```bash
$ task show dateformat

<table>
<thead>
<tr>
<th>Config Variable</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>dateformat</td>
<td>m/d/Y</td>
</tr>
<tr>
<td>dateformat.annotation</td>
<td>m/d/Y</td>
</tr>
<tr>
<td>dateformat.holiday</td>
<td>YMD</td>
</tr>
<tr>
<td>dateformat.report</td>
<td>m/d/Y</td>
</tr>
</tbody>
</table>

Some of your .taskrc variables differ from the default values. These are highlighted in color above.

$ task config dateformat YMD
Are you sure you want to add 'dateformat' with a value of 'YMD'? (y/n) y
Config file /home/user/.taskrc modified.

$ task config dateformat.annotation YMD
Are you sure you want to add 'dateformat.annotation' with a value of 'YMD'? (y/n) y
Config file /home/user/.taskrc modified.

$ task config dateformat.report YMD
Are you sure you want to add 'dateformat.report' with a value of 'YMD'? (y/n) y
Config file /home/user/.taskrc modified.

$ # my dateformat is YMD-HN
$```
Taskwarrior – What’s next?

Set weekstart

$ task show weekstart

<table>
<thead>
<tr>
<th>Config Variable</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>weekstart</td>
<td>Sunday</td>
</tr>
</tbody>
</table>

Some of your .taskrc variables differ from the default values. These are highlighted in color above.

$ task config weekstart Monday
Are you sure you want to add 'weekstart' with a value of 'Monday'? (y/n) y

Config file /home/user/.taskrc modified.

$ task show | wc -l # nearly everything can be customized
225

$ # try it yourself without ' | wc -l'
$
Special dates

- Relative wording
today, yesterday, tomorrow

- Day number with ordinal
23rd, 3wks, 1day, 9hrs

- At some point or later
later, someday

- Start / end of (work) week, calendar week (according to settings of weekstart), month, quarter and year
sow, eoww, socw, eom, soq, eoy

- Next occurring weekday
mon, tue, ..., sat, sun
Due and wait

```
$ task add due:20111231 "Celebrate Sylvester"
Created task 2.

$ task add due:Sunday "Drive home"
Created task 3.

$ task list

<table>
<thead>
<tr>
<th>ID</th>
<th>Project</th>
<th>Pri</th>
<th>Due</th>
<th>Active</th>
<th>Age</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td></td>
<td>2</td>
<td>20111016</td>
<td>3s</td>
<td>32s</td>
<td>Drive home</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>2</td>
<td>20111231</td>
<td>29s</td>
<td>29s</td>
<td>Celebrate Sylvester</td>
</tr>
<tr>
<td>1</td>
<td></td>
<td>1</td>
<td></td>
<td></td>
<td>21h</td>
<td>Pay rents</td>
</tr>
</tbody>
</table>

3 tasks
$ task 2 modify wait:20111201
Modified 1 task.
$ task list

<table>
<thead>
<tr>
<th>ID</th>
<th>Project</th>
<th>Pri</th>
<th>Due</th>
<th>Active</th>
<th>Age</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td></td>
<td>2</td>
<td>20111016</td>
<td>32s</td>
<td>32s</td>
<td>Drive home</td>
</tr>
<tr>
<td>1</td>
<td></td>
<td>1</td>
<td></td>
<td></td>
<td>21h</td>
<td>Pay rents</td>
</tr>
</tbody>
</table>

2 tasks
$  
```
### Recurrence

```bash
$ task waiting

<table>
<thead>
<tr>
<th>ID</th>
<th>Project</th>
<th>Pri</th>
<th>Wait</th>
<th>Age</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td></td>
<td></td>
<td>20111201</td>
<td>8m</td>
<td>Celebrate Sylvester</td>
</tr>
</tbody>
</table>

1 task

$ task 1 modify due:eom recur:monthly
Modified 1 task.

