C Standard Library

Last updated: 08/15/2022

Contents

- <assert.h> Diagnostics
- <complex.h> Complex Type Support
- <ctype.h> Character Classification Utilities
- <errno.h> Error Codes
- <fenv.h> Floating-point Environment
- <float.h> Implementation-defined Limits
- <inttypes.h> Integer Conversions
- <iso646.h> Alternative Operator Spellings
- <limits.h> Implementation-defined Limits
- <locale.h> Localization Utilities
- <math.h> Math
- <setjmp.h> Non-local Jumps
- <signal.h> Signal Handling
- <stdarg.h> Variadic Arguments
- <stdbool.h> Boolean Type
- <stdlib.h> Utilities
- <tgmath.h> Type-generic Math
- <time.h> Date and Time Utilities
- <uchar.h> Unicode Utilities

<assert.h>

Defines one or more assertion macros. Expression `assert(e)` is defined when `<assert.h>` is included (if `NDEBUG` preprocessor macro is not defined) and evaluates to `void` if `e` compares equal to zero (false), otherwise calls implementation-defined function `_Assert(e, "e", "filename", line)`, where `filename` is a string literal and `line` is a decimal integer.

Contents

<assert.h>

```
static_assert(e, s)
_Static_assert(e, s)
assert(e)
NDEBUG
_Assert(e, s, f, l)
```

<cassert>

```
static_assert(e, s)
assert(e)
NDEBUG
_Assert(e, s, f, l)
```

<cassert.h>

```
static_assert(e, s)
assert(e)
NDEBUG
_Assert(e, s, f, l)
```

<complex.h>

Defines macros and functions for manipulating complex numbers.

Contents

```
complex
_Complex
_Complex_I
I
imaginary
_Imaginary
_Imaginary_I
CMPLXF(x, y)
CMPLXL(x, y)
CMPLX(x, y)
cacos(z), cacosf(z), cacosl(z)
casin(z), casinf(z), casinl(z)
catan(z), catanf(z), catanl(z)
ccos(z), ccosf(z), ccosl(z)
csin(z), csinf(z), csinl(z)
ctan(z), ctanf(z), ctanl(z)
cacosh(z), cacoshf(z), cacoshl(z)
casinh(z), casinhf(z), casinhl(z)
catanh(z), catanhf(z), catanhl(z)
ccosh(z), ccoshf(z), ccoshl(z)
csinh(z), csinhf(z), csinhl(z)
ctanh(z), ctanhf(z), ctanhl(z)
cexp(z), cexpf(z), cexpl(z)
clog(z), clogf(z), clogl(z)
cabs(z), cabsf(z), cabsl(z)
cpow(z, w), cpowf(z, w), cpowl(z, w)
csqrt(z), csqrtf(z), csqrtl(z)
carg(z), cargf(z), cargl(z)
cimag(z), cimagf(z), cimagl(z)
conj(z), conjf(z), conjl(z)
cproj(z), cprojf(z), cprojl(z)
```

<ctype.h>

Defines functions for character classification and case conversion.

Contents

```
isalnum(c)
isalpha(c)
isblank(c)
iscntrl(c)
isdigit(c)
isgraph(c)
islower(c)
isprint(c)
ispunct(c)
isspace(c)
isupper(c)
isxdigit(c)
tolower(c)
toupper(c)
```

<errno.h>

Defines macros for error codes.

Contents

```
errno
EDOM
EILSEQ
ERANGE
```

<fenv.h>

Defines macros and functions for floating-point environment.

Contents

```
fexcept_t
fenv_t
FE_DIVBYZERO
FE_INVALID
FE_OVERFLOW
FE_UNDERFLOW
FE_INEXACT
FE_ALL_EXCEPT
FE_DOWNWARD
FE_TONEAREST
FE_TOWARDZERO
FE_UPWARD
FE_DFL_ENV
feclearexcept(x)
fegetexceptflag(p, x)
feraiseexcept(x)
fesetexceptflag(p, x)
fetestexcept(x)
fegetround()
fesetround(x)
fegetenv(p)
feholdexcept(p)
fesetenv(p)
feupdateenv(p)
```

<float.h>

Defines macros for implementation-defined limits.

