# Cabal Haskell build tool

## CONCEPTS

> Commands work on a project: a cabal.project (or just one .cabal file) and the packages it lists. Builds land in dist-newstyle.

## PROJECTS

**`cabal init --interactive`**
Create a package with prompts

## BUILD

**`cabal build`**
Build all default targets

**`cabal build`** *`<target>`*
Build one package or component

**`cabal build all`**
Build every package in the project

**`cabal build -w ghc-`***`<version>`*
Use a specific compiler

## RUN & REPL

**`cabal run`**
Run the only executable

**`cabal run`** *`<exe>`* **`--`** *`<args>`*
Run an executable with arguments

**`cabal repl`**
Open GHCi on the default target

## TEST & DOCS

**`cabal test`**
Build and run all test suites

## DEPENDENCIES

**`cabal update`**
Download the latest package index

**`cabal freeze`**
Write cabal.project.freeze

## INSTALL

**`cabal install`** *`<exe>`*
Install an executable from Hackage

## PACKAGE FILE

### .cabal stanzas

| Stanza | Use |
|---|---|
| `library` | Reusable code; list exposed-modules |
| `executable <name>` | Program; needs main-is |
| `test-suite <name>` | Tests; type: exitcode-stdio-1.0 |
| `benchmark <name>` | Benchmarks; same form as tests |
| `common <name>` | Shared fields, pulled in with import: |

### Common fields

| Field | Meaning |
|---|---|
| `cabal-version` | Spec version, e.g. 3.0 |
| `build-depends` | Packages with version bounds |
| `hs-source-dirs` | Where the .hs files live |
| `default-language` | GHC2021 or Haskell2010 |
| `ghc-options` | Flags such as -Wall |
| `other-modules` | Internal, not exported modules |

### .cabal skeleton

```
cabal-version: 3.0
name:          demo
version:       0.1.0.0
build-type:    Simple

common shared
    default-language: GHC2021
    ghc-options:      -Wall
    build-depends:    base <5

library
    import:           shared
    hs-source-dirs:   src
    exposed-modules:  Demo

executable demo
    import:           shared
    hs-source-dirs:   app
    main-is:          Main.hs
    build-depends:    demo
```

## CABAL.PROJECT

**`packages: .`**
List the packages in the project