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

Dirk Deimeke

Ubucon 2011

October 2011
Content

Introduction
Installation
Simple ToDo-Lists
General
Working with dates
Getting sorted
Dependencies
Reports
Filtering
Miscellaneous
That's all
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/
Dirk Deimeke (Ubuntu)

- President of German Ubuntu association *ubuntu Deutschland e. V.*
- Official point of contact for the *German LoCo-Team*
- Member of *Swiss LoCo-Team*
- Ubuntu Member
- More [http://d5e.org/ubuntu](http://d5e.org/ubuntu) (German) or [https://wiki.ubuntu.com/DirkDeimeke](https://wiki.ubuntu.com/DirkDeimeke) (English)
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 Taskwarrior
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 are the slides in English?**
  Taskwarrior is an international project.

- **Why do you use the beta-Version?**
  With Taskwarrior 2.0.0 a new syntax will be introduced.

**Attention!**

- This workshop is not a lecture, I want to do this together with you.
- Ask your questions if you have some!
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 and Ubuntu (PPA exists)
  ▶ 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.
**Installation with package management**

All download-addresses
http://taskwarrior.org/download

**Since Maverick Meerkat in Universe**
(Unfortunately version is behind recent)

$ sudo apt-get install task

There is a PPA with the most recent version
https://launchpad.net/~ultrafredde/+archive/ppa

$ sudo add-apt-repository ppa:ultrafredde/ppa
$ sudo apt-get update
$ sudo apt-get install task
Installation from source

The Meta-Package build-essential and cmake are all you need to compile.

$ sudo apt-get install build-essential cmake

Download of recent version

$ curl -O http://www.taskwarrior.org/download/task-2.0.0.beta3.tar.gz
$ # or wget http://www.taskwarrior.org/download/task-2.0.0.beta3.tar.gz

Untar and compile

$ tar task-2.0.0.beta3.tar.gz
$ cd task-2.0.0.beta3
$ cmake .
$ make
$ sudo make install

The last command installs Taskwarrior to /usr/local.
Installation from source with target directory

If you don’t have root permissions or in case you want to use other directories, this is possible as well. (No speciality of Taskwarrior).

```
$ cmake -DCMAKE_INSTALL_PREFIX=/home/dirk/task-2.0.0.beta3 
$ make
$ make install # without "sudo"
```

It makes sense to define the following three variables for the next steps. (The first one is not needed, I use it only for this topic to fit on one slide).

```
$ taskdir=/home/dirk/task_2.0.0.beta3
$ export PATH=${taskdir}/bin:${PATH}
$ export LD_LIBRARY_PATH=${taskdir}/lib:${LD_LIBRARY_PATH}
$ export MANPATH=${taskdir}/man:${MANPATH}
```
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? (y/n) y

task 2.0.0.beta3 built for linux
Copyright (C) 2006 — 2011 P. Beckingham, F. Hernandez.

Taskwarrior may be copied only under the terms of the GNU General Public 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
```
Remove Taskwarrior

Package manager

$ apt-get purge task

Sources

$ make uninstall
A simple example, part 1

$ 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>5s</td>
<td></td>
<td></td>
<td>Check Taskwarrior</td>
</tr>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td>10s</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>*16s</td>
<td></td>
<td></td>
<td>Check Taskwarrior</td>
</tr>
<tr>
<td>1</td>
<td></td>
<td></td>
<td>21s</td>
<td></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!

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.
Dirk Deimeke (Ubucon 2011)

Taskwarrior – What’s next?

October 2011 19 / 64

Scripts

```
$ 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  uada.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!
Dateformats – from 'man taskrc'

- m minimal-digit month, for example 1 or 12
- d minimal-digit day, for example 1 or 30
- y two-digit year, for example 09
- D two-digit day, for example 01 or 30
- M two-digit month, for example 01 or 12
- Y four-digit year, for example 2009
- a short name of weekday, for example Mon or Wed
- A long name of weekday, for example Monday or Wednesday
- b short name of month, for example Jan or Aug
- B long name of month, for example January or August
- V weeknumber, for example 03 or 37
- H two-digit hour, for example 03 or 11
- N two-digit minutes, for example 05 or 42
- S two-digit seconds, for example 07 or 47

The string may also contain other characters to act as spacers, or formatting. Examples for other values of dateformat:

- d/m/Y would use for input and output 24/7/2009
- yMD would use for input and output 090724
- M−D−Y would use for input and output 07−24−2009

Examples for other values of dateformat.report:

