



Demo project

Release 13.0.1 (experimental 20220101)

Martin Liska

Nov 08, 2022

CONTENTS

1 Comparison of GCC docs in Texinfo and Sphinx	1
1.1 HTML output	1
1.2 PDF output	6
2 Introduction	9
3 Multiple	13
3.1 Arithmetic functions	13
3.2 Names:	13
4 Tools/packages necessary for building GCC	15
5 Formatting	17
6 Changes, New Features, and Fixes	19
6.1 Caveats	19
6.2 General Improvements	19
6.3 New Languages and Language specific improvements	20
6.4 New Targets and Target Specific Improvements	21
6.5 Operating Systems	22
6.6 Other significant improvements	22
7 C++ Standards Support in GCC	23
7.1 C++23 Support in GCC	23
7.2 C++23 Language Features	24
7.3 C++20 Support in GCC	26
7.4 C++20 Language Features	26
7.5 C++17 Support in GCC	32
7.6 C++17 Language Features	32
7.7 Technical Specifications	35
7.8 C++14 Support in GCC	35
7.9 C++14 Language Features	35
7.10 C++11 Support in GCC	37
7.11 C++98 Support in GCC	40
Index	41

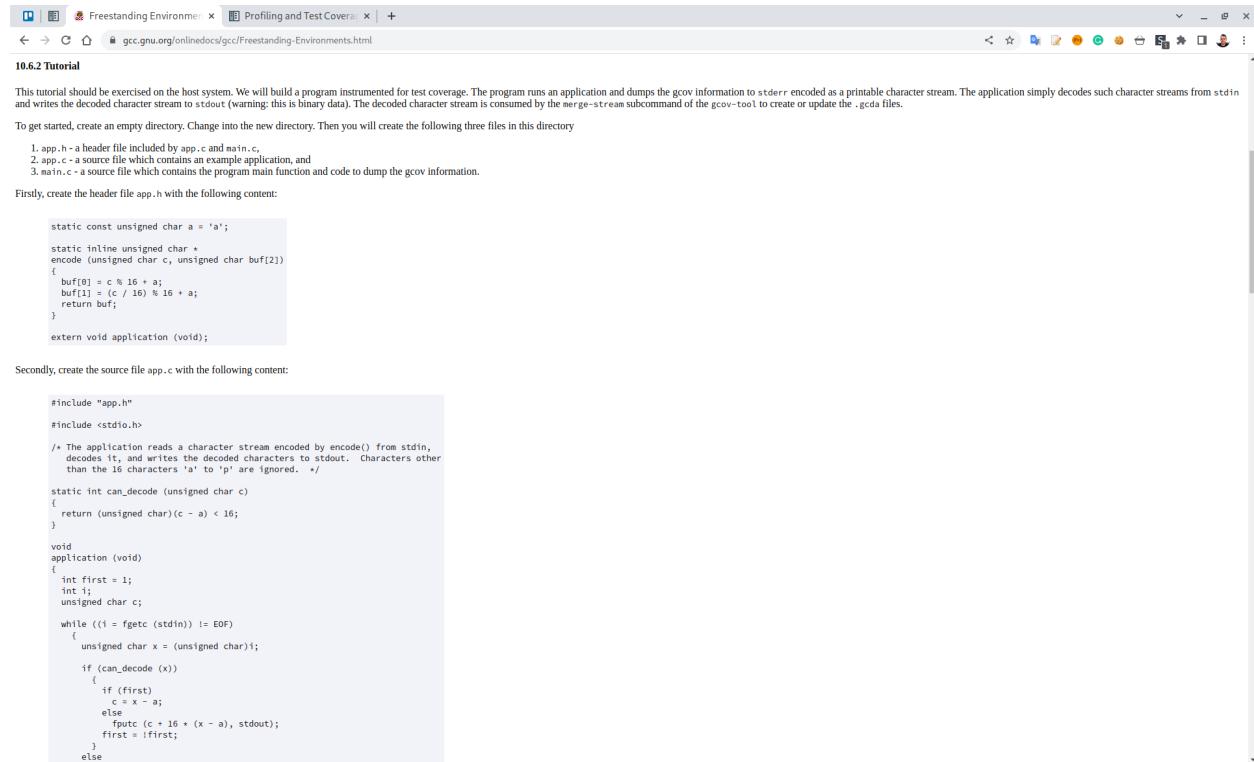
COMPARISON OF GCC DOCS IN TEXINFO AND SPHINX

1.1 HTML output

1.1.1 Formatting

Sphinx provides left menu navigation and side bar navigation (for subsections) on a page. Moreover, it's using a richer CSS formats including the code highlighting for code snippets, bash sessions or a JSON output.

Texinfo version:



The screenshot shows a web browser window displaying the Texinfo version of the GCC documentation. The title bar reads "Freestanding Environments x Profiling and Test Coverage x +". The address bar shows the URL "gcc.gnu.org/onlinedocs/gcc/Freestanding-Environments.html". The main content area displays the "10.6.2 Tutorial" section. The text describes a tutorial for building a program instrumented for test coverage, mentioning files like app.h, app.c, and main.c. It includes a code snippet for the application function and a note about ignoring characters other than 'a' to 'p'. Below this, another code snippet shows the implementation of can_decode and application functions, including file I/O and character decoding logic.

```
static const unsigned char a = 'a';

static inline unsigned char *
encode (unsigned char c, unsigned char buf[2])
{
  buf[0] = c % 16 + a;
  buf[1] = (c / 16) % 16 + a;
  return buf;
}

extern void application (void);

#include "app.h"
#include <stdio.h>

/* The application reads a character stream encoded by encode() from stdin,
   decodes it, and writes the decoded characters to stdout. Characters other
   than the 16 characters 'a' to 'p' are ignored. */
static int can_decode (unsigned char c)
{
  return (unsigned char)(c - a) < 16;
}

void
application (void)
{
  int first = 1;
  int i;
  unsigned char c;
  while ((i = fgetc (stdin)) != EOF)
  {
    unsigned char x = (unsigned char)i;
    if (can_decode (x))
    {
      if (first)
        c = x - a;
      else
        fputc (c + 16 * (x - a), stdout);
      first = !first;
    }
  }
}
```

Sphinx version (see Profiling and Test Coverage in Freestanding Environments):

Demo project, Release 13.0.1 (experimental 20220101)

The screenshot shows a web browser window with the title "Freestanding Environments" and "Profiling and Test Coverage". The URL is /home/maxin/Programming/txuzst-generated/sphinx/gccl_build/html/gcov/profiling-and-test-coverage-in-freestanding-environments.html#tutorial. The main content area is titled "Tutorial" and contains text about instrumenting programs for test coverage. It includes a bulleted list of files: app.h, app.c, and main.c. Below this, there are two code blocks: one for app.h containing a hex-to-decimal conversion function, and one for app.c containing a main function that reads from stdin, calls encode(), and writes to stdout. The sidebar on the left lists various GCC documentation topics, and the right sidebar shows a "CONTENTS" tree with "Overview", "Tutorial", and "System Initialization Caveats".

```
static const unsigned char a = 'a';
static inline unsigned char *
encode (unsigned char c, unsigned char buf[])
{
  buf[0] = c % 16 + a;
  buf[1] = (c / 16) % 16 + a;
  return buf;
}

extern void application (void);
```

```
#include "app.h"
#include <stdio.h>

/* The application reads a character stream encoded by encode() from stdin,
   decodes it, and writes the decoded characters to stdout. Characters other
   than the 16 characters 'a' to 'p' are ignored. */

static int can_decode (unsigned char c)
{
  return (unsigned char)(c - a) < 16;
```

1.1.2 Links

Entire manual can be easily visited with cross-references for thinks like options, attributes, etc.

Texinfo version:

Demo project, Release 13.0.1 (experimental 20220101)

-02

Optimize even more. GCC performs nearly all supported optimizations that do not involve a space-speed tradeoff. As compared to -O, this option increases both compilation time and the performance of the generated code.

-O2 turns on all optimization flags specified by -O1. It also turns on the following optimization flags:

```
-falign-functions -falign-jumps  
-falign-labels -falign-loops  
-fcaller-saves  
-fcode-coverage  
-fcommon-jumps  
-fcsse-follow-jumps -fcse-skip-blocks  
-fdelete-null-pointer-checks  
-fdevirtualize -fdevirtualize-speculatively  
-fexpensive-optimizations  
-ffinite-math-mode  
-fgcse -fgcse-lm  
-fhoist-adjacent-loads  
-finline-functions  
-finline-small-functions  
-findirect-inlining  
-fipa-bit-cp -fipa-cp -fipa-icf  
-fipa-ra -fipa-sra -fipa-vrp  
-fisolate-erroneous-paths-dereference  
-fipa-rename  
-foptimize-sibling-calls  
-foptimize-strlen  
-fpartial-inlining  
-fpeephole  
-freorder-blocks-algorithm=stc  
-freorder-blocks-and-partition -freorder-functions  
-frerun-cse-after-loop  
-fschedule-insns -fschedule-insns2  
-fscattered-interlock -fsched-spec  
-fstore-merging  
-fstrict-aliasing  
-fthread-jumps  
-ftrivial-within-call-dce  
-ftree-loop-vectorize  
-ftree-pre  
-ftree-slp-vectorize  
-ftree-switch-conversion -ftree-tail-merge  
-ftree-vrp  
-fvector-cost-model=very-cheap
```

Please note the warning under -fgcse about invoking -O2 on programs that use computed gotos.

