C++ Standard Library header files

[C++](http://en.cppreference.com/w/cpp)

**Standard Library header files**

The interface of C++ standard library is defined by the following collection of header files.

|  |  |
| --- | --- |
| **Utilities library** | |
| [**<cstdlib>**](http://en.cppreference.com/w/cpp/header/cstdlib) | General purpose utilities: [program control](http://en.cppreference.com/w/cpp/utility/program), [dynamic memory allocation](http://en.cppreference.com/w/cpp/memory/c), [random numbers](http://en.cppreference.com/w/cpp/numeric/random), [sort and search](http://en.cppreference.com/w/cpp/algorithm) |
| [**<csignal>**](http://en.cppreference.com/w/cpp/header/csignal) | [Functions and macro constants for signal management](http://en.cppreference.com/w/cpp/utility/program) |
| [**<csetjmp>**](http://en.cppreference.com/w/cpp/header/csetjmp) | [Macro (and function) that saves (and jumps) to an execution context](http://en.cppreference.com/w/cpp/utility/program/setjmp) |
| [**<cstdarg>**](http://en.cppreference.com/w/cpp/header/cstdarg) | [Handling of variable length argument lists](http://en.cppreference.com/w/cpp/utility/variadic) |
| [**<typeinfo>**](http://en.cppreference.com/w/cpp/header/typeinfo) | [Runtime type information utilities](http://en.cppreference.com/w/cpp/types) |
| [**<typeindex>**](http://en.cppreference.com/w/cpp/header/typeindex) (since C++11) | [std::type\_index](http://en.cppreference.com/w/cpp/types/type_index) |
| [**<type\_traits>**](http://en.cppreference.com/w/cpp/header/type_traits) (since C++11) | [Compile-time type information](http://en.cppreference.com/w/cpp/types) |
| [**<bitset>**](http://en.cppreference.com/w/cpp/header/bitset) | [std::bitset](http://en.cppreference.com/w/cpp/utility/bitset) class template |
| [**<functional>**](http://en.cppreference.com/w/cpp/header/functional) | [Function objects](http://en.cppreference.com/w/cpp/utility/functional), designed for use with the [standard algorithms](http://en.cppreference.com/w/cpp/algorithm) |
| [**<utility>**](http://en.cppreference.com/w/cpp/header/utility) | Various [utility components](http://en.cppreference.com/w/cpp/utility) |
| [**<ctime>**](http://en.cppreference.com/w/cpp/header/ctime) | [C-style time/date utilites](http://en.cppreference.com/w/cpp/chrono/c) |
| [**<chrono>**](http://en.cppreference.com/w/cpp/header/chrono) (since C++11) | [C++ time utilites](http://en.cppreference.com/w/cpp/chrono) |
| [**<cstddef>**](http://en.cppreference.com/w/cpp/header/cstddef) | [typedefs for types such as size\_t, NULL and others](http://en.cppreference.com/w/cpp/types) |
| [**<initializer\_list>**](http://en.cppreference.com/w/cpp/header/initializer_list) (since C++11) | [std::initializer\_list](http://en.cppreference.com/w/cpp/utility/initializer_list) class template |
| [**<tuple>**](http://en.cppreference.com/w/cpp/header/tuple) (since C++11) | [std::tuple](http://en.cppreference.com/w/cpp/utility/tuple) class template |
| **Dynamic memory management** | |
| [**<new>**](http://en.cppreference.com/w/cpp/header/new) | [Low-level memory management utilities](http://en.cppreference.com/w/cpp/memory/new) |
| [**<memory>**](http://en.cppreference.com/w/cpp/header/memory) | [Higher level memory management utilities](http://en.cppreference.com/w/cpp/memory) |
| [**<scoped\_allocator>**](http://en.cppreference.com/w/cpp/header/scoped_allocator) (since C++11) | [Nested allocator class](http://en.cppreference.com/w/cpp/memory/scoped_allocator_adaptor) |
| **Numeric limits** | |
| [**<climits>**](http://en.cppreference.com/w/cpp/header/climits) | [limits of integral types](http://en.cppreference.com/w/cpp/types/climits) |
| [**<cfloat>**](http://en.cppreference.com/w/cpp/header/cfloat) | [limits of float types](http://en.cppreference.com/w/cpp/types/climits) |