- a D b Y (V) would do an output as "Fri 24 Jul 2009 (30)"
- A, B D, Y would do an output as "Friday, July 24, 2009"
- vV a Y−M−D would do an output as "v30 Fri 2009−07−24"
- yMD.HN would do an output as "110124.2342"
- m/d/Y H:N would do an output as "1/24/2011 10:42"
- a D b Y H:N:S would do and output as "Mon 24 Jan 2011 11:19:42"
Set dateformat

```bash
$ task show dateformat

Config Variable         Value
dateformat              m/d/Y
dateformat.annotation   m/d/Y
dateformat.holiday      YMD
dateformat.report       m/d/Y

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
$```
Set weekstart

$ task show weekstart

Config Variable Value
weekstart Sunday

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 (1)

- **Relative wording**
  - task ... due:today
  - task ... due:yesterday
  - task ... due:tomorrow

- **Day number with ordinal**
  - task ... due:23rd
  - task ... due:3wks
  - task ... due:1day
  - task ... due:9hrs

- **At some point or later**
  - task ... wait:later task ... wait:someday This sets the wait date to 1/18/2038.
Special dates (2)

- Start / end of (work) week, calendar week (according to settings of weekstart), month, quarter and year
  
  - task ... due:sow
  - task ... due:eow
  - task ... due:soww
  - task ... due:eoww
  - task ... due:socw
  - task ... due:eocw
  - task ... due:som
  - task ... due:eom
  - task ... due:soq
  - task ... due:eoq
  - task ... due:soy
  - task ... due:eoy

- Next occurring weekday
  
  - task ... due:fri
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>Drive home</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>2</td>
<td>20111231</td>
<td>29s</td>
<td>Celebrate Sylvester</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>21h Pay rents</td>
<td></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>Drive home</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>21h Pay rents</td>
<td></td>
</tr>
</tbody>
</table>

2 tasks
$
Recurrence

```
$ 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>Drive home</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td>20111031</td>
<td></td>
<td>- Pay rents</td>
<td></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 (1)

- **hourly**
  Every hour.

- **daily, day, 1da, 2da, ...**
  Every day or a number of days.

- **weekdays**
  Mondays, Tuesdays, Wednesdays, Thursdays, Fridays and skipping weekend days.

- **weekly, 1wk, 2wks, ...**
  Every week or a number of weeks.

- **biweekly, fortnight**
  Every two weeks.

- **monthly**
  Every month.

- **quarterly, 1qtr, 2qtrs, ...**
  Every three months, a quarter, or a number of quarters.
Recurrence modifiers (2)

- **semiannual**
  Every six months.

- **annual, yearly, 1yr, 2yrs, ...**
  Every year or a number of years.

- **biannual, biyearly, 2yr**
  Every two years.
Until and entry

```bash
$ 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>20111016</td>
<td></td>
<td></td>
<td>1h</td>
<td>Drive home</td>
</tr>
<tr>
<td>11</td>
<td></td>
<td>20111031</td>
<td></td>
<td>-</td>
<td></td>
<td>Pay installment for credit</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>20111031</td>
<td></td>
<td>1h</td>
<td></td>
<td>Pay rents</td>
</tr>
<tr>
<td>10</td>
<td></td>
<td>20111031</td>
<td></td>
<td></td>
<td>1d</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>20111016</td>
<td>1h</td>
<td>Drive home</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td></td>
<td>20111031</td>
<td>3m</td>
<td>Pay installment for credit</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>20111031</td>
<td>1h</td>
<td>Pay rents</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td></td>
<td>20111031</td>
<td>1d</td>
<td>Prepare slides for workshop</td>
<td></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>20111016</td>
<td>1h</td>
<td>Drive home</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td></td>
<td>20111031</td>
<td>3m</td>
<td>Pay installment for credit</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>20111031</td>
<td>1h</td>
<td>Pay rents</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td></td>
<td>20111031</td>
<td>* 1d</td>
<td>Prepare slides for workshop</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

4 tasks

$ task 10 stop

Stopped 10 'Prepare slides for workshop'.

$ 
```
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

```
$ 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

<p>| | | | | | | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>39</td>
<td>1</td>
<td>2</td>
<td>44</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>48</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>40</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>45</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>10</td>
</tr>
<tr>
<td>41</td>
<td>10</td>
<td>11</td>
<td>12</td>
<td>13</td>
<td>14</td>
<td>15</td>
<td>16</td>
<td>46</td>
<td>14</td>
<td>15</td>
<td>16</td>
<td>17</td>
</tr>
<tr>
<td>42</td>
<td>17</td>
<td>18</td>
<td>19</td>
<td>20</td>
<td>21</td>
<td>22</td>
<td>23</td>
<td>47</td>
<td>21</td>
<td>22</td>
<td>23</td>
<td>24</td>
</tr>
<tr>
<td>43</td>
<td>24</td>
<td>25</td>
<td>26</td>
<td>27</td>
<td>28</td>
<td>29</td>
<td>30</td>
<td>48</td>
<td>28</td>
<td>29</td>
<td>30</td>
<td>48</td>
</tr>
<tr>
<td>44</td>
<td>31</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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

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

