NAME

rustc − The Rust compiler

SYNOPSIS

rustc [*OPTIONS*] *INPUT*

DESCRIPTION

This program is a compiler for the Rust language, available at https://www.rust−lang.org.

OPTIONS

−h, **−−help**
Display the help message.

−−cfg *SPEC*
Configure the compilation environment.

−L [*KIND*=]*PATH*
Add a directory to the library search path. The optional *KIND* can be one of:

dependency
only lookup transitive dependencies here

crate only lookup local ‘extern crate‘ directives here

native only lookup native libraries here

framework
only look for OSX frameworks here

all look for anything here (the default)

−l [*KIND*=]*NAME*
Link the generated crate(s) to the specified library *NAME*. The optional *KIND* can be one of *static*, *dylib*, or *framework*. If omitted, *dylib* is assumed.

−−crate−type [bin|lib|rlib|dylib|cdylib|staticlib|proc−macro]
Comma separated list of types of crates for the compiler to emit.

−−crate−name *NAME*
Specify the name of the crate being built.

−−emit [asm|llvm−bc|llvm−ir|obj|metadata|link|dep−info|mir][=*PATH*]
Configure the output that **rustc** will produce. Each emission may also have an optional explicit output *PATH* specified for that particular emission kind. This path takes precedence over the **-o** option.

−−print [crate−name|file−names|sysroot|target−libdir|cfg|target−list|target−cpus|target−features|relocation−models|code−models|tls−models|target−spec−json|native−static−libs|stack−protector−strategies|link−args]
Comma separated list of compiler information to print on stdout.

−g Equivalent to *−C debuginfo=2.*

−O Equivalent to *−C opt−level=2.*

−o *FILENAME*
Write output to *FILENAME*. Ignored if multiple *−−emit* outputs are specified which don’t have an explicit path otherwise.

−−out−dir *DIR*
Write output to compiler-chosen filename in *DIR*. Ignored if *−o* is specified. Defaults to the current directory.

−−explain *OPT*
Provide a detailed explanation of an error message.

−−test Build a test harness.

−−target *TARGET*
Target triple for which the code is compiled. This option defaults to the host's target triple. The target triple has the general format <arch><sub>−<vendor>−<sys>−<abi>, where:

<arch> x86, arm, thumb, mips, etc.

<sub> for example on ARM: v5, v6m, v7a, v7m, etc.

<vendor>
pc, apple, nvidia, ibm, etc.

<sys> none, linux, win32, darwin, cuda, etc.

<abi> eabi, gnu, android, macho, elf, etc.

−W help
Print 'lint' options and default settings.

−W *OPT*, **−−warn** *OPT*
Set lint warnings.

−A *OPT*, **−−allow** *OPT*
Set lint allowed.

−D *OPT*, **−−deny** *OPT*
Set lint denied.

−F *OPT*, **−−forbid** *OPT*
Set lint forbidden.

−C *FLAG*[=*VAL*], **−−codegen** *FLAG*[=*VAL*]
Set a codegen-related flag to the value specified. Use *−C help* to print available flags. See CODEGEN OPTIONS below.

−V, **−−version**
Print version info and exit.

−v, **−−verbose**
Use verbose output.

−−remap−path−prefix *from*=*to*
Remap source path prefixes in all output, including compiler diagnostics, debug information, macro expansions, etc. The *from*=*to* parameter is scanned from right to left, so *from* may contain '=', but *to* may not.

This is useful for normalizing build products, for example by removing the current directory out of pathnames emitted into the object files. The replacement is purely textual, with no consideration of the current system's pathname syntax. For example *−−remap−path−prefix foo*=*bar* will match **foo/lib.rs** but not **./foo/lib.rs**.

−−extern *NAME*=*PATH*
Specify where an external rust library is located. These should match *extern* declarations in the crate's source code.

−−sysroot *PATH*
Override the system root.

−Z *FLAG*
Set unstable / perma-unstable options. Use *−Z help* to print available options.