Contents

```
FLT_RADIX
FLT_ROUNDS
FLT_EVAL_METHOD
FLT_DECIMAL_DIG
DECIMAL_DIG
FLT_MANT_DIG
DBL_MANT_DIG
LDBL_MANT_DIG
FLT_MIN_EXP
DBL_MIN_EXP
LDBL_MIN_EXP
FLT_MAX_EXP
DBL_MAX_EXP
LDBL_MAX_EXP
FLT_DIG
DBL_DIG
LDBL_DIG
FLT_MIN_10_EXP
DBL_MIN_10_EXP
LDBL_MIN_10_EXP
FLT_MAX_10_EXP
DBL_MAX_10_EXP
LDBL_MAX_10_EXP
FLT_MAX
DBL_MAX
LDBL_MAX
FLT_EPSILON
DBL_EPSILON
LDBL_EPSILON
FLT_MIN
DBL_MIN
LDBL_MIN
FLT_DECIMAL_DIG
FLT_HUGE_VAL
FLT_SUBNORM_MIN
FLT_TRUE_MIN
DBL_SUBNORM_MIN
DBL_TRUE_MIN
LDBL_SUBNORM_MIN
LDBL_TRUE_MIN
```

<inttypes.h>

Defines macros and functions for integer conversions.

Contents

```
PRIdN
PRIdLEAST
PRIdFAST
PRIdMAX
PRIdPTR
PRIiN
PRIiLEAST
PRIiFAST
PRIiMAX
PRIiPTR
PRIoN
PRIoLEAST
PRIoFAST
PRIoMAX
PRIoPTR
PRIuN
PRIuLEAST
PRIuFAST
PRIuMAX
PRIuPTR
PRIxN
PRIxLEAST
PRIxFAST
PRIxMAX
PRIxPTR
PRIXN
PRIXLEAST
PRIXFAST
PRIXMAX
PRIXPTR
SCNdN
SCNdLEAST
SCNdFAST
SCNdMAX
SCNdPTR
SCNiN
SCNiLEAST
SCNiFAST
SCNiMAX
SCNiPTR
SCNoN
SCNoLEAST
SCNoFAST
SCNoMAX
SCNoPTR
SCNuN
SCNuLEAST
SCNuFAST
SCNuMAX
SCNuPTR
SCNxN
SCNxLEAST
SCNxFAST
SCNxMAX
SCNxPTR
imaxabs(j)
imaxdiv(x, y)
strtoimax(s, p, b)
strtoumax(s, p, b)
wcstoimax(s, p, b)
wcstoumax(s, p, b)
```

<iso646.h>

Defines macros for alternative operator spellings.

Contents

```
and
and_eq
bitand
bitor
compl
not
not_eq
or
or_eq
xor
xor_eq
```

<limits.h>

Defines macros for implementation-defined limits.

Contents

```
CHAR_BIT
MB_LEN_MAX
CHAR_MIN
CHAR_MAX
SCHAR_MIN
SCHAR_MAX
UCHAR_MAX
SHRT_MIN
SHRT_MAX
USHRT_MAX
INT_MIN
INT_MAX
UINT_MAX
LONG_MIN
LONG_MAX
ULONG_MAX
LLONG_MIN
LLONG_MAX
ULLONG_MAX
```

<locale.h>

Defines functions for localization.

Contents

```
setlocale(c, s)
LC_ALL
LC_COLLATE
LC_CTYPE
LC_MONETARY
LC_NUMERIC
LC_TIME
localeconv()
```

<math.h>

Defines functions for mathematical computations.