```
$ 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

```
$ task cal

<table>
<thead>
<tr>
<th></th>
<th>October 2011</th>
<th></th>
<th>November 2011</th>
<th></th>
<th>December 2011</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Mo</td>
<td>Tu</td>
<td>We</td>
<td>Th</td>
<td>Fr</td>
</tr>
<tr>
<td>39</td>
<td>1</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>40</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
</tr>
<tr>
<td>41</td>
<td>10</td>
<td>11</td>
<td>12</td>
<td>13</td>
<td>14</td>
</tr>
<tr>
<td>42</td>
<td>17</td>
<td>18</td>
<td>19</td>
<td>20</td>
<td>21</td>
</tr>
<tr>
<td>43</td>
<td>24</td>
<td>25</td>
<td>26</td>
<td>27</td>
<td>28</td>
</tr>
<tr>
<td>44</td>
<td>31</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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></td>
<td>Drive home</td>
</tr>
<tr>
<td>11</td>
<td></td>
<td></td>
<td>20111031</td>
<td>3d</td>
<td></td>
<td>Pay installment for credit</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td>20111031</td>
<td>3d</td>
<td></td>
<td>Pay rents</td>
</tr>
</tbody>
</table>

3 tasks
$ # did a 'task config calendar.holidays sparse' before
```
Timesheet

$ task add Drive to Leipzig
Created task 12.

$ task 12 done
Completed 12 'Drive to Leipzig'.
Marked 1 task as done.

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

20111010 - 20111016
Completed (2 tasks)

<table>
<thead>
<tr>
<th>Project</th>
<th>Due</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>20111031</td>
<td></td>
<td>Pay installment for credit</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Drive to Leipzig</td>
</tr>
</tbody>
</table>

Started (1 tasks)

<table>
<thead>
<tr>
<th>Project</th>
<th>Due</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Prepare slides for workshop</td>
</tr>
</tbody>
</table>
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></td>
<td></td>
<td></td>
<td>5d Prepare slides for workshop</td>
</tr>
</tbody>
</table>

3 tasks
$

$
## 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).
```

$
```
$ task 3 modify +leipzig
Modified 1 task.
$ task 4 modify +banking
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>ubucon</td>
<td>2</td>
<td>20111016</td>
<td>3d</td>
<td>3d</td>
<td>Drive home</td>
</tr>
<tr>
<td>4</td>
<td>private</td>
<td>2</td>
<td>20111031</td>
<td>3d</td>
<td>3d</td>
<td>Pay rents</td>
</tr>
</tbody>
</table>

2 tasks
$ task +leipzig 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>2</td>
<td>20111016</td>
<td>3d</td>
<td>3d</td>
<td>Drive home</td>
</tr>
</tbody>
</table>

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>2011016</td>
<td>3d</td>
<td>Drive home</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>private H</td>
<td>2011031</td>
<td>3d</td>
<td>Pay rents</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

2 tasks

$ task 3 annotate "Do not forget your head"
Annotated 3 'Drive home'
Annotated 1 task.

$ task 4 annotate "Use wifes account"
Annotated 4 'Pay rents'
Annotated 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>2011016</td>
<td>3d</td>
<td>Drive home</td>
<td>2011013 Do not forget your head</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>private H</td>
<td>2011031</td>
<td>3d</td>
<td>Pay rents</td>
<td>2011013 Use wifes account</td>
<td></td>
</tr>
</tbody>
</table>