$ task list

<table>
<thead>
<tr>
<th>ID</th>
<th>Project</th>
<th>Pri</th>
<th>Due</th>
<th>Active</th>
<th>Age</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td></td>
<td></td>
<td>20111016</td>
<td>8m</td>
<td></td>
<td>Drive home</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td>20111031</td>
<td></td>
<td>-</td>
<td>Pay rents</td>
</tr>
</tbody>
</table>

2 tasks

$ task recurring

<table>
<thead>
<tr>
<th>ID</th>
<th>Project</th>
<th>Pri</th>
<th>Due</th>
<th>Recur</th>
<th>Active</th>
<th>Age</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td></td>
<td></td>
<td>20111031</td>
<td>4wk</td>
<td>20s</td>
<td></td>
<td>Pay rents</td>
</tr>
</tbody>
</table>

1 task

$ # task id changed from 1 (task modify) to 4
$ # try task 1 edit
$  ```
Recurrence modifiers

- hourly
- daily, day, 1da, 2da, ...
- weekdays
- weekly, 1wk, 2wks, ...
- biweekly, fortnight
- monthly
- quarterly, 1qtr, 2qtrs, ...
- semiannual
- annual, yearly, 1yr, 2yrs, ...
- biannual, biyearly, 2yr
Until and entry

```
$ task add due:eom recur:monthly until:20121231 "Pay installment for credit"
Created task 9.

$ task add "Prepare slides for workshop"
Created task 10.

$ task 10 modify entry:yesterday
Modified 1 task.

$ task list

<table>
<thead>
<tr>
<th>ID</th>
<th>Project</th>
<th>Pri</th>
<th>Due</th>
<th>Active</th>
<th>Age</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td></td>
<td>2</td>
<td>20111016</td>
<td>1h</td>
<td></td>
<td>Drive home</td>
</tr>
<tr>
<td>11</td>
<td></td>
<td>2</td>
<td>20111031</td>
<td>-</td>
<td>1h</td>
<td>Pay installment for credit</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>2</td>
<td>20111031</td>
<td>1h</td>
<td></td>
<td>Pay rents</td>
</tr>
<tr>
<td>10</td>
<td></td>
<td></td>
<td>20111031</td>
<td>1d</td>
<td></td>
<td>Prepare slides for workshop</td>
</tr>
</tbody>
</table>
```

4 tasks

$
## Starting and stopping

$ task list

<table>
<thead>
<tr>
<th>ID</th>
<th>Project</th>
<th>Pri</th>
<th>Due</th>
<th>Active</th>
<th>Age</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td></td>
<td></td>
<td>20111016</td>
<td>1h</td>
<td>Drive home</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td></td>
<td></td>
<td>20111031</td>
<td>3m</td>
<td>Pay installment for credit</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td>20111031</td>
<td>1h</td>
<td>Pay rents</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td></td>
<td></td>
<td>20111031</td>
<td>1d</td>
<td>Prepare slides for workshop</td>
<td></td>
</tr>
</tbody>
</table>

4 tasks

$ task 10 start
Started 10 'Prepare slides for workshop'.

$ task list

<table>
<thead>
<tr>
<th>ID</th>
<th>Project</th>
<th>Pri</th>
<th>Due</th>
<th>Active</th>
<th>Age</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td></td>
<td></td>
<td>20111016</td>
<td>1h</td>
<td>Drive home</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td></td>
<td></td>
<td>20111031</td>
<td>3m</td>
<td>Pay installment for credit</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td>20111031</td>
<td>1h</td>
<td>Pay rents</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td></td>
<td></td>
<td>20111031</td>
<td>1d</td>
<td>Prepare slides for workshop</td>
<td></td>
</tr>
</tbody>
</table>

4 tasks

$ task 10 stop
Stopped 10 'Prepare slides for workshop'.

$
Holiday

Attention!

Holiday has nothing in common with the German words “Ferien“ or “Urlaub“ (this would be vacation). (Public) Holiday means “Feiertag“.

You can add holidays by either adding them via “task config“ on the commandline or by adding them directly to the ~/.taskrc-File or by including an external holiday definition.