Contents

```
HUGE_VAL
HUGE_VALF
HUGE_VALL
INFINITY
NAN
FP_INFINITE
FP_NAN
FP_NORMAL
FP_SUBNORMAL
FP_ZERO
FP_FAST_FMA
FP_FAST_FMAF
FP_FAST_FMAL
FP_ILOGB0
FP_ILOGBNAN
math_errhandling
MATH_ERRNO
MATH_ERREXCEPT
fpclassify(x)
isfinite(x)
isinf(x)
isnan(x)
isnormal(x)
signbit(x)
acos(x), acosf(x), acosl(x)
asin(x), asinf(x), asinl(x)
atan(x), atanf(x), atanl(x)
atan2(y, x), atan2f(y, x), atan2l(y, x)
cos(x), cosf(x), cosl(x)
sin(x), sinf(x), sinl(x)
tan(x), tanf(x), tanl(x)
acosh(x), acoshf(x), acoshl(x)
asinh(x), asinhf(x), asinhl(x)
atanh(x), atanhf(x), atanhl(x)
cosh(x), coshf(x), coshl(x)
sinh(x), sinhf(x), sinhl(x)
tanh(x), tanhf(x), tanhl(x)
exp(x), expf(x), expl(x)
exp2(x), exp2f(x), exp2l(x)
expm1(x), expm1f(x), expm1l(x)
frexp(x, p), frexpf(x, p), frexpl(x, p)
ldexp(x, n), ldexpf(x, n), ldexpl(x, n)
log(x), logf(x), logl(x)
log10(x), log10f(x), log10l(x)
log1p(x), log1pf(x), log1pl(x)
log2(x), log2f(x), log2l(x)
logb(x), logbf(x), logbl(x)
modf(x, p), modff(x, p), modfl(x, p)
scalbn(x, n), scalbnf(x, n), scalbnl(x, n)
scalbln(x, n), scalblnf(x, n), scalblnl(x, n)
cbrt(x), cbrtf(x), cbrtl(x)
fabs(x), fabsf(x), fabsl(x)
hypot(x, y), hypotf(x, y), hypotl(x, y)
pow(x, y), powf(x, y), powl(x, y)
sqrt(x), sqrtf(x), sqrtl(x)
erf(x), erff(x), erfl(x)
erfc(x), erfcf(x), erfcl(x)
lgamma(x), lgammaf(x), lgammal(x)
tgamma(x), tgammaf(x), tgammal(x)
ceil(x), ceilf(x), ceill(x)
floor(x), floorf(x), floorl(x)
nearbyint(x), nearbyintf(x), nearbyintl(x)
rint(x), rintf(x), rintl(x)
lrint(x), lrintf(x), lrintl(x)
nint(x), nintf(x), nintl(x)
lnint(x), lnintf(x), lnintl(x)
llrint(x), llrintf(x), llrintl(x)
llnint(x), llnintf(x), llnintl(x)
round(x), roundf(x), roundl(x)
lround(x), lroundf(x), lroundl(x)
llround(x), llroundf(x), llroundl(x)
trunc(x), truncf(x), truncl(x)
fmod(x, y), fmodf(x, y), fmodl(x, y)
remainder(x, y), remainderf(x, y), remainderl(x, y)
remquo(x, y, p), remquof(x, y, p), remquol(x, y, p)
copysign(x, y), copysignf(x, y), copysignl(x, y)
nan(s), nanf(s), nanl(s)
nextafter(x, y), nextafterf(x, y), nextafterl(x, y)
nexttoward(x, y), nexttowardf(x, y), nexttowardl(x, y)
nextup(x), nextupf(x), nextupl(x)
nextdown(x), nextdownf(x), nextdownl(x)
fdim(x, y), fdimf(x, y), fdiml(x, y)
fmax(x, y), fmaxf(x, y), fmaxl(x, y)
fmin(x, y), fminf(x, y), fminl(x, y)
fmaximum(x, y), fmaximumf(x, y), fmaximuml(x, y)
fminimum(x, y), fminimumf(x, y), fminimuml(x, y)
fmaximum_num(x, y), fmaximum_numf(x, y), fmaximum_numl(x, y)
fminimum_num(x, y), fminimum_numf(x, y), fminimum_numl(x, y)
fma(x, y, z), fmaf(x, y, z), fmal(x, y, z)
fdim(x, y), fdimf(x, y), fdiml(x, y)
fmax(x, y), fmaxf(x, y), fmaxl(x, y)
fmin(x, y), fminf(x, y), fminl(x, y)
fmaximum(x, y), fmaximumf(x, y), fmaximuml(x, y)
fminimum(x, y), fminimumf(x, y), fminimuml(x, y)
fmaximum_num(x, y), fmaximum_numf(x, y), fmaximum_numl(x, y)
fminimum_num(x, y), fminimum_numf(x, y), fminimum_numl(x, y)
fma(x, y, z), fmaf(x, y, z), fmal(x, y, z)
```

<setjmp.h>

Defines macros and functions for non-local jumps.