2 tasks

$ task 4 denotate "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>1m</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>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td></td>
<td>ubucon</td>
<td></td>
<td>20111016</td>
<td>3d</td>
<td>6d</td>
<td>Drive home</td>
<td>20111013 Do not forget your head</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>private H</td>
<td></td>
<td>20111031</td>
<td>3d</td>
<td>54s</td>
<td>Pay rents</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td></td>
<td></td>
<td></td>
<td>2011031</td>
<td></td>
<td>54s</td>
<td>Write letter</td>
<td></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></td>
</tr>
<tr>
<td>e3ba5a87-06a2-483a-ac33-a66c6bf-6a75</td>
<td>e3ba5a87-06a2-483a-ac33-a66c6bfb-6a75</td>
</tr>
<tr>
<td>description</td>
<td></td>
</tr>
<tr>
<td>Send letter to Fritz</td>
<td>Send letter to Fritz</td>
</tr>
<tr>
<td>entry</td>
<td></td>
</tr>
<tr>
<td>20111013</td>
<td>20111013</td>
</tr>
<tr>
<td>status</td>
<td>pending</td>
</tr>
<tr>
<td>completed</td>
<td></td>
</tr>
<tr>
<td>uuid</td>
<td></td>
</tr>
<tr>
<td>39c51e45-7696-4712-81e1-cb88cab-67818</td>
<td>39c51e45-7696-4712-81e1-cb88cab6-7818</td>
</tr>
<tr>
<td>end</td>
<td></td>
</tr>
<tr>
<td></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.
$
Predefined reports (from task reports), part 2

These reports were already used.

- **blocked** Lists all blocked tasks matching the specified criteria
- **list** Lists all tasks matching the specified criteria
- **long** Lists all task, all data, matching the specified criteria
- **projects** Shows a list of all project names used, and how many tasks are in each
- **recurring** Lists recurring tasks matching the specified criteria
- **unblocked** Lists all unblocked tasks matching the specified criteria
- **waiting** Lists all waiting tasks matching the specified criteria
Predefined reports (from task reports), part 2

New ones:

- **active** Lists active tasks matching the specified criteria
- **all** Lists all tasks matching the specified criteria, including parents of recurring tasks
- **burndown.daily** Shows a graphical burndown chart, by day
- **burndown.monthly** Shows a graphical burndown chart, by month
- **burndown.weekly** Shows a graphical burndown chart, by week
- **completed** Lists completed tasks matching the specified criteria
- **ghistory.annual** Shows a graphical report of task history, by year
- **ghistory.monthly** Shows a graphical report of task history, by month
- **history.annual** Shows a report of task history, by year
- **history.monthly** Shows a report of task history, by month
- **information** Shows all data and metadata for specified tasks
- **ls** Minimal listing of all tasks matching the specified criteria
- **minimal** A really minimal listing
And more:

- **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

26 reports in total
burndown.daily

Daily Burndown

Add rate: 3.8/d
Done/Delete rate: 2.3/d

Estimated completion: No convergence
### ghISTORY, hISTORY

#### ghISTORY.annual

<table>
<thead>
<tr>
<th>Year</th>
<th>Number Added/Completed/Deleted</th>
</tr>
</thead>
<tbody>
<tr>
<td>2011</td>
<td><img src="image" alt="Chart showing 23 added, 6 completed, 8 deleted tasks" /></td>
</tr>
</tbody>
</table>

**Legend:** Added, Completed, Deleted

#### 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><img src="image" alt="Chart showing 23 added, 6 completed, 8 deleted tasks" /></td>
</tr>
</tbody>
</table>

**Legend:** Added, Completed, Deleted

#### 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
ls, minimal, summary

$ task ls

<table>
<thead>
<tr>
<th>ID</th>
<th>Project</th>
<th>Pri</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>private</td>
<td>H</td>
<td>Pay rents</td>
</tr>
<tr>
<td>3</td>
<td>ubucon</td>
<td></td>
<td>Drive home</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>20111013 Do not forget your head</td>
</tr>
</tbody>
</table>

2 tasks

$ task minimal

<table>
<thead>
<tr>
<th>ID</th>
<th>Project</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>private</td>
<td>Pay rents</td>
</tr>
<tr>
<td>3</td>
<td>ubucon</td>
<td>Drive home</td>
</tr>
</tbody>
</table>

2 tasks

$ task summary

Project Remaining  Avg age Complete 0% 100%
private 1 3 days 0%
ubucon 1 3 days 0%

2 projects

$
## Report definitions

<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.

<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

```
$ echo "
  > report.ll.description=Dirks task list
  > report.ll.columns=id,project,priority,due,due.countdown,tags,description
  > report.ll.labels=ID,Project,Pri,Due,Countdown,Tags,Description
  > report.ll.sort=due+,priority-,project+,description+
  > report.ll.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</td>
<td>H</td>
<td>20111031</td>
<td>-2 wks</td>
<td>banking</td>
<td>Pay rents</td>
</tr>
</tbody>
</table>

2 tasks

```
Set default command

```
$ 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>2</td>
<td>20111016</td>
<td>-2 days</td>
<td>leipzig Drive home</td>
<td>20111013 Do not forget your head</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>private</td>
<td>H</td>
<td>20111031</td>
<td>-2 wks</td>
<td>banking Pay rents</td>
<td></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