-03

Optimize yet more. -O3 turns on all optimizations specified by -O2 and also turns on the following optimization flags:

```
-fgcse-after-reload  
-fipa-cp-clone  
-floop-interchange  
-floop-opt-cell-and-jam  
-fpeel-loops  
-fpredictive-commoning  
-fsplit-loops  
-funsafe-loop-optimizations
```

Sphinx version (see e.g. -02):

Using the GNU Compiler Collection 13.0.1 (experimental 20220101) documentation

Search

Copyright

Introduction

Programming Languages Supported by GCC

Language Standards Supported by GCC

Description

Option Summary

Options Controlling the Kind of Output

Compiling C++ Programs

Options Controlling C Dialect

Options Controlling C++ Dialect

Options Controlling Objective-C and Objective-C++ Dialects

Options to Control Diagnostic Messages Formatting

Options to Request or Suppress Warnings

Options That Control Static Analysis

Options for Debugging Your Program

Options That Control Optimization

Program Instrumentation

Program Options

-O2

Optimize even more. GCC performs nearly all supported optimizations that do not involve a space-speed tradeoff. As compared to `-O1`, this option increases both compilation time and the performance of the generated code.

`-O2` turns on all optimization flags specified by `-O1`. It also turns on the following optimization flags:

```
-falign-functions -falign-jumps -falign-labels -falign-loops -fcaller-saves  
-fcode-hoisting -fcrossjumping -fcse-follow-jumps -fcse-skip-blocks  
-fdelete-null-pointer-checks -fdevirtualize -fdevirtualize-speculatively  
-fexpensive-optimizations -ffinite-loops -fgcse -fgcse-lm -fhoist-adjacent-loads  
-finline-functions -finline-small-functions -findirect-inlining -fipa-bit-cp -fipa-cp  
-fipa-icf -fipa-ra -fipa-sra -fipa-vrp -fisolate-errorneous-paths-dereference  
-fira-remat -foptimize-sibling-calls -foptimize-stren -fpartial-inlining -fpeephole2  
-freorder-blocks-algorithm =stc -freorder-blocks-and-partition -freorder-functions  
-frerun-cse-after-loop -fschedule-insns -fschedule-insns2 -fsched-interblock  
-fsched-spec -fstore-merging -fstrict-aliasing -fthread-jumps -ftree-builtin-call-dce  
-ftree-loop-vectorize -ftree-pre -ftree-slp-vectorize -ftree-switch-conversion  
-ftree-tail-merge -ftree-vrp -fvect-cost-model =very-cheap
```

Please note the warning under `-fgcse` about invoking `-O2` on programs that use computed gotos.

-O3

Optimize yet more. `-O3` turns on all optimizations specified by `-O2` and also turns on the following optimization flags:

```
-fgcse-after-reload -fipa-cp-clone -floop-interchange -floop-unroll-and-jam  
-fpeel-loops -fpredictive-commoning -fsplit-loops -fsplit-paths  
-ftree-loop-distribution -ftree-partial-pre -funswitch-loops -fvect-cost-model =dynamic  
-fversion-loops-for-strides
```

-Os

Reduce compilation time and make debugging produce the expected results. This is the default.

-O1

Optimize for size. `-Os` enables all `-O2` optimizations except those that often increase code size:

```
-falign-functions -falign-jumps -falign-labels -falign-loops -fprefetch-loop-arrays  
-freorder-blocks-algorithm =stc
```

It also enables `-finline-functions`, causes the compiler to tune for code size rather than execution speed, and performs further optimizations designed to reduce code size.

1.1.3 Built-in Search Engine

Sphinx provides built-in search engine for a fast navigation:

The screenshot shows a web browser window with the address bar containing `/home/marxin/Programming/tex2rst-generated/sphinx/gcc/_build/html/search.html?q=stack+protector&check_keywords=yes&area=default`. The page title is "Search Results". The search query "stack protector" has found 22 pages. The results list includes various GCC options and attributes related to stack protection, such as `-fstack-protector`, `-fstack-protector-all`, `-fstack-protector-explicit`, `-fstack-protector-strong`, `-Wno-stack-protector`, `-Wstack-protector`, `AArch64.-mstack-protector-guard`, `ARM.-mstack-protector-guard`, `IBM-RS/6000-and-PowerPC.-mstack-protector-guard`, `no_stack_protector`, `RISC-V.-mstack-protector-guard`, `x86.-mstack-protector-guard`, `AArch64 Options`, `ARM Options`, and `Common Function Attributes`. The left sidebar contains a navigation menu for the GCC documentation, including links to "Using the GNU Compiler Collection 13.0.1 (experimental 20220101) documentation", "Copyright", "Introduction", "Programming Languages Supported by GCC", "Language Standards Supported by GCC", "Description", "Option Summary", "Options Controlling the Kind of Output", "Compiling C++ Programs", "Options Controlling C Dialect", "Options Controlling C++ Dialect", "Options Controlling Objective-C and Objective-C++ Dialects", "Options to Control Diagnostic Messages Formatting", "Options to Request or Suppress Warnings", "Options That Control Static Analysis", "Options for Debugging Your Program", "Options That Control Optimization", and "Program Instrumentation Options".

1.1.4 Link location

HTML links for options are misplaced by Texinfo:

The screenshot shows a web browser window with the address bar containing `gcc.gnu.org/onlinedocs/gcc/Optimize-Options.html#index-fdefer-pop`. The page content discusses the `-fdefer-pop` option. It states: "For machines that must pop arguments after a function call, always pop the arguments as soon as each function returns. At levels `-O1` and higher, `-fdefer-pop` is the default; this allows the compiler to let arguments accumulate on the stack for several function calls and pop them all at once." Below this, the `-fforward-propagate` option is described: "Perform a forward propagation pass on RTL. The pass tries to combine two instructions and checks if the result can be simplified. If loop unrolling is active, two passes are performed and the second is scheduled after loop unrolling. This option is enabled by default at optimization levels `-O1`, `-O2`, `-O3`, `-Os`". Further down, the `-ffp-contract-style` option is mentioned: "`-ffp-contract=off` disables floating-point expression contraction. `-ffp-contract=fast` enables floating-point expression contraction such as forming of fused multiply-add operations if the target has native support for them. `-ffp-contract=on` enables floating-point expression contraction if allowed by the language standard. This is currently not implemented and treated equal to `-ffp-contract=off`. The default is `-ffp-contract=fast`.

While it's precise for Sphinx (e.g. `-fdefer-pop`):

The screenshot shows a web browser window with the address bar containing `/home/marxin/Programming/tex2rst-generated/sphinx/gcc/_build/html/gcc-command-options/options-that-control-optimization.html#cmdoption-fdefer-pop`. The page content for the `-fdefer-pop` option is shown in a yellow-highlighted box: "Default setting; overrides `-fno-defer-pop`. Perform a forward propagation pass on RTL. The pass tries to combine two instructions and checks if the result can be simplified. If loop unrolling is active, two passes are performed and the second is scheduled after loop unrolling. This option is enabled by default at optimization levels `-O1`, `-O2`, `-O3`, `-Os`". The left sidebar contains a navigation menu for the GCC documentation, identical to the one in the previous screenshot.

1.1.5 Cross-manual references

Sphinx (using Intersphinx extension) provides an elegant way how to cross reference other manuals):
(e.g. :ref:`gcc:amd-gcn-options` [AMD GCN Options](#))

OpenMP Context Selectors



`vendor` is always `gnu`. References are to the GCC manual.

arch	kind	isa
<code>intel_mic</code> , <code>x86</code> , <code>x86_64</code> , <code>i386</code> , <code>i486</code> , <code>i586</code> , <code>i686</code> , <code>ia32</code>	<code>host</code>	See <code>-m...</code> flags in x86 Options (without <code>-m</code>)
<code>amdgcn</code> , <code>gcn</code>	<code>gpu</code>	See <code>-march=</code> in AMD GCN Options
<code>nvptx</code>	<code>gpu</code>	See <code>-misa=</code> in Nvidia PTX Options

1.1.6 Function documentation

Sphinx provides very rich directives when documenting a function, it's arguments, return value, etc.

```
.. c:function:: PyObject *PyType_GenericAlloc(PyTypeObject *type, Py_ssize_t nitems)

:param type: description of the first parameter.
:param nitems: description of the second parameter.
:returns: a result.
:retval NULL: under some conditions.
:retval NULL: under some other conditions as well.
```

which results in:

`PyObject *PyType_GenericAlloc(PyTypeObject *type, Py_ssize_t nitems)`

Parameters

- `type` – description of the first parameter.
- `nitems` – description of the second parameter.

Returns

a result.

Return values

- `NULL` – under some conditions.
- `NULL` – under some other conditions as well.

1.2 PDF output

1.2.1 Cross-manual references

Again, PDF version contains many references which are handy with preview feature:

Texinfo version:

166

Using the GNU Compiler Collection (GCC)

- ```
-ftree-ter
-funit-at-a-time

-02 Optimize even more. GCC performs nearly all supported optimizations that do
 not involve a space-speed tradeoff. As compared to '-0', this option increases
 both compilation time and the performance of the generated code.

 '-02' turns on all optimization flags specified by '-01'. It also turns on the
 following optimization flags:
 -falign-functions -falign-jumps
 -falign-labels -falign-loops
 -fcaller-saves
 -fcode-hoisting
 -fcrossjumping
 -fcse-follow-jumps -fcse-skip-blocks
 -fdelete-null-pointer-checks
 -fdevirtualize -fdevirtualize-speculatively
 -fexpensive-optimizations
 -ffinite-loops
 -fgcse -fgcse-lm
 -fhoist-adjacent-loads
 -finline-functions
 -finline-small-functions
 -findirect-inlining
 -fipa-bit-cp -fipa-cp -fipa-icf
 -fipa-ra -fipa-sra -fipa-vrp
 -fisolate-erroneous-paths-dereference
 -flra-remat
 -foptimize-sibling-calls
 -foptimize-strlen
 -fpartial-inlining
 -fpeephole2
 -freorder-blocks-algorithm=stc
 -freorder-blocks-and-partition -freorder-functions
 -frerun-cse-after-loop
 -fschedule-insns -fschedule-insns2
 -fsched-interblock -fsched-spec
 -fstore-merging
 -fstrict-aliasing
 -fthead-jumps
 -ftree-built-in-call-dce
 -ftree-loop-vectorize
 -ftree-pre
 -ftree-slp-vectorize
 -ftree-switch-conversion -ftree-tail-merge
 -ftree-vrp
 -fvect-cost-model=very-cheap

 Please note the warning under '-fgcse' about invoking '-02' on programs that
 use computed gotos.

-03 Optimize yet more. '-03' turns on all optimizations specified by '-02' and also
 turns on the following optimization flags:
 -fgcse-after-reload
 -fipa-cp-clone -floop-interchange
 -floop-unroll-and-jam
 -fpeel-loops
 -fpredictive-commoning
```

Sphinx version:

The screenshot shows two pages of the GCC documentation. The left page is titled 'Control Optimization' and contains sections for '-O2' and '-O3'. The right page is titled 'Experimental 20220101'.

