Task YAML task runner

CONCEPTS

Running task with no name runs the task called default. Tasks live in Taskfile.yml, found by searching upwards from the current directory.

task <name> CLI call
↓ deps
Dependencies run first, in parallel
↓ up-to-date?
Check sources / status
↓ cmds
Commands run in order

Variables are templated with {{.NAME}}; environment variables use $NAME in the shell.

RUNNING

`task --list`
List tasks that have a desc

`task --list-all`
List all tasks, even without desc

`task <name>`
Run a task

`task --dry <name>`
Print commands without executing

`task --force <name>`
Run even if up to date

`task --parallel <a> <b>`
Run the named tasks concurrently

`task --summary <name>`
Show the task's summary text

`task --watch <name>`
Re-run when sources change

`task --taskfile <file>`
Use a different Taskfile

`task --silent <name>`
Do not echo commands

`task --init`
Create a starter Taskfile.yml

TASK BASICS

`version: '3'`
Required first line

`tasks: <name>: cmds:`
Define a task and its commands

TASK BASICS (cont.)

`desc: <text>`
Text shown in --list

`- task: <other>`
Call another task as a command

`default:`
Task run by plain task

DEPENDENCIES AND LOOPS

`deps: [<a>, <b>]`
Run these first, concurrently

`for: [<a>, <b>] cmd: ... {{.ITEM}}`
Loop a command over values

VARIABLES

`vars: <NAME>: <value>`
Define a variable

`{{.NAME}}`
Use a variable in a command

`<NAME>: {sh: <command>}`
Variable from command output

`{{.CLI_ARGS}}`
Everything after -- on the CLI

`env: <NAME>: <value>`
Export an environment variable

`dotenv: ['.env']`
Load env vars from files

UP-TO-DATE AND CHECKS

`sources: ['**/*.go']`
Files whose change triggers a run

`generates: [<file>]`
Files the task produces

`status: [<test cmd>]`
Skip when all commands exit 0

`preconditions: - sh: <test> msg: <text>`
Abort with a message if the test fails

INCLUDES AND LAYOUT

`includes: <ns>: <path>`
Import tasks as ns:<task>

`task <ns>:<task>`
Run a namespaced task