rebar3 Erlang build tool

CONCEPTS

rebar3 is the Erlang build tool: one rebar.config per project, builds land in _build/<profile>. Run rebar3 help to list tasks, including plugin ones.

rebar.config deps, profiles
↓ compile
rebar.lock pinned deps
↓ build
_build/<profile> beams, deps, rel
↓ release / tar
Release relx, start script

Profiles pick the build variant: default, test (used by eunit and ct) and prod. Select with rebar3 as <profile> <task>.

PROJECTS

`rebar3 new app <name>`
OTP application with supervisor

`rebar3 new lib <name>`
Library without a supervisor

`rebar3 new release <name>`
Umbrella project for releases

`rebar3 new escript <name>`
Command-line escript project

BUILD & RUN

`rebar3 compile`
Fetch deps and compile all apps

`rebar3 clean`
Remove compiled beams

`rebar3 shell`
Shell with apps and deps loaded

`rebar3 shell --sname <name>`
Shell as a named node

`rebar3 shell --apps <a>,<b>`
Start only these apps

`rebar3 shell --config <file>`
Use a sys.config file

`rebar3 do <task>,<task>`
Run several tasks in a row

`rebar3 escriptize`
Package an escript executable

`rebar3 version`
Show rebar3 and Erlang versions

TEST

`rebar3 eunit`
Run all EUnit tests

`rebar3 eunit --module=<mod>`
Tests of one module

`rebar3 eunit --test=<mod>:<fun>`
Run one test function

`rebar3 eunit --application=<app>`
Tests of one app

`rebar3 eunit -v`
Verbose output

`rebar3 ct`
Run all Common Test suites

`rebar3 ct --suite=<path>`
Run one suite

`rebar3 ct --suite=<path> --group=<g>`
Run one test group

`rebar3 ct --suite=<path> --case=<c>`
Run one test case

`rebar3 ct --sys_config=<file>`
Use a sys.config for tests

`rebar3 eunit --cover`
Collect coverage data

DEPENDENCIES

`rebar3 deps`
List deps and lock status

`rebar3 tree`
Show the dependency tree

`rebar3 upgrade`
Upgrade all dependencies

`rebar3 upgrade <dep>`
Upgrade one dependency

deps in rebar.config

```
{deps, [
  cowboy,
  {jsx, "~> 3.1"},
  {mylib, {git, "<url>", {tag,
"1.0.0"}}},
  {other, {git, "<url>", {branch,
"main"}}}
]}.
```

List every dependency in the .app.src applications list too, or it will not start with your app.

PROFILES

`rebar3 as <profile> <task>`
Run a task in a profile

`rebar3 as <p1>,<p2> <task>`
Combine profiles

profiles in rebar.config

```
{profiles, [
  {test, [{deps, [meck]}]},
  {prod, [{erl_opts,
[no_debug_info]}]}
]}.
```

QUALITY

`rebar3 dialyzer`
Build the PLT and check types

`rebar3 xref`
Cross-reference check

RELEASES

`rebar3 release`
Build a dev release

`rebar3 as prod release`
Release with ERTS, no symlinks

`rebar3 as prod tar`
Release as a .tar.gz

`_build/<profile>/rel/<name>/bin/<name> console`
Start a release in a shell

`bin/<name> foreground`
Run in foreground (systemd, Docker)

`bin/<name> remote_console`
Attach to a running node

relx in rebar.config

```
{relx, [
  {release, {<name>, "0.1.0"},
[<name>, sasl]},
  {dev_mode, true},
  {include_erts, false},
  {extended_start_script, true}
]}.
```

CONFIG & PLUGINS

Plugins

```
{plugins, [rebar3_auto]}.
{project_plugins, [erlfmt]}.
```

HEX

`rebar3 hex publish`
Publish package and docs