**-O2**

- Optimize even more. GCC performs nearly all supported optimizations that do not involve a space-speed tradeoff. As compared to -O, this option increases both compilation time and the performance of the generated code.
- O2 turns on all optimization flags specified by -O1. It also turns on the following optimization flags:
  - falign-functions -falign-jumps
  - falign-labels -falign-loops
  - fcaller-saves
  - fcode-hoisting
  - fcrossjumping
  - fipa-bit-cp -fipa-cp -fipa-icf
  - fipa-ra -fipa-sra -fipa-vrp
  - fisolate-erroneous-paths-dereference
  - flra-remat
  - foptimize-sibling-calls
  - foptimize-strlen
  - fpartial-inlining
  - fpeephole2
  - freorder-blocks-algorithm=stc
  - freorder-blocks-and-partition -freorder-functions
  - frerun-cse-after-loop
  - fschedule-insns -fschedule-insns2
  - fsched-interblock -fsched-spec
  - fstore-merging
  - fstrict-aliasing
  - fthread-jumps
  - ftree-built-in-call-dce
  - ftree-loop-vectorize
  - ftree-pre
  - ftree-slp-vectorize
  - ftree-switch-conversion -ftree-tail-merge
  - ftree-vrp
  - fvect-cost-model=very-cheap
- Please note the warning under -fgcse about invoking -O2 on programs that use computed gotos.

**-O3**

- Optimize yet more. -O3 turns on all optimizations specified by -O2 and also turns on the following optimization flags:
  - fgcse-after-reload
  - fipa-cp-clone
  - floop-interchange
  - ..

## 1.2.2 Code highlighting

Again, the PDF version provides built-in syntax highlighting provided by Pygments package:

### **-Wmismatched-dealloc**

Warn for calls to deallocation functions with pointer arguments returned from allocations functions for which the former isn't a suitable deallocator. A pair of functions can be associated as matching allocators and deallocators by use of attribute `malloc`. Unless disabled by the `-fno-builtins` option the standard functions `calloc`, `malloc`, `realloc`, and `free`, as well as the corresponding forms of C++ operator `new` and operator `delete` are implicitly associated as matching allocators and deallocators. In the following example `mydealloc` is the deallocator for pointers returned from `myalloc`.

```
void mydealloc (void*);

__attribute__ ((malloc (mydealloc, 1))) void*
myalloc (size_t);

void f (void)
{
 void *p = myalloc (32);
 // ...use p...
 free (p); // warning: not a matching deallocator for myalloc
 mydealloc (p); // ok
}
```

In C++, the related option `-Wmismatched-new-delete` diagnoses mismatches involving either `operator new` or `operator delete`.

Option `-Wmismatched-dealloc` is included in `-Wall`.

### **-Wno-mismatched-dealloc**

Default setting; overrides `-Wmismatched-dealloc`.

---

CHAPTER  
TWO

---

## INTRODUCTION

I'm a demo documentation page.

**-Wno-shift-overflow, -Wshift-overflow=n, -Wshift-overflow**

---

**Note:** Default value is **-Wno-shift-overflow**, **-Wshift-overflow** is enabled by **-Wall**.

---

These options control warnings about left shift overflows.

**-Wshift-overflow=1**

This is the warning level of **-Wshift-overflow** and ...

**-Wno-shift-overflow2, -Wshift-overflow2=n** (non-default), **-Wshift-overflow2** (non-default)

These options control warnings about left shift overflows.

**-Wshift-overflow2=1**

This is the warning level of **-Wshift-overflow2** and ...

**-Wno-shift-overflow3**

Default option value for **-Wshift-overflow3**.

**-Wshift-overflow3=n, -Wshift-overflow3**

These options control warnings about left shift overflows.

**-Wshift-overflow3=1**

This is the warning level of **-Wshift-overflow3** and **-Wshift-overflow3** and **-Wshift-overflow3** and **-Wshift-overflow3** and **-Wshift-overflow3** and **-Wshift-overflow3** and ...

Enabled by **-Wall**.

**-Wauggest-attribute=[pure|const|noreturn|format|cold|malloc]**

Suggest it.

---

**Todo:** Improve wording!

---

**-Wall**

Enable it all :)

Test this: `-Wall-like-all`.

I am a **super-command**. I am **strong** and I am *emphasis*.

```
extern void *
my_memcpy (void *dest, const void *src, size_t len)
 __attribute__((nonnull (1, 2)));

void fatal () __attribute__ ((noreturn));

void
fatal /* ... */
{
 /* ... /* Print error message. */ /* ... */
 exit (1);
}
```

Show Samp with a *variable*.

Show it:

```
gcc a.c
./a.out
```

Code block:

```
gcc a.c
./a.out
```

Note1: <sup>(1)</sup> Note2: <sup>(2)</sup>

#### Diagnostic Message Formatting Options

See Options to Control Diagnostic Messages Formatting.

```
-fmessage-length=n -fdiagnostics-plain-output
-fdiagnostics-show-location=[once|every-line]
-fdiagnostics-color=[auto|never|always]
-fdiagnostics-urls=[auto|never|always]
-fdiagnostics-format=[text|json]
-fno-diagnostics-show-option -fno-diagnostics-show-caret
-fno-diagnostics-show-labels -fno-diagnostics-show-line-numbers
-fno-diagnostics-show-cwe
-fdiagnostics-minimum-margin-width=width
-fdiagnostics-parseable-fixits -fdiagnostics-generate-patch
-fdiagnostics-show-template-tree -fno-elide-type
-fdiagnostics-path-format=[none|separate-events|inline-events]
-fdiagnostics-show-path-depths
```

---

<sup>1</sup> Future versions of GCC may zero-extend, or use a target-defined `ptr_extend` pattern. Do not rely on sign extension.

<sup>2</sup> I am note 2.

```
-fno-show-column -fdiagnostics-column-unit=[display|byte]
-fdiagnostics-column-origin=origin
-fdiagnostics-escape-format=[unicode|bytes]

• -fdiagnostics-escape-format=[unicode|bytes]
• -fdiagnostics-escape-format=[unicode|bytes]
• -fdiagnostics-escape-format=[unicode|bytes]
• -fdiagnostics-escape-format=[unicode|bytes]
• -fdiagnostics-escape-format=[unicode|bytes]
• -fdiagnostics-escape-format=[unicode|bytes]
• -fdiagnostics-escape-format=[unicode|bytes]
• -fdiagnostics-escape-format=[unicode|bytes]
• -fdiagnostics-escape-format=[unicode|bytes]
```

#### Static Analyzer Options

|                                          |                                                   |                                            |
|------------------------------------------|---------------------------------------------------|--------------------------------------------|
| -fanalyzer                               | -fanalyzer-call-summaries                         | -fanalyzer-checker=name                    |
| -fno-analyzer-feasibility                | -fanalyzer-fine-grained                           | -fanalyzer-state-merge                     |
| -fanalyzer-state-purge                   | -fanalyzer-transitivity                           | -fanalyzer-verbose-edges                   |
| -fanalyzer-verbose-state-changes         | -fanalyzer-verbosity=level                        | -fdump-analyzer                            |
| -fdump-analyzer-stderr                   | -fdump-analyzer-callgraph                         | -fdump-analyzer-exploded-graph             |
| -fdump-analyzer-exploded-nodes           |                                                   | -fdump-analyzer-exploded-nodes-2           |
| -fdump-analyzer-exploded-nodes-3         |                                                   | -fdump-analyzer-exploded-paths             |
| -fdump-analyzer-feasibility              | -fdump-analyzer-json                              | -fdump-analyzer-state-purge                |
| -fdump-analyzer-supergraph               |                                                   | -Wno-analyzer-double-fclose                |
| -Wno-analyzer-double-free                |                                                   | -Wno-analyzer-exposure-through-output-file |
| -Wno-analyzer-file-leak                  |                                                   | -Wno-analyzer-free-of-non-heap             |
| -Wno-analyzer-malloc-leak                |                                                   | -Wno-analyzer-mismatching-deallocation     |
| -Wno-analyzer-null-argument              |                                                   | -Wno-analyzer-null-dereference             |
| -Wno-analyzer-possible-null-argument     |                                                   | -Wno-analyzer-possible-null-dereference    |
| -Wno-analyzer-shift-count-negative       |                                                   | -Wno-analyzer-shift-count-overflow         |
| -Wno-analyzer-stale-setjmp-buffer        |                                                   | -Wno-analyzer-tainted-allocation-size      |
| -Wno-analyzer-tainted-array-index        |                                                   | -Wno-analyzer-tainted-divisor              |
| -Wno-analyzer-tainted-offset             |                                                   | -Wno-analyzer-tainted-size                 |
| -W analyzer-too-complex                  | -Wno-analyzer-unsafe-call-within-signal-handler   |                                            |
| -Wno-analyzer-use-after-free             | -Wno-analyzer-use-of-pointer-in-stale-stack-frame |                                            |
| -Wno-analyzer-use-of-uninitialized-value |                                                   | -Wno-analyzer-write-to-const               |
| -Wno-analyzer-write-to-string-literal    |                                                   |                                            |

My name is Karlíček Koníčku and François-Xavier.



