Taskwarrior Command-line to-do list

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.

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

ADD AND CHANGE (cont.)

`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 undo`
Revert the last change

`task <filter> modify project:`
Clear an attribute

REPORTS

`task next`
Most urgent tasks

`task list`
All pending tasks

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

`task all`
Every task, any status

`task completed`
Finished tasks

REPORTS (cont.)

`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

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

`status:pending`
By status

`/<regex>/`
Match description

`<f1> or <f2>`
Combine with or / and / xor

`+OVERDUE`
Virtual tag: past due

FILTERS (cont.)

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

`+READY`
Pending and not blocked

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

`wait:<date>`
Hide until a date

ORGANISE

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

`task context <name>`
Activate a context

`task context none`
Deactivate context

ORGANISE (cont.)

`task context list`
Show defined contexts

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

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

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

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

DATA AND SYNC

`task sync`
Sync with the configured server

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

`task show`
Show effective configuration

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