| [**<cstdint>**](http://en.cppreference.com/w/cpp/header/cstdint) (since C++11) | [fixed-size types](http://en.cppreference.com/w/cpp/types/integer) and [limits of other types](http://en.cppreference.com/w/cpp/types/climits) |
| [**<cinttypes>**](http://en.cppreference.com/w/cpp/header/cinttypes) (since C++11) | [formatting macros](http://en.cppreference.com/w/cpp/types/integer), intmax\_t and uintmax\_t math and conversions |
| [**<limits>**](http://en.cppreference.com/w/cpp/header/limits) | [standardized way to query properties of fundamental types](http://en.cppreference.com/w/cpp/types/numeric_limits) |
| **Error handling** | |
| [**<exception>**](http://en.cppreference.com/w/cpp/header/exception) | [Exception handling utilities](http://en.cppreference.com/w/cpp/error) |
| [**<stdexcept>**](http://en.cppreference.com/w/cpp/header/stdexcept) | [Standard exception objects](http://en.cppreference.com/w/cpp/error) |
| [**<cassert>**](http://en.cppreference.com/w/cpp/header/cassert) | [Conditionally compiled macro that compares its argument to zero](http://en.cppreference.com/w/cpp/error/assert) |
| [**<system\_error>**](http://en.cppreference.com/w/cpp/header/system_error) (since C++11) | [defines std::error\_code, a platform-dependent error code](http://en.cppreference.com/w/cpp/error/error_code) |
| [**<cerrno>**](http://en.cppreference.com/w/cpp/header/cerrno) | [Macro containing the last error number](http://en.cppreference.com/w/cpp/error) |
| **Strings library** | |
| [**<cctype>**](http://en.cppreference.com/w/cpp/header/cctype) | [functions to determine the type contained in character data](http://en.cppreference.com/w/cpp/string/byte) |
| [**<cwctype>**](http://en.cppreference.com/w/cpp/header/cwctype) | [functions for determining the type of wide character data](http://en.cppreference.com/w/cpp/string/wide) |
| [**<cstring>**](http://en.cppreference.com/w/cpp/header/cstring) | various [narrow character string handling functions](http://en.cppreference.com/w/cpp/string/byte) |
| [**<cwchar>**](http://en.cppreference.com/w/cpp/header/cwchar) | various [wide](http://en.cppreference.com/w/cpp/string/wide)and [multibyte](http://en.cppreference.com/w/cpp/string/multibyte" \o "cpp/string/multibyte) string handling functions |
| [**<cuchar>**](http://en.cppreference.com/w/cpp/header/cuchar) (since C++11) | C-style [Unicode character conversion functions](http://en.cppreference.com/w/cpp/string/multibyte) |
| [**<string>**](http://en.cppreference.com/w/cpp/header/string) | [std::basic\_string](http://en.cppreference.com/w/cpp/string/basic_string) class template |
| **Containers library** | |
| [**<array>**](http://en.cppreference.com/w/cpp/header/array) (since C++11) | [std::array](http://en.cppreference.com/w/cpp/container/array) container |
| [**<vector>**](http://en.cppreference.com/w/cpp/header/vector) | [std::vector](http://en.cppreference.com/w/cpp/container/vector) container |
| [**<deque>**](http://en.cppreference.com/w/cpp/header/deque) | [std::deque](http://en.cppreference.com/w/cpp/container/deque) container |
| [**<list>**](http://en.cppreference.com/w/cpp/header/list) | [std::list](http://en.cppreference.com/w/cpp/container/list) container |
| [**<forward\_list>**](http://en.cppreference.com/w/cpp/header/forward_list) (since C++11) | [std::forward\_list](http://en.cppreference.com/w/cpp/container/forward_list) container |
| [**<set>**](http://en.cppreference.com/w/cpp/header/set) | [std::set](http://en.cppreference.com/w/cpp/container/set) and [std::multiset](http://en.cppreference.com/w/cpp/container/multiset" \o "cpp/container/multiset) associative containers |
| [**<map>**](http://en.cppreference.com/w/cpp/header/map) | [std::map](http://en.cppreference.com/w/cpp/container/map) and [std::multimap](http://en.cppreference.com/w/cpp/container/multimap" \o "cpp/container/multimap) associative containers |
