Taskwarrior Command-line to-do list

START HERE

`task` **`add`** *`<description>`*
Create a task

`task` *`<id>`* **`annotate`** *`<note>`*
Add a note

`task` **`next`**
What to do now

`task` **`project:`***`<name>`* **`list`**
Filter a report

`task` *`<id>`* **`done`**
Complete a task

`task` **`undo`**
Revert last change

`task` *`<id>`* **`modify`** *`<attrs>`*
Change a task

`task` **`sync`**
Sync with a server

pending new task → `start` → **active** has start date → `done` → **completed** history → `delete` → **deleted** kept in data

CONCEPTS

Syntax is task [filter] <command> [modifications]. The filter picks tasks, the command acts on them.

IDs are working-set numbers that change as tasks finish; use the UUID for anything lasting.

Quote text that the shell would eat: task add 'Call (Bob) & Eve'. Use -- to stop attribute parsing.

ADD AND CHANGE

`task` **`add`** *`<description>`*
Create a task

`task` **`add`** *`<desc>`* **`project:`***`<p>`* **`+`***`<tag>`* **`due:`***`<date>`* **`priority:H`**
Add with attributes

`task` *`<id>`* **`modify`** *`<attrs>`*
Change attributes

`task` *`<id>`* **`done`**
Mark completed

`task` *`<id>`* **`delete`**
Delete a task

`task` *`<id>`* **`annotate`** *`<note>`*
Attach a note

`task` *`<id>`* **`denotate`** *`<text>`*
Remove a note

`task` *`<id>`* **`start`**
Mark as active

`task` *`<id>`* **`stop`**
Stop working on it

`task` *`<id>`* **`edit`**
Edit in $EDITOR

`task` *`<id>`* **`duplicate`**
Copy a task

`task` **`log`** *`<description>`*
Record an already-done task

`task` *`<id>`* **`prepend`** *`<text>`*
Add text before description

`task` *`<id>`* **`append`** *`<text>`*
Add text after description

`task` **`undo`**
Revert the last change

`task` *`<filter>`* **`modify project:`**
Clear an attribute
Assign an empty value to remove project, due, tags etc.

REPORTS

`task` **`next`**
Most urgent tasks

REPORTS (cont.)

`task` **`list`**
All pending tasks

`task` *`<id>`* **`info`**
Full details of a task

`task` **`all`**
Every task, any status

`task` **`completed`**
Finished tasks

`task` **`waiting`**
Hidden until wait date

`task` **`overdue`**
Past their due date

`task` **`projects`**
Projects with counts

`task` **`tags`**
Tags in use

`task` **`calendar`**
Months with due dates

`task` **`summary`**
Progress per project

`task` **`reports`**
List all report names

`task` **`count`** *`<filter>`*
Count matching tasks

`task` **`_get`** *`<id>.<attr>`*
Script-friendly attribute value

`task` *`<filter>`* **`export`**
JSON output

FILTERS

`project:`*`<name>`*
Tasks in a project

`+`*`<tag>`* **`/ -`***`<tag>`*
Has / lacks a tag

`priority:H`
By priority (H, M, L)

`due:`*`<date>`*
Due on a date

`due.before:`*`<date>`*
Due earlier than

`due.after:`*`<date>`*
Due later than

`status:pending`
By status

`/`*`<regex>`***`/`**
Match description

`description.has:`*`<text>`*
Description contains

`<f1>` **`or`** *`<f2>`*
Combine with or / and / xor
Wrap groups in parentheses and quote them for the shell.

`+OVERDUE`
Virtual tag: past due

FILTERS (cont.)

`+TODAY / +WEEK`
Virtual tags: due today / this week

`+READY`
Pending and not blocked

`+BLOCKED / +BLOCKING`
Dependency state

`+ACTIVE`
Started tasks

`rc.`*`<setting>`***`=`***`<value>`**
Override config for one run

DATES

`due:tomorrow`
Relative day name

`due:eow`
End of week

`due:eom`
End of month

`due:friday`
Next named weekday

`due:2026-12-31`
Absolute ISO date

`due:now+3d`
Offset by a duration

`due:today+2w`
Weeks from today

`wait:`*`<date>`*
Hide until a date

`scheduled:`*`<date>`*
When to start working

`sow / eow / som / eom / soy / eoy`
Start or end of week, month, year

`3d / 2w / 1m / 1y`
Duration units

ORGANISE

`task` **`context define`** *`<name>`* *`<filter>`*
Save a filter as context

`task` **`context`** *`<name>`*
Activate a context

`task` **`context none`**
Deactivate context

`task` **`context list`**
Show defined contexts

`task` *`<id>`* **`modify depends:`***`<id>`*
Block on another task

`task` **`add`** *`<desc>`* **`recur:weekly due:eow`**
Recurring task

ORGANISE (cont.)

`task` **`add`** *`<desc>`* **`recur:monthly until:`***`<date>`*
Recurrence with end

`task` *`<id>`* **`modify priority:L`**
Set priority

`task` *`<id>`* **`modify +`***`<tag>`* **`-`** *`<tag>`*
Add / remove tags

`task` *`<filter>`* **`modify project:`***`<name>`*
Bulk move to a project

`task` **`project:`***`<a>.<b>`*
Sub-project by dot name

DATA AND SYNC

`task` **`sync`**
Sync with the configured server

`task` **`config`** *`<key>`* *`<value>`*
Set a taskrc value

`task` **`show`**
Show effective configuration

`task` **`diagnostics`**
Version and environment info

`task` **`export >`** *`<file>`*
Back up as JSON

`task` **`import`** *`<file>`*
Import JSON tasks

Config lives in ~/.taskrc (or ~/.config/task/taskrc); data in ~/.local/share/task. Sync settings are the sync.* keys; see task-sync(5).

Urgency is a sum of weighted coefficients (priority, due, age, tags, project). Tune with urgency.*.coefficient settings.

RECIPES

Daily triage

```
task +OVERDUE list
task +TODAY or +DUETODAY list
task next limit:5
```

Weekly project

```
task add 'Draft report'
project:work.q4 due:eow +writing
task project:work.q4 list
task 12 start
task 12 done
```

Work-only view

```
task context define work
project:work or +work
task context work
task next
task context none
```

Rent reminder

```
task add 'Pay rent'
recur:monthly due:1st
task all +CHILD
task 14 modify due:2nd
```