---

CHAPTER  
THREE

---

MULTIPLE

### 3.1 Arithmetic functions

```
_Decimal32 __dpd_addsd3(_Decimal32 a, _Decimal32 b)
_Decimal32 __bid_addsd3(_Decimal32 a, _Decimal32 b)
_Decimal64 __dpd_adddd3(_Decimal64 a, _Decimal64 b)
_Decimal64 __bid_adddd3(_Decimal64 a, _Decimal64 b)
_Decimal128 __dpd_addtd3(_Decimal128 a, _Decimal128 b)
_Decimal128 __bid_addtd3(_Decimal128 a, _Decimal128 b)
```

These functions return the sum of *a* and *b*.

### 3.2 Names:

- git
- SSH
- FooBar (all versions)

Install these requirements.



---

**CHAPTER  
FOUR**

---

## **TOOLS/PACKAGES NECESSARY FOR BUILDING GCC**

- autoconf version 2.69
- GNU m4 version 1.4.6 (or later)

Necessary when modifying `configure.ac`, `aclocal.m4`, etc. to regenerate `configure` and `config.in` files.

git (any version) SSH (any version) Foobar (all versions)

Necessary to access the source repository. Public releases and weekly snapshots of the development sources are also available via HTTPS.

**-mmmx**

**-msse**

**-msse2**

These switches enable the use of instructions in the MMX, SSE, SSE2, ...

Use these options:    `-Wno-shift-overflow3`    `-Wno-shift-overflow3`    `-Wno-shift-overflow3`  
`-Wno-shift-overflow3`    `-Wno-shift-overflow3`    `-Wno-shift-overflow3`    `-Wno-shift-overflow3`  
`-Wno-shift-overflow3`    `-Wno-shift-overflow3`    `-Wno-shift-overflow3`    `-Wno-shift-overflow3`  
`-Wno-shift-overflow3`    `-Wno-shift-overflow3`    `-Wno-shift-overflow3`    `-Wno-shift-overflow3`.

Code block:

```
gcc a.c
./a.out
```

And then there's a note:

---

**Note:** I am a note!

---

And then there's something else.

Listing:

**`file.cc file.cp file.cxx file.cpp file.CPP file.c++ file.C`**

C++ source code that must be preprocessed. Note that in `.cxx`, the last two letters must both be literally x. Likewise, `.C` refers to a literal capital C.

**file.mm file.M**

Objective-C++ source code that must be preprocessed.

**file.mii**

Objective-C++ source code that should not be preprocessed.

**file.hh file.H file.hp file.hxx file.hpp file.HPP file.h++ file.tcc**

C++ header file to be turned into a precompiled header or Ada spec.

**--verbose, -v**

Use verbose mode.

Use --verbose mode.

This works: --verbose and -v.

enum `gimple_code` **gimple\_code**(`gimple g`)

Return the code for statement G.

**-o** directory, **--object-directory** directory, **--object-file** file

Test it.

**-foo=BAR, -f BAR BAZ**

Using emphasising placeholders.

**-foo=BAR, -f BAR BAZ**

Normal option values.

---

**Todo:** FIX this page properly

---

---

**CHAPTER  
FIVE**

---

**FORMATTING**

just a test to reference a [PR12345](#).

---

**Todo:** FIX this page properly

---

(The *original entry* is located in /\_\_w/texi2rst/texi2rst/texi2rst-generated/sphinx/demo/demo2.rst, line 109.)

---

**Todo:** Improve wording!

---

(The *original entry* is located in /\_\_w/texi2rst/texi2rst/texi2rst-generated/sphinx/demo/demo.rst, line 43.)

GCC 13 Release Series



## CHANGES, NEW FEATURES, AND FIXES

This page is a “brief” summary of some of the huge number of improvements in GCC 13. You may also want to check out our [Porting to GCC 13](#) page and the [full GCC documentation](#).

Note: GCC 13 has not been released yet, so this document is a work-in-progress.

### 6.1 Caveats

- The support for the `cr16-elf`, `tilegx*-linux`, `tilepro*-linux`, `hppa[12]*-*-hpux10*`, `hppa[12]*-*-hpux11*` and `m32c-rtems` configurations has been removed.
- Support for emitting the STABS debugging format (including the `-gstabs` and `-gxcoff` options) has been removed. (This means the `dbx` debugger is no longer supported, either.)
- Legacy debug info compression option `-gz=zlib-gnu` was removed and the option is ignored right now.
- New debug info compression option value `-gz=zstd` has been added.

### 6.2 General Improvements

- OpenMP
  - Reverse offload is now supported and the all clauses to the `requires` directive are now accepted; however, the `requires_offload`, `unified_address` and `unified_shared_memory` clauses cause that the only available device is the initial device (the host).
  - The following OpenMP 5.1 features have been added: the `omp_all_memory` reserved locator, the `inoutset` modifier to the `depend` clause, the `nowait` clause for the `taskwait` directive and the `omp_target_is_accessible`, `omp_target_memcpy_async`, `omp_target_memcpy_rect_async` and `omp_get_mapped_ptr` API routines. Fortran now supports non-rectangular loop nests, which were added for C/C++ in GCC 11.
  - Initial support for OpenMP 5.2 features have been added: Support for `firstprivate` and `allocate` clauses on the `scope` construct and the OpenMP 5.2 syntax of the `linear` clause; the new enum/constants `omp_initial_device` and `omp_invalid_device`; and optionally omitting the map-type in `target enter/exit data`. The `enter` clause (as alias for `to`) has been added to the `declare target` directive.

- For user defined allocators requesting high bandwidth or large capacity memspaces or interleaved partitioning, the `memkind` library is used, if available at runtime.
- AddressSanitizer defaults to `detect_stack_use_after_return=1` on GNU/Linux targets. For compatibility, it can be disabled with env `ASAN_OPTIONS=detect_stack_use_after_return=0`.
- Link-time optimization improvements:
  - LTO supports the newly added jobserver of GNU make jobserver that uses named pipes (`--jobserver-style=fifo`) by default.
  - If make's jobserver is active, parallel LTO WPA streaming communicates with it and thus avoids system overcommitting.

## 6.3 New Languages and Language specific improvements

### 6.3.1 C

- Several C23 features have been implemented:
  - [N3042](#), Introduce the `nullptr` constant
  - Support for empty initializer braces
- New warnings:
  - `-Wenum-int-mismatch` warns about mismatches between an enumerated type and an integer type ([PR105131](#))

### 6.3.2 C++

- Several C++23 features have been implemented:
  - [P2324R1](#), Labels at the end of compound statements ([PR103539](#))
  - [P2255R2](#), A type trait to detect reference binding to temporary ([PR104477](#))
  - [P2327R1](#), De-deprecating volatile compound operations
  - [P2437R1](#), Support for `#warning` ([PR106646](#))
  - [P2290R3](#), Delimited escape sequences ([PR106645](#))
  - [P2071R2](#), Named universal character escapes ([PR106648](#))
  - [P2513R3](#), `char8_t` Compatibility and Portability Fix ([PR106656](#))
  - [P1169R4](#), `static operator()` ([PR106651](#))
  - [P2266R3](#), Simpler implicit move ([PR101165](#))
- New warnings:
  - `-Wself-move` warns when a value is moved to itself with `std::move` ([PR81159](#))

- The `-Wpessimizing-move` and `-Wredundant-move` warnings have been extended to warn in more contexts.

## Runtime Library (libstdc++)

- Improved experimental support for C++23, including:
  - Additions to the `<ranges>` header: `views::zip`, `views::zip_transform`, `views::adjacent`, `views::adjacent_transform`, `views::pairwise`, `views::slide`, `views::chunk`, `views::chunk_by`.
- Support for the `<experimental/scope>` header from v3 of the Library Fundamentals Technical Specification.

## 6.4 New Targets and Target Specific Improvements

### 6.4.1 AMD Radeon (GCN)

- Support for the Instinct MI200 series devices (``gtx90a <https://gcc.gnu.org/onlinedocs/gcc/AMD-GCN-Options.html>``) has been added.

### 6.4.2 arm

- The STAR-MC1 CPU is now supported through the `star-mc1` argument to the `-mcpu` and `-mtune` options.

### 6.4.3 IA-32/x86-64

- For both C and C++ the `_bf16` type is supported on x86 systems with SSE2 and above enabled.

### 6.4.4 NVPTX

- The default value for the ``-march <https://gcc.gnu.org/onlinedocs/gcc/Nvidia-PTX-Options.html>`` option can be now changed when building GCC using the `--with-arch= <https://gcc.gnu.org/install/specify.html#nvptx-x-none>`` configure option. GCC's target libraries are then build both with `sm_30` and the specified target architecture. If not specified, GCC defaults to `sm_30`.

## 6.5 Operating Systems

## 6.6 Other significant improvements

For questions related to the use of GCC, please consult these web pages and the [GCC manuals](#). If that fails, the [gcc-help@gcc.gnu.org](mailto:gcc-help@gcc.gnu.org) mailing list might help. Comments on these web pages and the development of GCC are welcome on our developer list at [gcc@gcc.gnu.org](mailto:gcc@gcc.gnu.org). All of [our lists](#) have public archives. Copyright (C) [Free Software Foundation, Inc.](#) Verbatim copying and distribution of this entire article is permitted in any medium, provided this notice is preserved.

These pages are [maintained by the GCC team](#). Last modified 2022-10-03.

## C++ STANDARDS SUPPORT IN GCC

**Warning:** The page was automatically converted by pandoc tool!

GCC supports different dialects of C++, corresponding to the multiple published ISO standards. Which standard it implements can be selected using the `-std=` command-line option.

- [\*C++98\*](#)
- [\*C++11\*](#)
- [\*C++14\*](#)
- [\*C++17\*](#)
- [\*C++20\*](#)
- [\*C++23\*](#)
- [\*Technical Specifications\*](#)

For information about the status of C++ defect reports, please see [this page](#).