On holidata.net you find a growing list of holiday dates, licensed CC-BY and offered by volunteers. Service was introduced by the Taskwarrior team, who is responsible for hosting and conversion to different formats.
Add holiday

```bash
$ task config holiday.carnivalstart.name Carnival Start
Are you sure you want to add 'holiday.carnivalstart.name' with a value of 'Carnival Start'? (y/n) y
Config file /home/user/.taskrc modified.

$ task config holiday.carnivalstart.date 20111111
Are you sure you want to add 'holiday.carnivalstart.date' with a value of '20111111'? (y/n) y
Config file /home/user/.taskrc modified.

$ task cal

October 2011    November 2011    December 2011

Mo Tu We Th Fr Sa Su    Mo Tu We Th Fr Sa Su    Mo Tu We Th Fr Sa Su

39     1     2
40     3     4     5     6     7     8     9
41     10    11    12    13       14    15    16
42     17    18    19    20    21    22    23
43     24    25    26    27    28    29    30
44     31

Legend: today, due, due-today, overdue, weekend, holiday, weeknumber.

$ # Holiday ist not highlighted by default
$
Calendar config with holiday

```bash
$ task show calendar

<table>
<thead>
<tr>
<th>Config Variable</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>alias.export.vcalendar</td>
<td>export.ical</td>
</tr>
<tr>
<td>calendar.details</td>
<td>sparse</td>
</tr>
<tr>
<td>calendar.details.report</td>
<td>list</td>
</tr>
<tr>
<td>calendar.holidays</td>
<td>none</td>
</tr>
<tr>
<td>calendar.legend</td>
<td>yes</td>
</tr>
<tr>
<td>calendar.offset</td>
<td>no</td>
</tr>
<tr>
<td>calendar.offset.value</td>
<td>-1</td>
</tr>
<tr>
<td>color.calendar.due</td>
<td>color0 on color1</td>
</tr>
<tr>
<td>color.calendar.due.today</td>
<td>color15 on color1</td>
</tr>
<tr>
<td>color.calendar.holiday</td>
<td>color0 on color11</td>
</tr>
<tr>
<td>color.calendar.overdue</td>
<td>color0 on color9</td>
</tr>
<tr>
<td>color.calendar.today</td>
<td>color15 on rgb013</td>
</tr>
<tr>
<td>color.calendar.weekend</td>
<td>color235</td>
</tr>
<tr>
<td>color.calendar.weeknumber</td>
<td>rgb013</td>
</tr>
</tbody>
</table>
```

Some of your .taskrc variables differ from the default values. These are highlighted in color above.

$ task config calendar.holidays full
Are you sure you want to add 'calendar.holidays' with a value of 'full'? (y/n) y

Config file /home/user/.taskrc modified.

$
Calendar with holiday

Legend: today, due, due-today, overdue, weekend, holiday, weeknumber.

Date  Holiday
20111111 Carnival Start

$
Calendar with due tasks

```
$ task config calendar.details full
Are you sure you want to add 'calendar.details' with a value of 'full'? (y/n) y
Config file /home/user/.taskrc modified.
$ task cal

Legend: **today**, due, due-today, overdue, weekend, **holiday**, weeknumber.

<table>
<thead>
<tr>
<th>ID</th>
<th>Project</th>
<th>Pri</th>
<th>Due</th>
<th>Active</th>
<th>Age</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td></td>
<td></td>
<td>20111016</td>
<td>3d</td>
<td>3d</td>
<td>Drive home</td>
</tr>
<tr>
<td>11</td>
<td></td>
<td></td>
<td>20111031</td>
<td>3d</td>
<td>3d</td>
<td>Pay installment for credit</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td>20111031</td>
<td>3d</td>
<td>3d</td>
<td>Pay rents</td>
</tr>
</tbody>
</table>

3 tasks

$ # did a 'task config calendar.holidays sparse' before
Project and subproject

$ task 3 modify pro:ubucon
Modified 1 task.
The project 'ubucon' has changed. Project 'ubucon' is 0% complete (1 of 1 tasks remaining).