Contents

```
setjmp(e)
longjmp(e, v)
```

<signal.h>

Defines macros and functions for signal handling.

Contents

```
sig_atomic_t
SIGABRT
SIGFPE
SIGILL
SIGINT
SIGSEGV
SIGTERM
SIG_DFL
SIG_IGN
SIG_ERR
signal(s, f)
raise(s)
```

<stdarg.h>

Defines macros and functions for variadic arguments.

Contents

```
va_list
va_start(v, p)
va_arg(v, t)
va_end(v)
va_copy(d, s)
```

<stdbool.h>

Defines macros for boolean type.

Contents

```
bool
true
false
__bool_true_false_are_defined
```

<stdlib.h>

Defines functions for general utilities.

Contents

```
size_t
wchar_t
NULL
EXIT_FAILURE
EXIT_SUCCESS
RAND_MAX
MB_CUR_MAX
div_t
ldiv_t
lldiv_t
atof(s)
atoi(s)
atol(s)
atoll(s)
strtod(s, p)
strtof(s, p)
strtol(s, p, b)
strtoll(s, p, b)
strtoul(s, p, b)
strtoull(s, p, b)
rand()
srand(s)
aligned_alloc(a, s)
calloc(n, s)
free(p)
malloc(s)
realloc(p, s)
abort()
atexit(f)
at_quick_exit(f)
exit(s)
_Exit(s)
getenv(s)
quick_exit(s)
system(s)
bsearch(k, b, n, s, c)
qsort(b, n, s, c)
abs(j)
labs(j)
llabs(j)
div(x, y)
ldiv(x, y)
lldiv(x, y)
mblen(s, n)
mbtowc(p, s, n)
wctomb(s, w)
mbstowcs(p, s, n)
wcstombs(s, p, n)
```

<tgmath.h>

Defines macros for type-generic math.