For information about the status of the library implementation, please see the [Implementation Status](#) section of the Libstdc++ manual.

### 7.1 C++23 Support in GCC

GCC has experimental support for the next revision of the C++ standard, which is expected to be published in 2023.

C++23 features are available since GCC 11. To enable C++23 support, add the command-line parameter `-std=c++2b` to your `g++` command line. Or, to enable GNU extensions in addition to C++23 features, add `-std=gnu++2b`.

**Important:** Because the ISO C++23 standard is still evolving, GCC's support is **experimental**. No attempt will be made to maintain backward compatibility with implementations of C++23 features that do not reflect the final standard.

## 7.2 C++23 Language Features

| Language Feature                                                    | Proposal | Available in GCC? | SD-6 Feature Test                           |
|---------------------------------------------------------------------|----------|-------------------|---------------------------------------------|
| Literal Suffix for (signed) size_t                                  | P0330R8  | 11                | __cpp_size_t_suffix >= 202006L              |
| Make () more optional for lambdas                                   | P1102R2  | 11                |                                             |
| DR: Declarations and where to find them                             | P1787R6  | No                |                                             |
| if consteval                                                        | P1938R3  | 12                | __cpp_if_consteval >= 202106L               |
| C++ Identifier Syntax using Unicode Standard Annex 31               | P1949R7  | 12                |                                             |
| Allow Duplicate Attributes                                          | P2156R1  | 11                |                                             |
| Narrowing contextual conversions to bool                            | P1401R5  | 9                 |                                             |
| Trimming whitespaces before line splicing                           | P2223R2  | Yes               |                                             |
| Mixed string literal concatenation                                  | P2201R1  | Yes               |                                             |
| Make declaration order layout mandated                              | P1847R4  | Yes               |                                             |
| Removing Garbage Collection Support                                 | P2186R2  | 12                |                                             |
| Simpler implicit move                                               | P2266R3  | 13                | __cpp_implicit_move >= 202207L              |
| Deducing this                                                       | P0847R7  | No                | __cpp_explicit_this_parameter >= 202110L    |
|                                                                     | CWG2586  |                   |                                             |
| Change scope of lambda trailing-return-type                         | P2036R3  | No                |                                             |
|                                                                     | P2579R0  |                   |                                             |
| Multidimensional subscript operator                                 | P2128R6  | 12                | __cpp_multidimensional_subscript >= 202110L |
|                                                                     | CWG2507  | 13                |                                             |
| Non-literal variables (and labels and gotos) in constexpr functions | P2242R3  | 12                | __cpp_constexpr >= 202110L                  |
| Character encoding of diagnostic text                               | P2246R1  | No                |                                             |
| Character sets and encodings                                        | P2314R4  | No                |                                             |

continues on next page

Table 1 – continued from previous page

| Language Feature                                                                               | Proposal | Available in GCC?                     | SD-6 Feature Test                                        |
|------------------------------------------------------------------------------------------------|----------|---------------------------------------|----------------------------------------------------------|
| Consistent character literal encoding                                                          | P2316R2  | Yes                                   |                                                          |
| Add support for preprocessing directives <code>#elifdef</code> and <code>#elifndef</code>      | P2334R1  | 12                                    |                                                          |
| Extend init-statement to allow alias-declaration                                               | P2360R0  | 12                                    |                                                          |
| <code>auto(x): decay-copy</code> in the language                                               | P0849R8  | 12                                    |                                                          |
| Labels at the end of compound statements                                                       | P2324R1  | 13                                    |                                                          |
| CWG 2397: <code>auto</code> specifier for pointers and references to arrays                    | CWG2397  | 12                                    |                                                          |
| <b>CWG 2481:</b><br><b>Cv-qualification of temporary to which a reference is bound</b>         | CWG2481  | Yes                                   |                                                          |
| Attributes on lambda-expressions                                                               | P2173R1  | 9                                     |                                                          |
| A type trait to detect reference binding to temporary                                          | P2255R2  | 13                                    |                                                          |
| The Equality Operator You Are Looking For                                                      | P2468R2  | No                                    |                                                          |
| De-deprecating <code>volatile</code> compound operations                                       | P2327R1  | 13                                    |                                                          |
| Support for <code>#warning</code>                                                              | P2437R1  | Yes<br>(extension)<br>13<br>(P2437R1) |                                                          |
| <b>Remove non-encodable wide character literals and multicharacter wide character literals</b> | P2362R3  | 13                                    |                                                          |
| Delimited escape sequences                                                                     | P2290R3  | 13                                    |                                                          |
| Named universal character escapes                                                              | P2071R2  | 13                                    | <code>__cpp_named_character_escapes</code><br>>= 202207L |
| Relaxing some <code>constexpr</code> restrictions                                              | P2448R2  | No                                    | <code>__cpp_constexpr</code> >= 202207L                  |

continues on next page

Table 1 – continued from previous page

| Language Feature                                              | Proposal | Available in GCC? | SD-6 Feature Test                     |
|---------------------------------------------------------------|----------|-------------------|---------------------------------------|
| Using unknown references in constant expressions              | P2280R4  | No                |                                       |
| static operator()                                             | P1169R4  | 13                | __cpp_static_call_operator >= 202207L |
| Extended floating-point types and standard names              | P1467R9  | No                |                                       |
| Class template argument deduction from inherited constructors | P2582R1  | No                |                                       |
| Portable assumptions                                          | P1774R8  | 13                |                                       |
| Support for UTF-8 as a portable source file encoding          | P2295R6  | 13                |                                       |
| char8_t Compatibility and Portability Fix                     | P2513R3  | 13                | __cpp_char8_t >= 202207L              |
| Relax requirements on wchar_t to match existing practices     | P2460R2  | Yes               |                                       |
| Explicit lifetime management                                  | P2590R2  | No                |                                       |

## 7.3 C++20 Support in GCC

GCC has experimental support for the latest revision of the C++ standard, which was published in 2020.

C++20 features are available since GCC 8. To enable C++20 support, add the command-line parameter `-std=c++20` (use `-std=c++2a` in GCC 9 and earlier) to your `g++` command line. Or, to enable GNU extensions in addition to C++20 features, add `-std=gnu++20`.

**Important:** Because the ISO C++20 standard is very recent, GCC's support is **experimental**.

## 7.4 C++20 Language Features

The following table lists new language features that have been accepted into the C++20 working draft. The “Proposal” column provides a link to the ISO C++ committee proposal that describes the feature, while the “Available in GCC?” column indicates the first version of GCC that contains an implementation of this feature (if it has been implemented).

| Language Feature                           | Proposal | Available in GCC? | SD-6 Feature Test |
|--------------------------------------------|----------|-------------------|-------------------|
| Default member initializers for bit-fields | P0683R1  | 8                 |                   |
| Fixing const-qualified pointers to members | P0704R1  | 8                 |                   |

continues on next page

Table 2 – continued from previous page

| Language Feature                             | Proposal           | Available in GCC?                                                                                                               | SD-6 Feature Test                          |
|----------------------------------------------|--------------------|---------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------|
| Allow lambda capture [=, this]               | P0409R2            | 8                                                                                                                               |                                            |
| __VA_OPT__ for preprocessor comma elision    | P0306R4<br>P1042R1 | 8 (partial,<br>no<br>#__VA_OPT__<br>support)<br>10 (par-<br>tial, no<br>place-<br>marker<br>token<br>handling<br>changes)<br>12 |                                            |
| Designated initializers                      | P0329R4            | 8                                                                                                                               | __cpp_designated_initializers<br>=> 201707 |
| Familiar template syntax for generic lambdas | P0428R2            | 8                                                                                                                               | __cpp_generic_lambdas<br>=> 201707         |
| List deduction of vector                     | P0702R1            | 8                                                                                                                               |                                            |
| Concepts                                     | P0734R0            | 10                                                                                                                              | __cpp_concepts >= 201907                   |
|                                              | P0857R0            |                                                                                                                                 |                                            |
|                                              | P1084R2            |                                                                                                                                 |                                            |
|                                              | P1141R2            |                                                                                                                                 |                                            |
|                                              | P0848R3            |                                                                                                                                 | __cpp_concepts >= 202002                   |
|                                              | P1616R1            |                                                                                                                                 | __cpp_concepts >= 201907                   |
|                                              | P1452R2            |                                                                                                                                 |                                            |
|                                              | P1972R0            |                                                                                                                                 |                                            |
|                                              | P1980R0            |                                                                                                                                 |                                            |
|                                              | P2092R0            |                                                                                                                                 |                                            |
|                                              | P2103R0            |                                                                                                                                 |                                            |
|                                              | P2113R0            | 10.2 (no<br>reversed<br>operator<br>han-<br>dling)                                                                              |                                            |
| Range-based for statements with initializer  | P0614R1            | 9                                                                                                                               |                                            |