$ task 10 modify pro:ubucon.workshop
Modified 1 task.
The project 'ubucon.workshop' has changed. Project 'ubucon.workshop' is 0% complete (1 of 1 tasks remaining).

$ task 4 modify pro:private
Modified 1 task.
The project 'private' has changed. Project 'private' is 0% complete (1 of 1 tasks remaining).

$ task list

<table>
<thead>
<tr>
<th>ID</th>
<th>Project</th>
<th>Pri</th>
<th>Due</th>
<th>Active</th>
<th>Age</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>ubucon</td>
<td></td>
<td>20111016</td>
<td>3d</td>
<td>3d</td>
<td>Drive home</td>
</tr>
<tr>
<td>4</td>
<td>private</td>
<td></td>
<td>20111031</td>
<td>3d</td>
<td>3d</td>
<td>Pay rents</td>
</tr>
<tr>
<td>10</td>
<td>ubucon.workshop</td>
<td></td>
<td>20111031</td>
<td>*</td>
<td>5d</td>
<td>Prepare slides for workshop</td>
</tr>
</tbody>
</table>

3 tasks
$
## List of Projects

```
$ task projects

<table>
<thead>
<tr>
<th>Project</th>
<th>Tasks</th>
<th>Pri:None</th>
<th>Pri:H</th>
<th>Pri:M</th>
<th>Pri:L</th>
</tr>
</thead>
<tbody>
<tr>
<td>(none)</td>
<td>7</td>
<td>7</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>private</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>ubucon</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>ubucon.workshop</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

3 projects (10 tasks)

$ task pro:ubucon ls

<table>
<thead>
<tr>
<th>ID</th>
<th>Project</th>
<th>Pri</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>ubucon</td>
<td></td>
<td>Drive home</td>
</tr>
<tr>
<td>10</td>
<td>ubucon.workshop</td>
<td></td>
<td>Prepare slides for workshop</td>
</tr>
</tbody>
</table>

2 tasks

$ task 10 done
Completed 10 'Prepare slides for workshop'.
Marked 1 task as done.
Project 'ubucon.workshop' is 100% complete (0 of 1 tasks remaining).

$
Tags

```
$ task 3 modify +leipzig
Modified 1 task.
$ task 4 modify +banking
Modified 1 task.
$ task list

ID  Project  Pri  Due      Active  Age  Description
3   ubucon   2011016  3d  Drive home
4   private  2011031  3d  Pay rents

2 tasks
$ task +leipzig list

ID  Project  Pri  Due      Active  Age  Description
3   ubucon   2011016  3d  Drive home

1 task
$
```
$ task list

<table>
<thead>
<tr>
<th>ID</th>
<th>Project</th>
<th>Pri</th>
<th>Due</th>
<th>Active</th>
<th>Age</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>ubucon</td>
<td></td>
<td>20111016</td>
<td></td>
<td>3d</td>
<td>Drive home</td>
</tr>
<tr>
<td>4</td>
<td>private H</td>
<td></td>
<td>20111031</td>
<td></td>
<td>3d</td>
<td>Pay rents</td>
</tr>
</tbody>
</table>

2 tasks
$ task 3 annotate "Do not forget your head"
Annotated 3 'Drive home'

$ task 4 annotate "Use wifes account"
Annotated 4 'Pay rents'

$ task list

<table>
<thead>
<tr>
<th>ID</th>
<th>Project</th>
<th>Pri</th>
<th>Due</th>
<th>Active</th>
<th>Age</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>ubucon</td>
<td></td>
<td>20111016</td>
<td>20111013</td>
<td>3d</td>
<td>Do not forget your head</td>
</tr>
<tr>
<td>4</td>
<td>private H</td>
<td></td>
<td>20111031</td>
<td></td>
<td>3d</td>
<td>Pay rents</td>
</tr>
</tbody>
</table>

2 tasks
$ task 4 annotate "Use wifes account"
Found annotation '' and deleted it.
### Dependency, part 1

```
$ task add "Send letter to Fritz"
Created task 10.

