Meson Fast build system

CONCEPTS

Meson generates build files (Ninja by default) and always builds out of source: meson.build stays in the source tree, every build directory is separate and can use different options.

meson.build project definition
↓ meson setup
Build dir options + Ninja files
↓ meson compile
Artifacts targets
↓ meson test
Test results testlog
↓ meson install
Install tree prefix / DESTDIR

Configure once per build dir, then build, test and install from it

Edits to meson.build trigger an automatic reconfigure on the next compile. Options set with -D are remembered in the build dir.

SETUP

meson setup *<builddir>*
Configure source in cwd into builddir

meson setup *<builddir>* *<srcdir>*
Configure another source dir

meson setup *<builddir>* **--buildtype=***<type>*
debug, debugoptimized, release, plain

meson setup *<builddir>* **--prefix=***<dir>*
Set the install prefix

meson setup *<builddir>* **-D***<opt>***=***<val>*
Set a project or built-in option

meson setup *<builddir>* **--reconfigure**
Re-run configure with same options

meson setup *<builddir>* **--wipe**
Delete and recreate the build dir

meson setup *<builddir>* **--cross-file** *<file>*
Cross-compile with a machine file

CONFIGURE

meson configure *<builddir>*
List all options and values

meson configure *<builddir>* **-D***<opt>***=***<val>*
Change an option later

-Dwarning_level=*<0-3|everything>*
Compiler warning level

-Dwerror=true
Treat warnings as errors

-Db_sanitize=*<kind>*
address, undefined, thread

BUILD

meson compile -C *<builddir>*
Build all targets

meson compile -C *<builddir>* *<target>*
Build one target

meson compile -C *<builddir>* **-v**
Show full command lines

meson compile -C *<builddir>* **--clean**
Remove build outputs

TEST

meson test -C *<builddir>*
Run all tests

meson test -C *<builddir>* **--print-errorlogs**
Show output of failures

meson test -C *<builddir>* *<name>*
Run tests by name

meson test -C *<builddir>* **--suite** *<suite>*
Run one suite

meson test -C *<builddir>* **--gdb** *<name>*
Run one test under gdb

INSTALL AND DIST

meson install -C *<builddir>*
Install to the prefix

meson install -C *<builddir>* **--destdir** *<dir>*
Stage into a root dir

INSTALL AND DIST (cont.)

meson dist -C *<builddir>*
Create a release tarball

SUBPROJECTS AND WRAPS

meson wrap search *<name>*
Search WrapDB

meson wrap install *<name>*
Add a wrap file to subprojects/

INTROSPECT AND TOOLS

meson introspect *<builddir>* **--targets**
List build targets (JSON)

meson devenv -C *<builddir>*
Shell with the build's environment

MESON.BUILD

project('*<name>***', 'c', version: '***<x.y>***')**
Declare project and languages

executable('*<name>***', '***<main.c>***')**
Define an executable

library('*<name>***', *<sources>*)**
Shared or static per default_library

dependency('*<name>***')**
Find a lib via pkg-config/CMake

dependency('*<n>***', required: false)**
Optional dependency

include_directories('*<dir>***')**
Add include paths

subdir('*<dir>***')**
Descend into a directory

test('*<name>***', *<exe>*)**
Register a test

declare_dependency(link_with: *<lib>*)
Expose a library to parents

meson_options.txt
Project-defined options file

option('*<name>***', type: 'boolean', value: true)**
Declare an option