continues on next page

Table 2 – continued from previous page

| Language Feature                                                          | Proposal | Available in GCC?              | SD-6 Feature Test                                             |
|---------------------------------------------------------------------------|----------|--------------------------------|---------------------------------------------------------------|
| Simplifying implicit lambda capture                                       | P0588R1  | 8                              |                                                               |
| ADL and function templates that are not visible                           | P0846R0  | 9                              |                                                               |
| <code>const</code> mismatch with defaulted copy constructor               | P0641R2  | 9                              |                                                               |
| Less eager instantiation of <code>constexpr</code> functions              | P0859R0  | 5.2 (mostly)<br>9<br>(P0859R0) | <code>__cpp_constexpr_in_decltype</code><br>$\geq 201711$     |
| Consistent comparison ( <code>operator&lt;=</code> )                      | P0515R3  | 10                             | <code>__cpp_impl_three_way_comparison</code><br>$\geq 201711$ |
|                                                                           | P0905R1  |                                |                                                               |
|                                                                           | P1120R0  |                                |                                                               |
|                                                                           | P1185R2  |                                |                                                               |
|                                                                           | P1186R3  | 11                             |                                                               |
|                                                                           | P1630R1  | 10                             |                                                               |
|                                                                           | P1946R0  |                                |                                                               |
|                                                                           | P1959R0  |                                |                                                               |
|                                                                           | P2002R1  | 10.2                           |                                                               |
|                                                                           | P2085R0  |                                |                                                               |
| Access checking on specializations                                        | P0692R1  | Yes                            |                                                               |
| Default constructible and assignable stateless lambdas                    | P0624R2  | 9                              |                                                               |
| Lambdas in unevaluated contexts                                           | P0315R4  | 9                              |                                                               |
| Language support for empty objects                                        | P0840R2  | 9                              |                                                               |
| Relaxing the range-for loop customization point finding rules             | P0962R1  | 8                              |                                                               |
| Allow structured bindings to accessible members                           | P0969R0  | 8                              |                                                               |
| <b>Relaxing the structured bindings customization point finding rules</b> | P0961R1  | 8                              |                                                               |
| Down with typename!                                                       | P0634R3  | 9                              |                                                               |
| Allow pack expansion in lambda init-capture                               | P0780R2  | 9                              | <code>__cpp_init_captures</code><br>$\geq 201803$             |

continues on next page

Table 2 – continued from previous page

| Language Feature                                        | Proposal | Available in GCC?                                                               | SD-6 Feature Test                             |
|---------------------------------------------------------|----------|---------------------------------------------------------------------------------|-----------------------------------------------|
|                                                         | P2095R0  | 10                                                                              |                                               |
| Proposed wording for likely and unlikely attributes     | P0479R5  | 9                                                                               |                                               |
| Deprecate implicit capture of this via [=]              | P0806R2  | 9                                                                               |                                               |
| Class Types in Non-Type Template Parameters             | P0732R2  | 9                                                                               | __cpp_nontype_template >= 201806 parameter_cl |
| Inconsistencies with non-type template parameters       | P1907R1  | <b>10</b> (no floating point, union, or sub-object template args) 11 (complete) | __cpp_nontype_template args >= 201911         |
| Atomic Compare-and-Exchange with Padding Bits           | P0528R3  | <b>11</b> (compiler side only)<br>13 (full)                                     |                                               |
| Efficient sized delete for variable sized classes       | P0722R3  | 9                                                                               | __cpp_impl_destroying_delete >= 201806        |
| Allowing Virtual Function Calls in Constant Expressions | P1064R0  | 9                                                                               |                                               |
| Prohibit aggregates with user-declared constructors     | P1008R1  | 9                                                                               |                                               |
| explicit(bool)                                          | P0892R2  | 9                                                                               | __cpp_conditional_explicit >= 201806          |
| Signed integers are two's complement                    | P1236R1  | 9                                                                               |                                               |
| char8_t                                                 | P0482R6  | 9                                                                               | __cpp_char8_t >= 201811                       |
| Immediate functions (consteval)                         | P1073R3  | <b>10</b> (no consteval virtual)<br>11 (full)                                   | __cpp_consteval >= 201811                     |
|                                                         | P1937R2  | 10                                                                              |                                               |
| std::is_constant_evaluated                              | P0595R2  | 9                                                                               |                                               |
| Nested inline namespaces                                | P1094R2  | 9                                                                               |                                               |

continues on next page

Table 2 – continued from previous page

| Language Feature                      | Proposal | Available in GCC?                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | SD-6 Feature Test                                             |
|---------------------------------------|----------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------|
| Relaxations of constexpr restrictions | P1002R1  | 9                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |                                                               |
|                                       | P1327R1  | 10                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |                                                               |
|                                       | P1330R0  | 9                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | <code>__cpp_constexpr &gt;= 202002</code>                     |
|                                       | P1331R2  | 10                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | <code>__cpp_constexpr &gt;= 201907</code>                     |
|                                       | P1668R1  | 10                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |                                                               |
|                                       | P0784R7  | 10                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | <code>__cpp_constexpr_dynamic_alloc &gt;= 201907</code>       |
| Feature test macros                   | P0941R2  | <p>4.9 (<code>__cpp_macros</code>)<br/> <code>5</code><br/> <code>(__has_cpp</code></p>                                                                                                                                                                                                                                                                                                                                                                                                                                                              |                                                               |
| Modules                               | P1103R3  | <p>11 (requires <code>-fmodules-ts</code>)<br/>         (No<br/>         Private<br/>         Module<br/>         Frag-<br/>         ment,<br/>         Parser-<br/>         level<br/>         Global<br/>         Module<br/>         Entity<br/>         Merging,<br/>         Global<br/>         Module<br/>         Implica-<br/>         tions of<br/>         extern<br/>         "C/<br/>         C++", or<br/>         Partition-<br/>         specific<br/>         Defi-<br/>         nition<br/>         Visibil-<br/>         ity)</p> | <code>__cpp_modules &gt;= 201810L</code><br>(Date of p1103r3) |

continues on next page

Table 2 – continued from previous page

| Language Feature                                                                               | Proposal                      | Available in GCC?                       | SD-6 Feature Test                       |
|------------------------------------------------------------------------------------------------|-------------------------------|-----------------------------------------|-----------------------------------------|
|                                                                                                | P1766R1                       | No                                      |                                         |
|                                                                                                | P1811R0                       | 11                                      |                                         |
|                                                                                                | P1703R1 (superceded by p1857) | 11                                      |                                         |
|                                                                                                | P1874R1                       | 11                                      |                                         |
|                                                                                                | P1979R0                       | 11                                      |                                         |
|                                                                                                | P1779R3                       | 11                                      |                                         |
|                                                                                                | P1857R3                       | 11                                      |                                         |
|                                                                                                | P2115R0                       | 11                                      |                                         |
|                                                                                                | P1815R2                       | No                                      |                                         |
| Coroutines                                                                                     | P0912R5 as applied to n4861   | 10 (requires -fcoroutines)<br>11        | __cpp_impl_coroutine<br>>= 201902       |
| Parenthesized initialization of aggregates                                                     | P0960R3                       | 10                                      | __cpp_aggregate_paren_init<br>>= 201902 |
|                                                                                                | P1975R0                       | 11                                      |                                         |
| DR: array size deduction in <i>new-expression</i>                                              | P1009R2                       | 11                                      |                                         |
| <b>DR: Converting from <code>T*</code> to <code>bool</code> should be considered narrowing</b> | P1957R2                       | 10 (C++20 mode only),<br>11 (all modes) |                                         |
| Stronger Unicode requirements                                                                  | P1041R4<br>P1139R2            | 10                                      |                                         |
| Structured binding extensions                                                                  | P1091R3<br>P1381R1            | 10<br>8                                 |                                         |
| Deprecate <code>a[b,c]</code>                                                                  | P1161R3                       | 10                                      |                                         |
| Deprecating some uses of <code>volatile</code>                                                 | P1152R4                       | 10                                      |                                         |
| <code>[[nodiscard("with reason")]]</code>                                                      | P1301R4                       | 10                                      |                                         |
| <code>using enum</code>                                                                        | P1099R5                       | 11                                      |                                         |
| Class template argument deduction for aggregates                                               | P1816R0                       | 10                                      | __cpp_deduction_guides<br>>= 201907L    |
|                                                                                                | P2082R1                       | 11                                      |                                         |

continues on next page

Table 2 – continued from previous page

| Language Feature                                            | Proposal | Available in GCC? | SD-6 Feature Test                         |
|-------------------------------------------------------------|----------|-------------------|-------------------------------------------|
| Class template argument deduction for alias templates       | P1814R0  | 10                |                                           |
| Permit conversions to arrays of unknown bound               | P0388R4  | 10                |                                           |
| <code>constinit</code>                                      | P1143R2  | 10                | <code>__cpp_constinit &gt;= 201907</code> |
| Layout-compatibility and Pointer-interconvertibility Traits | P0466R5  | 12                | (in library)                              |
| DR: Checking for abstract class types                       | P0929R2  | 11                |                                           |
| DR: More implicit moves (merge P0527R1 and P1155R3)         | P1825R0  | 11 (C++20 mode)   |                                           |
| DR: Pseudo-destructors end object lifetimes                 | P0593R6  | 11                |                                           |

## 7.5 C++17 Support in GCC

GCC has almost full support for the previous revision of the C++ standard, which was published in 2017. Some library features are missing or incomplete, as described in [the library documentation](#).

C++17 features are available since GCC 5. This mode is the default in GCC 11; it can be explicitly selected with the `-std=c++17` command-line flag, or `-std=gnu++17` to enable GNU extensions as well.