$ task add "Write letter"
Created task 11.

$ task 10 modify depends:11
Modified 1 task.
$ task blocked

<table>
<thead>
<tr>
<th>ID</th>
<th>Deps</th>
<th>Project</th>
<th>Pri</th>
<th>Due</th>
<th>Active</th>
<th>Age</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>11</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Send letter to Fritz</td>
</tr>
</tbody>
</table>

1 task
$ task unblocked

<table>
<thead>
<tr>
<th>ID</th>
<th>Deps</th>
<th>Project</th>
<th>Pri</th>
<th>Due</th>
<th>Active</th>
<th>Age</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td></td>
<td>ubucon</td>
<td>20111016</td>
<td></td>
<td>3d</td>
<td></td>
<td>Drive home</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>20111013 Do not forget your head</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>private H</td>
<td>20111031</td>
<td></td>
<td>3d</td>
<td></td>
<td>Pay rents</td>
</tr>
<tr>
<td>11</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Write letter</td>
</tr>
</tbody>
</table>

3 tasks
$
Dependency, part 2

$ task 10 done
Task 10 is blocked by:
  11 Write letter
Completed 10 'Send letter to Fritz'.
Marked 1 task as done.

You have more urgent tasks.
$ task undo

The last modification was made 10/13/2011

<table>
<thead>
<tr>
<th>Prior Values</th>
<th>Current Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>depends</td>
<td>e3ba5a87-06a2-483a-ac33-a66c6bf6a75</td>
</tr>
<tr>
<td>description</td>
<td>Send letter to Fritz</td>
</tr>
<tr>
<td>entry</td>
<td>20111013</td>
</tr>
<tr>
<td>status</td>
<td>pending</td>
</tr>
<tr>
<td>uuid</td>
<td>39c51e45-7696-4712-81e1-cb88cab67818</td>
</tr>
<tr>
<td>end</td>
<td>20111013</td>
</tr>
</tbody>
</table>

The undo command is not reversible. Are you sure you want to revert to the previous state? (y/n) y
Modified task reverted.
Dependency, part 3

$ task 10,11 done
Task 10 is blocked by:
  11 Write letter
Completed 10 'Send letter to Fritz'.
Completed 11 'Write letter'.
Marked 2 tasks as done.

You have more urgent tasks.

$ task blocked
No matches.
$
Some predefined reports

See task reports for a full list (26 reports).

- **active** Lists active tasks matching the specified criteria
- **all** Lists all tasks matching the specified criteria, including parents of recurring tasks
- **completed** Lists completed tasks matching the specified criteria
- **information** Shows all data and metadata for specified tasks
- **newest** Shows the newest tasks
- **next** Lists the most urgent tasks
- **oldest** Shows the oldest tasks
- **overdue** Lists overdue tasks matching the specified criteria
- **summary** Shows a report of task status by project
- **tags** Shows a list of all tags used
```bash
$ task ghistory.annual

<table>
<thead>
<tr>
<th>Year</th>
<th>Number Added/Completed/Deleted</th>
</tr>
</thead>
<tbody>
<tr>
<td>2011</td>
<td>23</td>
</tr>
</tbody>
</table>

Legend: Added, Completed, Deleted

$ task ghistory.monthly

<table>
<thead>
<tr>
<th>Year</th>
<th>Month</th>
<th>Number Added/Completed/Deleted</th>
</tr>
</thead>
<tbody>
<tr>
<td>2011</td>
<td>October</td>
<td>23</td>
</tr>
</tbody>
</table>

Legend: Added, Completed, Deleted

$ task history.monthly

<table>
<thead>
<tr>
<th>Year</th>
<th>Month</th>
<th>Added</th>
<th>Completed</th>
<th>Deleted</th>
<th>Net</th>
</tr>
</thead>
<tbody>
<tr>
<td>2011</td>
<td>October</td>
<td>23</td>
<td>6</td>
<td>8</td>
<td>9</td>
</tr>
</tbody>
</table>

Average 23 6 8 9

$
Report definitions