| [**<unordered\_set>**](http://en.cppreference.com/w/cpp/header/unordered_set) (since C++11) | [std::unordered\_set](http://en.cppreference.com/w/cpp/container/unordered_set) and [std::unordered\_multiset](http://en.cppreference.com/w/cpp/container/unordered_multiset" \o "cpp/container/unordered multiset)unordered associative containers |
| [**<unordered\_map>**](http://en.cppreference.com/w/cpp/header/unordered_map) (since C++11) | [std::unordered\_map](http://en.cppreference.com/w/cpp/container/unordered_map) and [std::unordered\_multimap](http://en.cppreference.com/w/cpp/container/unordered_multimap" \o "cpp/container/unordered multimap)unordered associative containers |
| [**<stack>**](http://en.cppreference.com/w/cpp/header/stack) | [std::stack](http://en.cppreference.com/w/cpp/container/stack) container adaptor |
| [**<queue>**](http://en.cppreference.com/w/cpp/header/queue) | [std::queue](http://en.cppreference.com/w/cpp/container/queue) and [std::priority\_queue](http://en.cppreference.com/w/cpp/container/priority_queue" \o "cpp/container/priority queue) container adaptors |
| **Algorithms library** | |
| [**<algorithm>**](http://en.cppreference.com/w/cpp/header/algorithm) | [Algorithms that operate on containers](http://en.cppreference.com/w/cpp/algorithm) |
| **Iterators library** | |
| [**<iterator>**](http://en.cppreference.com/w/cpp/header/iterator) | [Container iterators](http://en.cppreference.com/w/cpp/iterator) |
| **Numerics library** | |
| [**<cmath>**](http://en.cppreference.com/w/cpp/header/cmath) | [Common mathematics functions](http://en.cppreference.com/w/cpp/numeric/math) |
| [**<complex>**](http://en.cppreference.com/w/cpp/header/complex) | [Complex number type](http://en.cppreference.com/w/cpp/numeric/complex) |
| [**<valarray>**](http://en.cppreference.com/w/cpp/header/valarray) | [Class for representing and manipulating arrays of values](http://en.cppreference.com/w/cpp/numeric/valarray) |
| [**<random>**](http://en.cppreference.com/w/cpp/header/random) (since C++11) | [Random number generators and distributions](http://en.cppreference.com/w/cpp/numeric/random) |
| [**<numeric>**](http://en.cppreference.com/w/cpp/header/numeric) | [Numeric operations on values in containers](http://en.cppreference.com/w/cpp/numeric) |
| [**<ratio>**](http://en.cppreference.com/w/cpp/header/ratio) (since C++11) | [Compile-time rational arithmetic](http://en.cppreference.com/w/cpp/numeric/ratio) |
| [**<cfenv>**](http://en.cppreference.com/w/cpp/header/cfenv) (since C++11) | [Floating-point environment](http://en.cppreference.com/w/cpp/numeric/fenv) access functions |
| **Input/output library** | |
| [**<iosfwd>**](http://en.cppreference.com/w/cpp/header/iosfwd) | forward declarations of all classes in the input/output library |
| [**<ios>**](http://en.cppreference.com/w/cpp/header/ios) | [std::ios\_base](http://en.cppreference.com/w/cpp/io/ios_base) class, [std::basic\_ios](http://en.cppreference.com/w/cpp/io/basic_ios" \o "cpp/io/basic ios) class template and several typedefs |
| [**<istream>**](http://en.cppreference.com/w/cpp/header/istream) | [std::basic\_istream](http://en.cppreference.com/w/cpp/io/basic_istream) class template and several typedefs |
| [**<ostream>**](http://en.cppreference.com/w/cpp/header/ostream) | [std::basic\_ostream](http://en.cppreference.com/w/cpp/io/basic_ostream), [std::basic\_iostream](http://en.cppreference.com/w/cpp/io/basic_iostream" \o "cpp/io/basic iostream) class templates and several typedefs |
| [**<iostream>**](http://en.cppreference.com/w/cpp/header/iostream) | several standard stream objects |