## 7.6 C++17 Language Features

The following table lists new language features that have been accepted into the C++17 working draft. The “Proposal” column provides a link to the ISO C++ committee proposal that describes the feature, while the “Available in GCC?” column indicates the first version of GCC that contains an implementation of this feature (if it has been implemented).

| Language Feature                   | Proposal | Available in GCC? | SD-6 Feature Test                                  |
|------------------------------------|----------|-------------------|----------------------------------------------------|
| Removing trigraphs                 | N4086    | 5                 |                                                    |
| <code>u8</code> character literals | N4267    | 6                 | <code>__cpp_unicode_characters &gt;= 201411</code> |
| Folding expressions                | N4295    | 6                 | <code>__cpp_fold_expressions &gt;= 201411</code>   |

continues on next page

Table 3 – continued from previous page

| Language Feature                                              | Proposal | Available in GCC?                           | SD-6 Feature Test                                                                   |
|---------------------------------------------------------------|----------|---------------------------------------------|-------------------------------------------------------------------------------------|
| Attributes for namespaces and enumerators                     | N4266    | 4.9 (namespaces)<br>6 (enumerators)         | __cpp_namespace_attributes<br>>= 201411<br>__cpp_enumerator_attributes<br>>= 201411 |
| Nested namespace definitions                                  | N4230    | 6                                           | __cpp_nested_namespace<br>>= 201411                                                 |
| Allow constant evaluation for all non-type template arguments | N4268    | 6                                           | __cpp_nontype_template_args<br>>= 201411                                            |
| Extending <code>static_assert</code>                          | N3928    | 6                                           | __cpp_static_assert >= 201411                                                       |
| New Rules for auto deduction from braced-init-list            | N3922    | 5                                           |                                                                                     |
| Allow typename in a template template parameter               | N4051    | 5                                           |                                                                                     |
| <code>[[fallthrough]]</code> attribute                        | P0188R1  | 7                                           | __has_cpp_attribute(fallthrough)                                                    |
| <code>[[nodiscard]]</code> attribute                          | P0189R1  | 4.8<br>([[gnu::warning]])<br>7<br>(P0189R1) | __has_cpp_attribute(nodiscard)                                                      |
| <code>[[maybe_unused]]</code> attribute                       | P0212R1  | 4.8<br>([[gnu::unused]])<br>7<br>(P0212R1)  | __has_cpp_attribute(maybe_unused)                                                   |
| Extension to aggregate initialization                         | P0017R1  | 7                                           | __cpp_aggregate_bases<br>>= 201603                                                  |
| Wording for <code>constexpr</code> lambda                     | P0170R1  | 7                                           | __cpp_constexpr >= 201603                                                           |
| Unary Folds and Empty Parameter Packs                         | P0036R0  | 6                                           | __cpp_fold_expressions<br>>= 201603                                                 |
| Generalizing the Range-Based For Loop                         | P0184R0  | 6                                           | __cpp_range_based_for<br>>= 201603                                                  |
| Lambda capture of <code>*this</code> by Value                 | P0018R3  | 7                                           | __cpp_capture_star_this<br>>= 201603                                                |
| Construction Rules for <code>enum class</code> variables      | P0138R2  | 7                                           |                                                                                     |

continues on next page

Table 3 – continued from previous page

| Language Feature                                          | Proposal           | Available in GCC? | SD-6 Feature Test                                                           |
|-----------------------------------------------------------|--------------------|-------------------|-----------------------------------------------------------------------------|
| Hexadecimal floating literals for C++                     | P0245R1            | 3.0               | __cpp_hex_float >= 201603                                                   |
| Dynamic memory allocation for over-aligned data           | P0035R4            | 7                 | __cpp_aligned_new >= 201606                                                 |
| Guaranteed copy elision                                   | P0135R1            | 7                 | __cpp_guaranteed_copy_elision >= 201606                                     |
| Refining Expression Evaluation Order for Idiomatic C++    | P0145R3            | 7                 |                                                                             |
| constexpr if                                              | P0292R2            | 7                 | __cpp_if_constexpr >= 201606                                                |
| Selection statements with initializer                     | P0305R1            | 7                 |                                                                             |
| Template argument deduction for class templates           | P0091R3<br>P0512R0 | 7 8               | __cpp_deduction_guides >= 201606<br>__cpp_deduction_guides >= 201611        |
| Declaring non-type template parameters with auto          | P0127R2            | 7                 | __cpp_template_auto >= 201606<br>__cpp_nontype_template_parameter >= 201606 |
| Using attribute namespaces without repetition             | P0028R4            | 7                 |                                                                             |
| Ignoring unsupported non-standard attributes              | P0283R2            | Yes               |                                                                             |
| Structured bindings                                       | P0217R3            | 7                 | __cpp_structured_bindings >= 201606                                         |
| Remove Deprecated Use of the register Keyword             | P0001R1            | 7                 |                                                                             |
| Remove Deprecated operator++(bool)                        | P0002R1            | 7                 |                                                                             |
| Make exception specifications be part of the type system  | P0012R1            | 7                 | __cpp_noexcept_function_type >= 201510                                      |
| __has_include for C++17                                   | P0061R1            | 5                 |                                                                             |
| Rewording inheriting constructors (core issue 1941 et al) | P0136R1            | 7                 | __cpp_inheriting_constructors >= 201511                                     |
| Inline variables                                          | P0386R2            | 7                 | __cpp_inline_variables >= 201606                                            |
| DR 150, Matching of template template arguments           | P0522R0            | 7                 | __cpp_template_template_args >= 201611                                      |

continues on next page

Table 3 – continued from previous page

| Language Feature                             | Proposal | Available in GCC? | SD-6 Feature Test              |
|----------------------------------------------|----------|-------------------|--------------------------------|
| Removing dynamic exception specifications    | P0003R5  | 7                 |                                |
| Pack expansions in <i>using-declarations</i> | P0195R2  | 7                 | __cpp_variadic_using >= 201611 |
| A byte type definition                       | P0298R0  | 7                 |                                |

## 7.7 Technical Specifications

GCC also implements experimental support for some language Technical Specifications published by the C++ committee.

**Important:** Because these Technical Specifications are still evolving toward future inclusion in a C++ standard, GCC's support is **experimental**. No attempt will be made to maintain backward compatibility with implementations of features that do not reflect the final standard.

| Technical Specification | Document | Available in GCC?    | Compiler Option | SD-6 Feature Test                    |
|-------------------------|----------|----------------------|-----------------|--------------------------------------|
| Concepts                | N4377    | 6                    | -fconcepts      | __cpp_concepts >= 201507             |
| Transactional Memory    | N4514    | 6 (no atomic_cancel) | -fgnu-tm        | __cpp_transactional_memory >= 201505 |
| Coroutines              | N4649    | 10                   | -fcoroutines    | __cpp_impl_coroutine >= 201902L      |
| Modules                 | N4720    | 11                   | -fmodules-ts    | __cpp_modules >= 201810L             |

## 7.8 C++14 Support in GCC

GCC has full support for the of the 2014 C++ standard.

This mode is the default in GCC 6.1 up until GCC 10 (including); it can be explicitly selected with the `-std=c++14` command-line flag, or `-std=gnu++14` to enable GNU extensions as well.

## 7.9 C++14 Language Features

The following table lists new language features that are part of the C++14 standard. The “Proposal” column provides a link to the ISO C++ committee proposal that describes the feature, while the “Available in GCC?” column indicates the first version of GCC that contains an implementation of this feature.

| Language Feature                              | Proposal | Available in<br>GCC?            | SD-6 Feature Test                            |
|-----------------------------------------------|----------|---------------------------------|----------------------------------------------|
| Tweak to certain C++ contextual conversions   | N3323    | 4.9                             |                                              |
| Binary literals                               | N3472    | 4.3 (GNU)<br>4.9<br>(N3472)     | __cpp_binary_literals<br>>= 201304           |
| Return type deduction for normal functions    | N3638    | 4.8 (N3386)<br>4.9<br>(N3638)   | __cpp_decltype_auto<br>>= 201304             |
| Generalized lambda capture (init-capture)     | N3648    | 4.5 (partial)<br>4.9<br>(N3648) | __cpp_init_captures<br>>= 201304             |
| Generic (polymorphic) lambda expressions      | N3649    | 4.9                             | __cpp_generic_lambdas<br>>= 201304           |
| Variable templates                            | N3651    | 5                               | __cpp_variable_templates<br>>= 201304        |
| Relaxing requirements on const-expr functions | N3652    | 5                               | __cpp_constexpr<br>>= 201304                 |
| Member initializers and aggregates            | N3653    | 5                               | __cpp_aggregate_nsdmi<br>>= 201304           |
| Clarifying memory allocation                  | N3664    | N/A                             |                                              |
| Sized deallocation                            | N3778    | 5                               | __cpp_sized_deallocation<br>>= 201309        |
| [[deprecated]] attribute                      | N3760    | 4.9 (N3797)                     | __has_cpp_attribute(deprecated)<br>>= 201309 |
| Single-quotation-mark as a digit separator    | N3781    | 4.9 (N3797)                     | __cpp_digit_separator<br>>= 201309           |

This feature was briefly part of the C++14 working paper, but was not part of the published standard; as a result, it has been removed from the compiler.

| Language Feature                                                                           | Proposal | Available in<br>GCC?                                   | SD-6 Feature Test                 |
|--------------------------------------------------------------------------------------------|----------|--------------------------------------------------------|-----------------------------------|
| <b>Runtime-sized arrays with automatic storage duration</b><br>(Removed from the standard) | N3639    | ?.? (GNU<br>VLAs)<br>4.9<br>(N3639)<br>5 (GNU<br>VLAs) | __cpp_runtime_arrays<br>>= 198712 |

## 7.10 C++11 Support in GCC

GCC 4.8.1 was the first feature-complete implementation of the 2011 C++ standard, previously known as C++0x.

This mode can be selected with the `-std=c++11` command-line flag, or `-std=gnu++11` to enable GNU extensions as well.