```
$ task show report.minimal

<table>
<thead>
<tr>
<th>Config Variable</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>report.minimal.columns</td>
<td>id,project,description.truncated</td>
</tr>
<tr>
<td>report.minimal.description</td>
<td>A really minimal listing</td>
</tr>
<tr>
<td>report.minimal.filter</td>
<td>status:pending</td>
</tr>
<tr>
<td>report.minimal.labels</td>
<td>ID,Project,Description</td>
</tr>
<tr>
<td>report.minimal.sort</td>
<td>project+,description+</td>
</tr>
</tbody>
</table>

Some of your .taskrc variables differ from the default values. These are highlighted in **color** above.

$ task show report.list

<table>
<thead>
<tr>
<th>Config Variable</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>report.list.columns</td>
<td>id,project,priority,due,start.active,entry.age,description</td>
</tr>
<tr>
<td>report.list.description</td>
<td>Lists all tasks matching the specified criteria</td>
</tr>
<tr>
<td>report.list.filter</td>
<td>status:pending</td>
</tr>
<tr>
<td>report.list.labels</td>
<td>ID,Project,Pri,Due,Active,Age,Description</td>
</tr>
<tr>
<td>report.list.sort</td>
<td>due+,priority-,start-,project+,description+</td>
</tr>
</tbody>
</table>

Some of your .taskrc variables differ from the default values. These are highlighted in **color** above.

$ # try 'task show report' to see all definitions
```
## Dirks task list

```bash
$ echo "
> report.bb.description=Dirks task list
> report.bb.columns=id,project,priority,de,due,de.countdown,de.tags,de.description
> report.bb.labels=ID,Project,Pri,De,Due,Countdown,Tags,Description
> report.bb.sort=due+,priority-,project+,de.description+
> report.bb.filter=status:pending
> " >> ~/.taskrc
$ task ll
```

<table>
<thead>
<tr>
<th>ID</th>
<th>Project</th>
<th>Pri</th>
<th>Due</th>
<th>Countdown</th>
<th>Tags</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>ubucon</td>
<td></td>
<td>20111016</td>
<td>-2 days</td>
<td>leipzig</td>
<td>Drive home</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>20111013 Do not forget your head</td>
</tr>
<tr>
<td>4</td>
<td>private H</td>
<td>2</td>
<td>20111031</td>
<td>-2 wks</td>
<td>banking</td>
<td>Pay rents</td>
</tr>
</tbody>
</table>

2 tasks

$
Set default command

```bash
$ task show default

Config Variable Value
default.command next
defaultheight 24
defaultwidth 80

Some of your .taskrc variables differ from the default values. These are highlighted in color above.

$ task config default.command ll
Are you sure you want to add 'default.command' with a value of 'll'? (y/n) y
Config file /home/user/.taskrc modified.

$ task
[task ll]

<table>
<thead>
<tr>
<th>ID</th>
<th>Project</th>
<th>Pri</th>
<th>Due</th>
<th>Countdown</th>
<th>Tags</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>ubucon</td>
<td>3</td>
<td>20111016</td>
<td>-2 days</td>
<td>leipzig</td>
<td>Drive home 20111013 Do not forget your head</td>
</tr>
<tr>
<td>4</td>
<td>private</td>
<td>H</td>
<td>20111031</td>
<td>-2 wks</td>
<td>banking</td>
<td>Pay rents</td>
</tr>
</tbody>
</table>

2 tasks
$
Filtering in general

You can filter for any modifier. If you don’t use a modifier description is searched for the term, which may be a regular expression, on the command line. Filters may be combined.

The following attribute modifiers maybe applied as well. Names in brackets can be used alternatively.

So a filter can look like “attribute.modifier:value“.

