

Linux

GCC Flags for Gentoo

```
- - ada : Build the ADA language (GNAT) frontend
+ + cet : (Restricted to >=sys-devel/gcc-10)
           Enable support for control flow hijacking protection. On amd64, this
provides Intel Control
           Flow Enforcement Technology (CET). On arm64, this provides Branch
Target Identification
           (BTI) and Pointer Authentication Code (PAC) support. This is only
effective on amd64 or
           arm64. Only provides benefits on newer CPUs. For Intel, the CPU must
be at least as new as
           Tiger Lake. For AMD, it must be at least as new as Zen 3. This is
harmless on older CPUs,
           but provides no benefit either. For ARM64, PAC was introduced in
armv8.3-a, and BTI was
           introduced in armv8.5-a. When combined with USE=hardened on amd64,
GCC will set
           -fcf-protection by default when building software. The effect is
minimal on systems which do
           not support it, other than a possible small increase in codesize for
the NOPs. The generated
           code is therefore compatible with i686 at the earliest. On arm64, GCC
will set
           -mbranch-protection=standard by default when building software.
- - d : Enable support for the D programming language
- - debug : Enables GCC's 'checking' facility via --enable-checking=yes,extra,rtl.
This adds checks to
           various compiler passes for integrity and input validation. This can
help catch possible
           miscompilations early as well as latent bugs which could become real
problems in future, but
           at the cost of slower compile times when using GCC. Unrelated to
backtraces.
- - doc : Add extra documentation (API, Javadoc, etc). It is recommended to enable
per package instead
           of globally
+ + fortran : Add support for fortran
- - go : Build the GCC Go language frontend.
+ + graphite : Add support for the framework for loop optimizations based on a
polyhedral intermediate
           representation
- - hardened : Activate default security enhancements for toolchain (gcc, glibc,
binutils)
+ + jit : Enable libgccjit so other applications can embed gcc for Just-In-Time
compilation.
+ + lto : Build using Link Time Optimizations (LTO). Note that GCC is always built
with support for
           building other programs with LTO. This USE flag is for whether GCC
itself is built and
           optimized with LTO.
- - modula2 : Build the GCC Modula-2 language frontend.
+ + nls : Add Native Language Support (using gettext - GNU locale utilities)
+ + objc : Build support for the Objective C code language
+ + objc++ : Build support for the Objective C++ language
- - objc-gc : Build support for the Objective C code language Garbage Collector
+ + openmp : Build support for the OpenMP (support parallel computing), requires
>=sys-devel/gcc-4.2
           built with USE="openmp"
```

Linux

+ + pgo : Build GCC using Profile Guided Optimization (PGO). GCC will build itself and then analyze the just-built binary and then rebuild itself using the data obtained from analysis of codepaths taken. It does not affect whether GCC itself supports PGO when building other software. This substantially increases the build time needed for building GCC itself.

- - rust : Build support for the Rust language, installs gccrs.

+ + sanitize : Build support for various sanitizer functions (ASAN/TSAN/etc...) to find runtime problems in applications.

+ + ssp : Build packages with stack smashing protection on by default

- - systemtap : enable systemtap static probe points

- - test : Enable dependencies and/or preparations necessary to run tests (usually controlled by FEATURES=test but can be toggled independently)

- - valgrind : Enable annotations for accuracy. May slow down runtime slightly. Safe to use even if not currently using dev-debug/valgrind

- - vanilla : Do not add extra patches which change default behaviour; DO NOT USE THIS ON A GLOBAL SCALE as the severity of the meaning changes drastically

- - vtv : Build support for virtual table verification (a C++ hardening feature). This does not control whether GCC defaults to using VTV> Note that actually using VTV breaks ABI and hence the whole system must be built with -fvtable-verify.

+ + zstd : Enable support for ZSTD compression

μ μ ::

: #1015

:: Kosmas

μ : 2026-05-11 14:26