Contents

```
cacos(z), cacosf(z), cacosl(z)
casin(z), casinf(z), casinl(z)
catan(z), catanf(z), catanl(z)
ccos(z), ccosf(z), ccosl(z)
csin(z), csinf(z), csinl(z)
ctan(z), ctanf(z), ctanl(z)
cacosh(z), cacoshf(z), cacoshl(z)
casinh(z), casinhf(z), casinhl(z)
catanh(z), catanhf(z), catanhl(z)
ccosh(z), ccoshf(z), ccoshl(z)
csinh(z), csinhf(z), csinhl(z)
ctanh(z), ctanhf(z), ctanhl(z)
cexp(z), cexpf(z), cexpl(z)
clog(z), clogf(z), clogl(z)
cabs(z), cabsf(z), cabsl(z)
cpow(z, w), cpowf(z, w), cpowl(z, w)
csqrt(z), csqrtf(z), csqrtl(z)
carg(z), cargf(z), cargl(z)
cimag(z), cimagf(z), cimagl(z)
conj(z), conjf(z), conjl(z)
cproj(z), cprojf(z), cprojl(z)
exp(x), expf(x), expl(x)
exp2(x), exp2f(x), exp2l(x)
expm1(x), expm1f(x), expm1l(x)
log(x), logf(x), logl(x)
log10(x), log10f(x), log10l(x)
log1p(x), log1pf(x), log1pl(x)
log2(x), log2f(x), log2l(x)
logb(x), logbf(x), logbl(x)
sqrt(x), sqrtf(x), sqrtl(x)
erf(x), erff(x), erfl(x)
erfc(x), erfcf(x), erfcl(x)
lgamma(x), lgammaf(x), lgammal(x)
tgamma(x), tgammaf(x), tgammal(x)
ceil(x), ceilf(x), ceill(x)
floor(x), floorf(x), floorl(x)
nearbyint(x), nearbyintf(x), nearbyintl(x)
rint(x), rintf(x), rintl(x)
lrint(x), lrintf(x), lrintl(x)
nint(x), nintf(x), nintl(x)
lnint(x), lnintf(x), lnintl(x)
llrint(x), llrintf(x), llrintl(x)
llnint(x), llnintf(x), llnintl(x)
round(x), roundf(x), roundl(x)
lround(x), lroundf(x), lroundl(x)
llround(x), llroundf(x), llroundl(x)
trunc(x), truncf(x), truncl(x)
fmod(x, y), fmodf(x, y), fmodl(x, y)
remainder(x, y), remainderf(x, y), remainderl(x, y)
remquo(x, y, p), remquof(x, y, p), remquol(x, y, p)
copysign(x, y), copysignf(x, y), copysignl(x, y)
nan(s), nanf(s), nanl(s)
nextafter(x, y), nextafterf(x, y), nextafterl(x, y)
nexttoward(x, y), nexttowardf(x, y), nexttowardl(x, y)
nextup(x), nextupf(x), nextupl(x)
nextdown(x), nextdownf(x), nextdownl(x)
fdim(x, y), fdimf(x, y), fdiml(x, y)
fmax(x, y), fmaxf(x, y), fmaxl(x, y)
fmin(x, y), fminf(x, y), fminl(x, y)
fmaximum(x, y), fmaximumf(x, y), fmaximuml(x, y)
fminimum(x, y), fminimumf(x, y), fminimuml(x, y)
fmaximum_num(x, y), fmaximum_numf(x, y), fmaximum_numl(x, y)
fminimum_num(x, y), fminimum_numf(x, y), fminimum_numl(x, y)
fma(x, y, z), fmaf(x, y, z), fmal(x, y, z)
```

<time.h>

Defines macros and functions for date and time utilities.

Contents

```
clock_t
time_t
CLOCKS_PER_SEC
TIME_UTC
struct tm
struct timespec
struct itimerspec
clock()
time(p)
difftime(e, b)
timespec_get(p, b)
timespec_getres(p, b)
timespec_cmp(a, b)
timespec_add(a, b)
timespec_sub(a, b)
timespec_div(a, b)
timespec_mul(a, b)
timespec_neg(a)
timespec_abs(a)
timespec_rem(a, b)
timespec_round(a, b)
timespec_trunc(a, b)
timespec_floor(a, b)
timespec_ceil(a, b)
timespec_nearbyint(a, b)
timespec_rint(a, b)
timespec_lrint(a, b)
timespec_nint(a, b)
timespec_lnint(a, b)
timespec_llrint(a, b)
timespec_llnint(a, b)
timespec_fmod(a, b)
timespec_remainder(a, b)
timespec_remquo(a, b, p)
timespec_copysign(a, b)
timespec_nan(s)
timespec_nextafter(a, b)
timespec_nexttoward(a, b)
timespec_nextup(a)
timespec_nextdown(a)
timespec_fdim(a, b)
timespec_fmax(a, b)
timespec_fmin(a, b)
timespec_fmaximum(a, b)
timespec_fminimum(a, b)
timespec_fmaximum_num(a, b)
timespec_fminimum_num(a, b)
timespec_fma(a, b, c)
asctime(p)
ctime(p)
gmtime(p)
localtime(p)
strftime(s, n, f, p)
asctime_r(p, s)
ctime_r(p, s)
gmtime_r(p, s)
localtime_r(p, s)
```

<uchar.h>

Defines macros and functions for Unicode utilities.

Contents

```
char16_t
char32_t
__STDC_UTF_16__
__STDC_UTF_32__
__STDC_ISO_10646__
mbrtoc16(p, s, n, ps)
c16rtomb(s, c16, ps)
mbrtoc32(p, s, n, ps)
c32rtomb(s, c32, ps)
```