−−color auto|always|never
Configure coloring of output:

auto colorize, if output goes to a tty (default);

always always colorize output;

never never colorize output.

CODEGEN OPTIONS

linker=*/path/to/cc*
Path to the linker utility to use when linking libraries, executables, and objects.

link−args='*−flag1 −flag2*'
A space-separated list of extra arguments to pass to the linker when the linker is invoked.

lto Perform LLVM link-time optimizations.

target−cpu=*help*
Selects a target processor. If the value is 'help', then a list of available CPUs is printed.

target−feature='*+feature1*,*−feature2*'
A comma-separated list of features to enable or disable for the target. A preceding '+' enables a feature while a preceding '−' disables it. Available features can be discovered through *llc -mcpu=help*.

passes=*val*
A space-separated list of extra LLVM passes to run. A value of 'list' will cause **rustc** to print all known passes and exit. The passes specified are appended at the end of the normal pass manager.

llvm−args='*−arg1 −arg2*'
A space-separated list of arguments to pass through to LLVM.

save−temps
If specified, the compiler will save more files (.bc, .o, .no−opt.bc) generated throughout compilation in the output directory.

rpath If specified, then the rpath value for dynamic libraries will be set in either dynamic library or executable outputs.

no−prepopulate−passes
Suppresses pre-population of the LLVM pass manager that is run over the module.

no−vectorize−loops
Suppresses running the loop vectorization LLVM pass, regardless of optimization level.

no−vectorize−slp
Suppresses running the LLVM SLP vectorization pass, regardless of optimization level.

soft−float
Generates software floating point library calls instead of hardware instructions.

prefer−dynamic
Prefers dynamic linking to static linking.

no−integrated−as
Force usage of an external assembler rather than LLVM's integrated one.

no−redzone
Disable the use of the redzone.

relocation−model=[pic,static,dynamic−no−pic]
The relocation model to use. (Default: *pic*)

code−model=[small,kernel,medium,large]
Choose the code model to use.

metadata=*val*
Metadata to mangle symbol names with.

extra−filename=*val*
Extra data to put in each output filename.

codegen−units=*n*
Divide crate into *n* units to optimize in parallel.

remark=*val*
Print remarks for these optimization passes (space separated, or "all").

no−stack−check
Disable checks for stack exhaustion (a memory-safety hazard!).

debuginfo=*val*
Debug info emission level:

0 no debug info;

1 line-tables only (for stacktraces and breakpoints);

2 full debug info with variable and type information.

opt−level=*VAL*
Optimize with possible levels 0–3, s (optimize for size), or z (for minimal size)

ENVIRONMENT

Some of these affect only test harness programs (generated via rustc --test); others affect all programs which link to the Rust standard library.

RUST_TEST_THREADS
The test framework Rust provides executes tests in parallel. This variable sets the maximum number of threads used for this purpose. This setting is overridden by the --test-threads option.

RUST_TEST_NOCAPTURE
If set to a value other than "0", a synonym for the --nocapture flag.

RUST_MIN_STACK
Sets the minimum stack size for new threads.

RUST_BACKTRACE
If set to a value different than "0", produces a backtrace in the output of a program which panics.

EXAMPLES

To build an executable from a source file with a main function:

```
$ rustc −o hello hello.rs
```

To build a library from a source file:

```
$ rustc −−crate−type=lib hello−lib.rs
```

To build either with a crate (.rs) file:

```
$ rustc hello.rs
```

To build an executable with debug info:

```
$ rustc −g −o hello hello.rs
```

SEE ALSO

rustdoc(1)

BUGS

See https://github.com/rust−lang/rust/issues for issues.

AUTHOR

See https://github.com/rust−lang/rust/graphs/contributors or use ‘git log --all --format=’%cN <%cE>’ | sort -u‘ in the rust source distribution.

COPYRIGHT

This work is dual-licensed under Apache 2.0 and MIT terms. See *COPYRIGHT* file in the rust source distribution.