

Search:

Go

C++

Information

Tutorials

Reference

Articles

Forum

Reference

C library:

<cassert> (assert.h)

<cctype> (ctype.h)

<cerrno> (errno.h)

<cfenv> (fenv.h)

<cfloat> (float.h)

<inttypes> (inttypes.h)

<ciso646> (iso646.h)

<climits> (limits.h)

<locale> (locale.h)

<cmath> (math.h)

<setjmp> (setjmp.h)

<csignal> (signal.h)

<csdarg> (stdarg.h)

<csdbool> (stdbool.h)

<csddef> (stddef.h)

<csdint> (stdint.h)

<csdio> (stdio.h)

<csdlib> (stdlib.h)

<cstring> (string.h)

<ctgmth> (tgmath.h)

<ctime> (time.h)

<cuchar> (uchar.h)

<wchar> (wchar.h)

<cwctype> (wctype.h)

Containers:

Input/Output:

Multi-threading:

Other:

header

<stdint> (stdint.h)

Integer types

This header defines a set of integral type aliases with specific width requirements, along with macros specifying their limits and macro functions to create values of these types.

Types

The following are typedefs of *fundamental integral types* or *extended integral types*.

signed type	unsigned type	description
intmax_t	uintmax_t	Integer type with the maximum width supported.
int8_t	uint8_t	Integer type with a width of exactly 8, 16, 32, or 64 bits.
int16_t	uint16_t	For signed types, negative values are represented using 2's complement.
int32_t	uint32_t	No padding bits.
int64_t	uint64_t	Optional: These typedefs are not defined if no types with such characteristics exist.*
int_least8_t	uint_least8_t	Integer type with a minimum of 8, 16, 32, or 64 bits. No other integer type exists with lesser size and at least the specified width.
int_least16_t	uint_least16_t	
int_least32_t	uint_least32_t	
int_least64_t	uint_least64_t	
int_fast8_t	uint_fast8_t	Integer type with a minimum of 8, 16, 32, or 64 bits. At least as fast as any other integer type with at least the specified width.
int_fast16_t	uint_fast16_t	
int_fast32_t	uint_fast32_t	
int_fast64_t	uint_fast64_t	
intptr_t	uintptr_t	Integer type capable of holding a value converted from a void pointer and then be converted back to that type with a value that compares equal to the original pointer. Optional: These typedefs may not be defined in some library implementations.*

Some of these typedefs may denote the same types. Therefore, function overloads should not rely on these being different.

* Notice that some types are optional (and thus, with no portability guarantees). A particular library implementation may also define additional types with other widths supported by its system. In any case, if either the signed or the unsigned version is defined, both the signed and unsigned versions are defined.

Macros

Пределы csdintтипов

Макрос	Описание	определяется как
INTMAX_MIN	Минимальное значение intmax_t	$-(2^{63-1})$ или ниже
INTMAX_MAX	Максимальное значение intmax_t	2^{63-1} или выше
UINTMAX_MAX	Максимальное значение uintmax_t	2^{64-1} или выше
INTN_MIN	Минимальное значение типа со знаком точной ширины	Ровно $-2^{(N-1)}$
INTN_MAX	Максимальное значение типа со знаком точной ширины	Ровно $2^{(N-1)}-1$
UINTN_MAX	Максимальное значение типа exact-width unsigned	Ровно 2^N-1
INT_LEASTN_MIN	Минимальное значение типа со знаком минимальной ширины	$-(2^{(N-1)}-1)$ или ниже
INT_LEASTN_MAX	Максимальное значение типа со знаком минимальной ширины	$2^{(N-1)}-1$ или выше
UINT_LEASTN_MAX	Максимальное значение типа без знака минимальной ширины	2^N-1 или выше
INT_FASTN_MIN	Минимальное значение самого быстрого подписанного типа минимальной ширины	$-(2^{(N-1)}-1)$ или ниже
INT_FASTN_MAX	Максимальное значение самого быстрого подписанного типа минимальной ширины	$2^{(N-1)}-1$ или выше
UINT_FASTN_MAX	Максимальное значение самого быстрого типа без знака минимальной ширины	2^N-1 или выше
INTPTR_MIN	Минимальное значение intptr_t	$-(2^{15-1})$ или ниже
INTPTR_MAX	Максимальное значение intptr_t	2^{15-1} или выше
UINTPTR_MAX	Максимальное значение uintptr_t	2^{16-1} или выше

Где N-единица в 8, 16, 32, 64 или любой другой ширине типа, поддерживаемой библиотекой.

Определены только макросы, соответствующие типам, поддерживаемым библиотекой.

Ограничения других типов

Пределы других стандартных интегральных типов:

Макрос	Описание	определяется как
SIZE_MAX	Максимальное значение size_t	2^{64-1} или выше
PTRDIFF_MIN	Минимальное значение ptrdiff_t	$-(2^{16-1})$ или ниже
PTRDIFF_MAX	Максимальное значение ptrdiff_t	2^{16-1} или выше
SIG_ATOMIC_MIN	Минимальное значение sig_atomic_t	если sig_atomic_t подписано: -127, или ниже если sig_atomic_t без знака: 0
SIG_ATOMIC_MAX	Максимальное значение sig_atomic_t	если sig_atomic_t подписано: 127 или выше если sig_atomic_t без знака: 255 или выше
WCHAR_MIN	Минимальное значение wchar_t	если wchar_t подписано: -127, или ниже если wchar_t без знака: 0
WCHAR_MAX	Максимальное значение wchar_t	если wchar_t подписано: 127 или выше

		, если wchar_t без знака: 255 или выше
WINT_MIN	Минимальное значение wint_t	если wint_t подписано: -32767, или ниже если wint_t без знака: 0
WINT_MAX	Максимальное значение wint_t	если wint_t подписано: 32767 или выше если wint_t без знака: 65535 или выше

Функциональные макросы

Эти функциональные макросы расширяются до целочисленных констант, подходящих для инициализации объектов указанных выше типов:

Макрос	Описание
INTMAX_C	расширяется до значения типа intmax_t
UINTMAX_C	расширяется до значения типа uintmax_t
INTN_C	расширяется до значения типа int_leastN_t
UINTN_C	расширяется до значения типа uint_leastN_t

Например:

```
INTMAX_C(2012) // expands to 2012LL or similar
```