| [**<fstream>**](http://en.cppreference.com/w/cpp/header/fstream) | [std::basic\_fstream](http://en.cppreference.com/w/cpp/io/basic_fstream), [std::basic\_ifstream](http://en.cppreference.com/w/cpp/io/basic_ifstream),[std::basic\_ofstream](http://en.cppreference.com/w/cpp/io/basic_ofstream) class templates and several typedefs |
| [**<sstream>**](http://en.cppreference.com/w/cpp/header/sstream) | [std::basic\_stringstream](http://en.cppreference.com/w/cpp/io/basic_stringstream), [std::basic\_istringstream](http://en.cppreference.com/w/cpp/io/basic_istringstream),[std::basic\_ostringstream](http://en.cppreference.com/w/cpp/io/basic_ostringstream) class templates and several typedefs |
| [**<strstream>**](http://en.cppreference.com/w/cpp/header/strstream) | [std::strstream](http://en.cppreference.com/w/cpp/io/strstream), [std::istrstream](http://en.cppreference.com/w/cpp/io/istrstream), [std::ostrstream](http://en.cppreference.com/w/cpp/io/ostrstream)(deprecated) |
| [**<iomanip>**](http://en.cppreference.com/w/cpp/header/iomanip) | [Helper functions to control the format or input and output](http://en.cppreference.com/w/cpp/io/manip) |
| [**<streambuf>**](http://en.cppreference.com/w/cpp/header/streambuf) | [std::basic\_streambuf](http://en.cppreference.com/w/cpp/io/basic_streambuf) class template |
| [**<cstdio>**](http://en.cppreference.com/w/cpp/header/cstdio) | [C-style input-output functions](http://en.cppreference.com/w/cpp/io/c) |
| **Localization library** | |
| [**<locale>**](http://en.cppreference.com/w/cpp/header/locale) | [Localization utilities](http://en.cppreference.com/w/cpp/locale) |
| [**<clocale>**](http://en.cppreference.com/w/cpp/header/clocale) | [C localization utilities](http://en.cppreference.com/w/cpp/locale) |
| [**<codecvt>**](http://en.cppreference.com/w/cpp/header/codecvt) (since C++11) | [Unicode conversion facilities](http://en.cppreference.com/w/cpp/locale) |
| **Regular Expressions library** | |
| [**<regex>**](http://en.cppreference.com/w/cpp/header/regex) (since C++11) | [Classes, algorithms and iterators to support regular expression processing](http://en.cppreference.com/w/cpp/regex) |
| **Atomic Operations library** | |
| [**<atomic>**](http://en.cppreference.com/w/cpp/header/atomic) (since C++11) | [Atomic operations library](http://en.cppreference.com/w/cpp/atomic) |
| **Thread support library** | |
| [**<thread>**](http://en.cppreference.com/w/cpp/header/thread) (since C++11) | [std::thread](http://en.cppreference.com/w/cpp/thread/thread) class and [supporting functions](http://en.cppreference.com/w/cpp/thread) |
| [**<mutex>**](http://en.cppreference.com/w/cpp/header/mutex) (since C++11) | [mutual exclusion primitives](http://en.cppreference.com/w/cpp/thread) |
| [**<shared\_mutex>**](http://en.cppreference.com/w/cpp/header/shared_mutex) (since C++14) | [shared mutual exclusion primitives](http://en.cppreference.com/w/cpp/thread) |
| [**<future>**](http://en.cppreference.com/w/cpp/header/future) (since C++11) | [primitives for asynchronous computations](http://en.cppreference.com/w/cpp/thread) |
| [**<condition\_variable>**](http://en.cppreference.com/w/cpp/header/condition_variable) (since C++11) | [thread waiting conditions](http://en.cppreference.com/w/cpp/thread) |
| **Experimental libraries** | |
| [**<experimental/algorithm>**](http://en.cppreference.com/mwiki/index.php?title=cpp/header/experimental/algorithm&action=edit&redlink=1) (library fundamentals TS) | [Standard libraries extensions](http://en.cppreference.com/w/cpp/experimental/lib_extensions) and [Extensions for Parallelism](http://en.cppreference.com/w/cpp/experimental/parallelism) |
| [**<experimental/any>**](http://en.cppreference.com/w/cpp/header/experimental/any) (library fundamentals TS) | [Standard libraries extensions](http://en.cppreference.com/w/cpp/experimental/lib_extensions) |
