mise Tool versions, env and tasks

CONCEPTS

mise manages tool versions, env variables and tasks from one mise.toml per project. Commit it so everyone gets the same setup.

mise.toml [tools] [env] [tasks]
↓ `mise install`
Installed tools in the mise data dir
↓ `activate or shims`
PATH + env per directory
↓ `mise run`
Tasks run with these tools
What mise does in a project

Activation (mise activate in your shell rc) updates PATH on every prompt. Shims suit IDEs and scripts that skip rc files; mise exec and mise run need neither.

Config with [env], hooks or templates loads only once trusted. mise use, set, install, exec and run trust it for you; for a cloned repo read it, then mise trust.

INSTALL AND ACTIVATE

`curl -fsSL https://mise.run | sh`
Install mise to ~/.local/bin

`eval "$(mise activate bash)"`
Activate in bash (put in ~/.bashrc)

`eval "$(mise activate zsh)"`
Activate in zsh (put in ~/.zshrc)

`mise activate fish | source`
Activate in fish (config.fish)

`mise self-update`
Update mise itself

TOOLS

`mise use <tool>@<version>`
Install and add to mise.toml

`mise use -g <tool>@<version>`
Same, in the global config

`mise use --pin <tool>@<version>`
Record the exact resolved version

TOOLS (cont.)

`mise use --remove <tool>`
Drop a tool from the config

`mise install`
Install every tool the config lists

`mise install <tool>@<version>`
Install without touching config

`mise ls`
List installed and active tools

`mise ls --missing`
Requested but not installed

`mise ls-remote <tool>`
Versions available to install

`mise latest <tool>`
Latest version matching a request

`mise outdated`
Show tools with newer versions

`mise upgrade`
Upgrade within configured ranges

`mise upgrade --bump`
Upgrade and rewrite the versions

`mise prune`
Delete versions nothing uses

`mise uninstall <tool>@<version>`
Remove an installed version

`mise search <text>`
Search for available tools

`mise which <bin>`
Path of the executable mise resolves

RUNNING WITH TOOLS

`mise exec <tool>@<version> -- <cmd>`
Run a command with that tool

`mise x -- <cmd>`
Run with the project's tools and env

`mise shell <tool>@<version>`
Use a version in this shell only

ENVIRONMENT

`mise set <KEY>=<value>`
Set a variable in mise.toml

`mise set`
List variables and their sources

ENVIRONMENT (cont.)

`mise unset <KEY>`
Remove a variable from mise.toml

`mise env`
Print the env mise would set

TASKS

`mise run <task>`
Run a task and its dependencies

`mise <task>`
Shorthand for mise run <task>

`mise tasks`
List tasks

`mise run <a> ::: <b>`
Run several tasks in parallel

`mise run 'test:*'`
Run tasks matching a pattern

`mise run <task> <args>`
Pass arguments to a task

`mise run -n <task>`
Show run order, execute nothing

`mise watch <task>`
Rerun a task when files change

Task fields in [tasks.<name>]

Field	Effect
`run`	Command, or array run in order
`description`	Shown in mise tasks
`depends`	Tasks that must succeed first
`alias`	Short name for mise run
`dir`	Working directory
`env`	Variables for this task only
`sources / outputs`	Skip when outputs are current
`hide`	Hide from the list, still runnable
`confirm`	Ask before running

CONFIG AND TRUST

`mise trust`
Trust the nearest untrusted config

`mise config`
List the config files in use

`mise doctor`
Check the setup for problems