- before, after
- none, any
- is (equals), isnt (not)
- has (contains), hasnt
- startswith (left), endswith (right)
- word, noword
# Searches

```
$ task
[task ll]

<table>
<thead>
<tr>
<th>ID</th>
<th>Project</th>
<th>Pri</th>
<th>Due</th>
<th>Countdown</th>
<th>Tags</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>ubucon</td>
<td></td>
<td>20111016</td>
<td>-2 days</td>
<td>leipzig</td>
<td>Drive home</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>20111013 Do not forget your head</td>
</tr>
<tr>
<td>4</td>
<td>private H</td>
<td>20111031</td>
<td>-2 wks</td>
<td>banking</td>
<td>Pay rents</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>five</td>
</tr>
<tr>
<td>13</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>four</td>
</tr>
<tr>
<td>10</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>one</td>
</tr>
<tr>
<td>12</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>three</td>
</tr>
<tr>
<td>11</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>two</td>
</tr>
</tbody>
</table>

7 tasks
```

$ task pay
[task ll pay]
No matches.

$ task /[Pp]ay/  
[task ll /[Pp]ay/]

```
<table>
<thead>
<tr>
<th>ID</th>
<th>Project</th>
<th>Pri</th>
<th>Due</th>
<th>Countdown</th>
<th>Tags</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>private H</td>
<td>20111031</td>
<td>-2 wks</td>
<td>banking</td>
<td>Pay rents</td>
<td></td>
</tr>
</tbody>
</table>
```

1 task
$
Attribute modifiers

$ task due.before:2011020
[task ll due.before:2011020]

<table>
<thead>
<tr>
<th>ID</th>
<th>Project</th>
<th>Pri</th>
<th>Due</th>
<th>Countdown</th>
<th>Tags</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>ubucon</td>
<td></td>
<td>2011016</td>
<td>-2 days</td>
<td>leipzig</td>
<td>Drive home 2011013 Do not forget your head</td>
</tr>
</tbody>
</table>

1 task

$ task project.not:ubucon
[task ll project.not:ubucon]

<table>
<thead>
<tr>
<th>ID</th>
<th>Project</th>
<th>Pri</th>
<th>Due</th>
<th>Countdown</th>
<th>Tags</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>private</td>
<td>H</td>
<td>2011031</td>
<td>-2 wks</td>
<td>banking</td>
<td>Pay rents</td>
</tr>
<tr>
<td>14</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>five</td>
</tr>
<tr>
<td>13</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>four</td>
</tr>
<tr>
<td>10</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>one</td>
</tr>
<tr>
<td>12</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>three</td>
</tr>
<tr>
<td>11</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>two</td>
</tr>
</tbody>
</table>

6 tasks

$
Combining

```
$ task project:ubucon +banking
No matches.
$ task project:ubucon -banking

<table>
<thead>
<tr>
<th>ID</th>
<th>Project</th>
<th>Pri</th>
<th>Due</th>
<th>Countdown</th>
<th>Tags</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>ubucon</td>
<td></td>
<td>20111016</td>
<td>-2 days</td>
<td>leipzig</td>
<td>Drive home</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>20111013 Do not forget your head</td>
</tr>
</tbody>
</table>

1 task
$ task status:completed project:ubucon
No matches.
$ task status:completed project:ubucon completed

Complete Project       Pri | Age | Description
-----------------------|-----|--------------------------
20111013 ubucon.workshop| 5d  | Prepare slides for workshop

1 task
$ task show report.ll.filter

Config Variable   Value
-------------------|------------------
report.ll.filter   status:pending
```
This is by far not all

- **task log**
  for logging a task after it is already done.

- **task diag**
  to help support for diagnostic purpose.

- **task shell**
  a simple shell to get rid of the necessity to type “task“ all the time.

- ... and many more!
Questions?
Support

taskwarrior.org

support@taskwarrior.org

#taskwarrior on freenode.net – @taskwarrior on Twitter or identi.ca
Thanks for your patience!

Dirk Deimeke, Taskwarrior-Team, 2011, CC-BY

dirk@deimeke.net
d5e.org – dirk.deimeke.net