| [**<experimental/chrono>**](http://en.cppreference.com/mwiki/index.php?title=cpp/header/experimental/chrono&action=edit&redlink=1) (library fundamentals TS) | [Standard libraries extensions](http://en.cppreference.com/w/cpp/experimental/lib_extensions) |
| [**<experimental/deque>**](http://en.cppreference.com/mwiki/index.php?title=cpp/header/experimental/deque&action=edit&redlink=1) (library fundamentals TS) | [Standard libraries extensions](http://en.cppreference.com/w/cpp/experimental/lib_extensions) |
| [**<experimental/execution\_policy>**](http://en.cppreference.com/mwiki/index.php?title=cpp/header/experimental/execution_policy&action=edit&redlink=1) (parallelism TS) | [Extensions for Parallelism](http://en.cppreference.com/w/cpp/experimental/parallelism) |
| [**<experimental/exception\_list>**](http://en.cppreference.com/mwiki/index.php?title=cpp/header/experimental/exception_list&action=edit&redlink=1) (parallelism TS) | [Extensions for Parallelism](http://en.cppreference.com/w/cpp/experimental/parallelism) |
| [**<experimental/filesystem>**](http://en.cppreference.com/w/cpp/header/experimental/filesystem) (filesystem TS) | [Filesystem library](http://en.cppreference.com/w/cpp/experimental/fs) |
| [**<experimental/forward\_list>**](http://en.cppreference.com/mwiki/index.php?title=cpp/header/experimental/forward_list&action=edit&redlink=1) (library fundamentals TS) | [Standard libraries extensions](http://en.cppreference.com/w/cpp/experimental/lib_extensions) |
| [**<experimental/future>**](http://en.cppreference.com/mwiki/index.php?title=cpp/header/experimental/future&action=edit&redlink=1) (library fundamentals TS) | [Standard libraries extensions](http://en.cppreference.com/w/cpp/experimental/lib_extensions) |
| [**<experimental/list>**](http://en.cppreference.com/mwiki/index.php?title=cpp/header/experimental/list&action=edit&redlink=1) (library fundamentals TS) | [Standard libraries extensions](http://en.cppreference.com/w/cpp/experimental/lib_extensions) |
| [**<experimental/functional>**](http://en.cppreference.com/mwiki/index.php?title=cpp/header/experimental/functional&action=edit&redlink=1) (library fundamentals TS) | [Standard libraries extensions](http://en.cppreference.com/w/cpp/experimental/lib_extensions) |
| [**<experimental/map>**](http://en.cppreference.com/mwiki/index.php?title=cpp/header/experimental/map&action=edit&redlink=1) (library fundamentals TS) | [Standard libraries extensions](http://en.cppreference.com/w/cpp/experimental/lib_extensions) |
| [**<experimental/memory>**](http://en.cppreference.com/mwiki/index.php?title=cpp/header/experimental/memory&action=edit&redlink=1) (library fundamentals TS) | [Standard libraries extensions](http://en.cppreference.com/w/cpp/experimental/lib_extensions) |
| [**<experimental/memory\_resource>**](http://en.cppreference.com/w/cpp/header/experimental/memory_resource) (library fundamentals TS) | [Standard libraries extensions](http://en.cppreference.com/w/cpp/experimental/lib_extensions) |
| [**<experimental/numeric>**](http://en.cppreference.com/mwiki/index.php?title=cpp/header/experimental/numeric&action=edit&redlink=1) (parallelism TS) | [Extensions for Parallelism](http://en.cppreference.com/w/cpp/experimental/parallelism) |
| [**<experimental/optional>**](http://en.cppreference.com/w/cpp/header/experimental/optional) (library fundamentals TS) | [Standard libraries extensions](http://en.cppreference.com/w/cpp/experimental/lib_extensions) |
| [**<experimental/ratio>**](http://en.cppreference.com/mwiki/index.php?title=cpp/header/experimental/ratio&action=edit&redlink=1) (library fundamentals TS) | [Standard libraries extensions](http://en.cppreference.com/w/cpp/experimental/lib_extensions) |