```bash
$ task

<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>4</td>
<td>private</td>
<td></td>
<td>20111031</td>
<td>-2 wks</td>
<td>banking</td>
<td>Pay rents</td>
</tr>
<tr>
<td></td>
<td>H</td>
<td>14</td>
<td></td>
<td></td>
<td></td>
<td>five</td>
</tr>
<tr>
<td></td>
<td></td>
<td>13</td>
<td></td>
<td></td>
<td></td>
<td>four</td>
</tr>
<tr>
<td></td>
<td></td>
<td>10</td>
<td></td>
<td></td>
<td></td>
<td>one</td>
</tr>
<tr>
<td></td>
<td></td>
<td>12</td>
<td></td>
<td></td>
<td></td>
<td>three</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>two</td>
</tr>
</tbody>
</table>

7 tasks

$ task pay

No matches.

$ task /[Pp]ay/  

No matches.

$ task /[Pp]ay/

1 task

$ task
```
Attribute modifiers

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

ID  Project  Pri  Due      Countdown  Tags    Description
3   ubucon   2011016 -2 days  leipzig  Drive home
     2011013 Do not forget your head

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

ID  Project  Pri  Due      Countdown  Tags    Description
4   private H 2011031  -2 wks  banking  Pay rents
14  five
13  four
10  one
12  three
11  two

6 tasks
$  
```
Combining

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

ID  Project     Pri  Due  Countdown   Tags         Description
3   ubucon      2011016  -2 days leiipzig Drive home

2011013 Do not forget your head

1 task
$ task status:completed project:ubucon
[task ll status:completed project:ubucon]
No matches.
$ task status:completed project:ubucon completed

Complete Project     Pri  Age  Description
2011013 ubucon.workshop  5d  Prepare slides for workshop

1 task
$ task show report.ll.filter

Config Variable      Value
report.ll.filter     status:pending
```
Or ...

```
$ 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>2011016</td>
<td>4d</td>
<td>Drive home</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2011013 Do not forget your head</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>private H</td>
<td></td>
<td>2011031</td>
<td>3d</td>
<td>Pay rents</td>
<td></td>
</tr>
</tbody>
</table>

7 tasks
$ task \( \text{pro:ubucon or pro:private} \) 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>2011016</td>
<td>4d</td>
<td>Drive home</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2011013 Do not forget your head</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>private H</td>
<td></td>
<td>2011031</td>
<td>3d</td>
<td>Pay rents</td>
<td></td>
</tr>
</tbody>
</table>

2 tasks
$ # Brackets must be escaped for the shell
$ ```
Search configuration

$ task show search

<table>
<thead>
<tr>
<th>Config Variable</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>search.case.sensitive</td>
<td>yes</td>
</tr>
</tbody>
</table>

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

$ task show reg

<table>
<thead>
<tr>
<th>Config Variable</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>regex</td>
<td>no</td>
</tr>
</tbody>
</table>

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

$
Filter in reports

```
$ task show filter

<table>
<thead>
<tr>
<th>Config Variable</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>report.active.filter</td>
<td>status:pending start.any:</td>
</tr>
<tr>
<td>report.all.filter</td>
<td>status:completed</td>
</tr>
<tr>
<td>report.blocked.filter</td>
<td>status:pending depends.any:</td>
</tr>
<tr>
<td>report.completed.filter</td>
<td>status:completed</td>
</tr>
<tr>
<td>report.list.filter</td>
<td>status:pending</td>
</tr>
<tr>
<td><strong>report.ll.filter</strong></td>
<td><strong>status:pending</strong></td>
</tr>
<tr>
<td>report.long.filter</td>
<td>status:pending</td>
</tr>
<tr>
<td>report.ls.filter</td>
<td>status:pending</td>
</tr>
<tr>
<td>report.minimal.filter</td>
<td>status:pending</td>
</tr>
<tr>
<td>report.newest.filter</td>
<td>status:pending limit:10</td>
</tr>
<tr>
<td>report.next.filter</td>
<td>status:pending limit:page</td>
</tr>
<tr>
<td>report.oldest.filter</td>
<td>status:pending limit:10</td>
</tr>
<tr>
<td>report.overdue.filter</td>
<td>status:pending due.before:now</td>
</tr>
<tr>
<td>report.recurring.filter</td>
<td>status:pending parent.any:</td>
</tr>
<tr>
<td>report.unblocked.filter</td>
<td>status:pending depends.none:</td>
</tr>
<tr>
<td>report.waiting.filter</td>
<td>status:waiting</td>
</tr>
</tbody>
</table>

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

$
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