For information about C++11 support in a specific version of GCC, please see:

- [GCC 4.3 C++0x Status](#)
- [GCC 4.4 C++0x Status](#)
- [GCC 4.5 C++0x Status](#)
- [GCC 4.6 C++0x Status](#)
- [GCC 4.7 C++11 Status](#)
- [GCC 4.8 C++11 Status](#)

| Language Feature                                          | Proposal | Available in<br>GCC? | SD-6 Feature Test                                  |
|-----------------------------------------------------------|----------|----------------------|----------------------------------------------------|
| Rvalue references                                         | N2118    | GCC 4.3              | <code>__cpp_rvalue_references &gt;= 200610</code>  |
| Rvalue references for <code>*this</code>                  | N2439    | GCC 4.8.1            | <code>__cpp_ref_qualifiers &gt;= 200710</code>     |
| Initialization of class objects by rvalues                | N1610    | Yes                  |                                                    |
| Non-static data member initializers                       | N2756    | GCC 4.7              | <code>__cpp_nsmdi &gt;= 200809</code>              |
| Variadic templates                                        | N2242    | GCC 4.3              | <code>__cpp_variadic_templates &gt;= 200704</code> |
| Extending variadic template template parameters           | N2555    | GCC 4.4              |                                                    |
| Initializer lists                                         | N2672    | GCC 4.4              | <code>__cpp_initializer_lists &gt;= 200806</code>  |
| Static assertions                                         | N1720    | GCC 4.3              | <code>__cpp_static_assert &gt;= 200410</code>      |
| <code>auto</code> -typed variables                        | N1984    | GCC 4.4              |                                                    |
| Multi-declarator <code>auto</code>                        | N1737    | GCC 4.4              |                                                    |
| Removal of <code>auto</code> as a storage-class specifier | N2546    | GCC 4.4              |                                                    |
| New function declarator syntax                            | N2541    | GCC 4.4              |                                                    |
| New wording for C++0x lambdas                             | N2927    | GCC 4.5              | <code>__cpp_lambdas &gt;= 200907</code>            |

continues on next page

Table 4 – continued from previous page

| Language Feature                                  | Proposal | Available in GCC? | SD-6 Feature Test                                                                                                                                                                      |
|---------------------------------------------------|----------|-------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Declared type of an expression                    | N2343    | GCC 4.3           | <code>__cpp_decltype &gt;= 200707</code>                                                                                                                                               |
| decltype and call expressions                     | N3276    | GCC 4.8.1         |                                                                                                                                                                                        |
| Right angle brackets                              | N1757    | GCC 4.3           |                                                                                                                                                                                        |
| Default template arguments for function templates | DR226    | GCC 4.3           |                                                                                                                                                                                        |
| Solving the SFINAE problem for expressions        | DR339    | GCC 4.4           |                                                                                                                                                                                        |
| Template aliases                                  | N2258    | GCC 4.7           | <code>__cpp_alias_templates &gt;= 200704</code>                                                                                                                                        |
| Extern templates                                  | N1987    | Yes               |                                                                                                                                                                                        |
| Null pointer constant                             | N2431    | GCC 4.6           |                                                                                                                                                                                        |
| Strongly-typed enums                              | N2347    | GCC 4.4           |                                                                                                                                                                                        |
| Forward declarations for enums                    | N2764    | GCC 4.6           |                                                                                                                                                                                        |
| Generalized attributes                            | N2761    | GCC 4.8           | <code>__cpp_attributes &gt;= 200809;</code><br><code>__has_cpp_attribute(noreturn) &gt;= 200809;</code><br><code>__has_cpp_attribute(carries_dependency) == 0</code> (not implemented) |
| Generalized constant expressions                  | N2235    | GCC 4.6           | <code>__cpp_constexpr &gt;= 200704</code>                                                                                                                                              |
| Alignment support                                 | N2341    | GCC 4.8           |                                                                                                                                                                                        |
| Delegating constructors                           | N1986    | GCC 4.7           | <code>__cpp_delegating_constructors &gt;= 200604</code>                                                                                                                                |
| Inheriting constructors                           | N2540    | GCC 4.8           | <code>__cpp_inheriting_constructors &gt;= 200802</code>                                                                                                                                |
| Explicit conversion operators                     | N2437    | GCC 4.5           |                                                                                                                                                                                        |
| New character types                               | N2249    | GCC 4.4           | <code>__cpp_unicode_characters &gt;= 200704</code>                                                                                                                                     |
| Unicode string literals                           | N2442    | GCC 4.5           | <code>__cpp_unicode_literals &gt;= 200710</code>                                                                                                                                       |
| Raw string literals                               | N2442    | GCC 4.5           | <code>__cpp_raw_strings &gt;= 200710</code>                                                                                                                                            |
| Universal character name literals                 | N2170    | GCC 4.5           |                                                                                                                                                                                        |
| User-defined literals                             | N2765    | GCC 4.7           | <code>__cpp_user_defined_literals &gt;= 200809</code>                                                                                                                                  |
| Standard Layout Types                             | N2342    | GCC 4.5           |                                                                                                                                                                                        |
| Defaulted and deleted functions                   | N2346    | GCC 4.4           |                                                                                                                                                                                        |

continues on next page

Table 4 – continued from previous page

| Language Feature                                                                    | Proposal                | Available in<br>GCC?                  | SD-6 Feature Test                                                      |
|-------------------------------------------------------------------------------------|-------------------------|---------------------------------------|------------------------------------------------------------------------|
| Extended friend declarations                                                        | N1791                   | GCC 4.7                               |                                                                        |
| Extending <code>sizeof</code>                                                       | N2253                   | GCC 4.4                               |                                                                        |
| Inline namespaces                                                                   | N2535                   | GCC 4.4                               |                                                                        |
| Unrestricted unions                                                                 | N2544                   | GCC 4.6                               |                                                                        |
| Local and unnamed types as template arguments                                       | N2657                   | GCC 4.5                               |                                                                        |
| Range-based for                                                                     | N2930                   | GCC 4.6                               | <code>__cpp_range_based_for</code><br><code>&gt;= 200907</code>        |
| Explicit virtual overrides                                                          | N2928<br>N3206<br>N3272 | GCC 4.7                               |                                                                        |
| <b>Minimal support for garbage collection and reachability-based leak detection</b> | N2670                   | No                                    |                                                                        |
| Allowing move constructors to throw [noexcept]                                      | N3050                   | GCC 4.6                               |                                                                        |
| Defining move special member functions                                              | N3053                   | GCC 4.6                               |                                                                        |
| Concurrency                                                                         |                         |                                       |                                                                        |
| Sequence points                                                                     | N2239                   | Yes                                   |                                                                        |
| Atomic operations                                                                   | N2427                   | GCC 4.4                               |                                                                        |
| Strong Compare and Exchange                                                         | N2748                   | GCC 4.5                               |                                                                        |
| Bidirectional Fences                                                                | N2752                   | GCC 4.8                               |                                                                        |
| Memory model                                                                        | N2429                   | GCC 4.8                               |                                                                        |
| Data-dependency ordering: atomics and memory model                                  | N2664                   | <b>GCC 4.4</b><br>(mem-<br>ory_order) |                                                                        |
| Propagating exceptions                                                              | N2179                   | GCC 4.4                               |                                                                        |
| Abandoning a process and at_quick_exit                                              | N2440                   | GCC 4.8                               |                                                                        |
| Allow atomics use in signal handlers                                                | N2547                   | Yes                                   |                                                                        |
| Thread-local storage                                                                | N2659                   | GCC 4.8                               |                                                                        |
| Dynamic initialization and destruction with concurrency                             | N2660                   | GCC 4.3                               | <code>__cpp_threadsafe_static_init</code><br><code>&gt;= 200806</code> |
| C99 Features in C++11                                                               |                         |                                       |                                                                        |

continues on next page

Table 4 – continued from previous page

| Language Feature                          | Proposal | Available in<br>GCC? | SD-6 Feature Test |
|-------------------------------------------|----------|----------------------|-------------------|
| <code>_func_</code> predefined identifier | N2340    | GCC 4.3              |                   |
| C99 preprocessor                          | N1653    | GCC 4.3              |                   |
| <code>long long</code>                    | N1811    | GCC 4.3              |                   |
| Extended integral types                   | N1988    | Yes                  |                   |

## 7.11 C++98 Support in GCC

GCC has full support for the 1998 C++ standard as modified by the 2003 technical corrigendum and some later defect reports, excluding the `export` feature which was later removed from the language.

This mode is the default in GCC versions prior to 6.1; it can be explicitly selected with the `-std=c++98` command-line flag, or `-std=gnu++98` to enable GNU extensions as well.

For questions related to the use of GCC, please consult these web pages and the [GCC manuals](#). If that fails, the [gcc-help@gcc.gnu.org](mailto:gcc-help@gcc.gnu.org) mailing list might help. Comments on these web pages and the development of GCC are welcome on our developer list at [gcc@gcc.gnu.org](mailto:gcc@gcc.gnu.org). All of [our lists](#) have public archives. Copyright (C) [Free Software Foundation, Inc.](#) Verbatim copying and distribution of this entire article is permitted in any medium, provided this notice is preserved.

These pages are [maintained by the GCC team](#). Last modified 2022-10-21.

genindex

# INDEX

## Symbols

`_bid_adddd3` (*C++ function*), 13  
`_bid_addsd3` (*C++ function*), 13  
`_bid_addtd3` (*C++ function*), 13  
`_dpd_adddd3` (*C++ function*), 13  
`_dpd_addsd3` (*C++ function*), 13  
`_dpd_addtd3` (*C++ function*), 13  
`-Wall`  
    command line option, 9  
`-Waugment-attribute`  
    command line option, 9  
`-Wno-shift-overflow`  
    command line option, 9  
`-Wno-shift-overflow2`  
    command line option, 9  
`-Wno-shift-overflow3`  
    command line option, 9  
`-Wshift-overflow`  
    command line option, 9  
`-Wshift-overflow2`  
    command line option, 9  
`-Wshift-overflow3`  
    command line option, 9  
`--object-directory`  
    command line option, 16  
`--object-file`  
    command line option, 16  
`--verbose`  
    make command line option, 16  
`-f`  
    command line option, 16  
`-foo`  
    command line option, 16  
`-mmmx`  
    command line option, 15  
`-msse`  
    command line option, 15

`-msse2`  
    command line option, 15  
`-o`  
    command line option, 16  
`-v`  
    make command line option, 16

## C

command line option  
    `-Wall`, 9  
    `-Waugment-attribute`, 9  
    `-Wno-shift-overflow`, 9  
    `-Wno-shift-overflow2`, 9  
    `-Wno-shift-overflow3`, 9  
    `-Wshift-overflow`, 9  
    `-Wshift-overflow2`, 9  
    `-Wshift-overflow3`, 9  
    `--object-directory`, 16  
    `--object-file`, 16  
    `-f`, 16  
    `-foo`, 16  
    `-mmmx`, 15  
    `-msse`, 15  
    `-msse2`, 15  
    `-o`, 16

## G

`gimple_code` (*C++ function*), 16

## M

make command line option  
    `--verbose`, 16  
    `-v`, 16

## P

`PyType_GenericAlloc` (*C function*), 5