| [**<experimental/regex>**](http://en.cppreference.com/mwiki/index.php?title=cpp/header/experimental/regex&action=edit&redlink=1) (library fundamentals TS) | [Standard libraries extensions](http://en.cppreference.com/w/cpp/experimental/lib_extensions) |
| [**<experimental/set>**](http://en.cppreference.com/mwiki/index.php?title=cpp/header/experimental/set&action=edit&redlink=1) (library fundamentals TS) | [Standard libraries extensions](http://en.cppreference.com/w/cpp/experimental/lib_extensions) |
| [**<experimental/string>**](http://en.cppreference.com/mwiki/index.php?title=cpp/header/experimental/string&action=edit&redlink=1) (library fundamentals TS) | [Standard libraries extensions](http://en.cppreference.com/w/cpp/experimental/lib_extensions) |
| [**<experimental/string\_view>**](http://en.cppreference.com/w/cpp/header/experimental/string_view) (library fundamentals TS) | [Standard libraries extensions](http://en.cppreference.com/w/cpp/experimental/lib_extensions) |
| [**<experimental/system\_error>**](http://en.cppreference.com/mwiki/index.php?title=cpp/header/experimental/system_error&action=edit&redlink=1) (library fundamentals TS) | [Standard libraries extensions](http://en.cppreference.com/w/cpp/experimental/lib_extensions) |
| [**<experimental/tuple>**](http://en.cppreference.com/mwiki/index.php?title=cpp/header/experimental/tuple&action=edit&redlink=1) (library fundamentals TS) | [Standard libraries extensions](http://en.cppreference.com/w/cpp/experimental/lib_extensions) |
| [**<experimental/type\_traits>**](http://en.cppreference.com/mwiki/index.php?title=cpp/header/experimental/type_traits&action=edit&redlink=1) (library fundamentals TS) | [Standard libraries extensions](http://en.cppreference.com/w/cpp/experimental/lib_extensions) |
| [**<experimental/unordered\_map>**](http://en.cppreference.com/mwiki/index.php?title=cpp/header/experimental/unordered_map&action=edit&redlink=1) (library fundamentals TS) | [Standard libraries extensions](http://en.cppreference.com/w/cpp/experimental/lib_extensions) |
| [**<experimental/unordered\_set>**](http://en.cppreference.com/mwiki/index.php?title=cpp/header/experimental/unordered_set&action=edit&redlink=1) (library fundamentals TS) | [Standard libraries extensions](http://en.cppreference.com/w/cpp/experimental/lib_extensions) |
| [**<experimental/utility>**](http://en.cppreference.com/mwiki/index.php?title=cpp/header/experimental/utility&action=edit&redlink=1) (library fundamentals TS) | [Standard libraries extensions](http://en.cppreference.com/w/cpp/experimental/lib_extensions) |
| [**<experimental/vector>**](http://en.cppreference.com/mwiki/index.php?title=cpp/header/experimental/vector&action=edit&redlink=1) (library fundamentals TS) | [Standard libraries extensions](http://en.cppreference.com/w/cpp/experimental/lib_extensions) |
| **C compatibility headers** | |
| [**<ciso646>**](http://en.cppreference.com/w/cpp/header/ciso646) | empty header. The macros that appear in iso646.h in C are keywords in C++ |
| [**<ccomplex>**](http://en.cppreference.com/w/cpp/header/ccomplex) (since C++11) | simply includes the header <complex> |
| [**<ctgmath>**](http://en.cppreference.com/w/cpp/header/ctgmath) (since C++11) | simply includes the headers <ccomplex> and <cmath>: the overloads equivalent to the contents of the C headertgmath.h are already provided by <cmath> and <complex> |
| [**<cstdalign>**](http://en.cppreference.com/w/cpp/header/cstdalign) (since C++11) | defines one [compatibility macro constant](http://en.cppreference.com/w/cpp/types) |
| [**<cstdbool>**](http://en.cppreference.com/w/cpp/header/cstdbool) (since C++11) | defines one [compatibility macro constant](http://en.cppreference.com/w/cpp/types) |
| **Deprecated headers** | |
| **<assert.h>** (deprecated) | behaves as if each name from [<cassert>](http://en.cppreference.com/w/cpp/header/cassert) is placed in global namespace |
| **<complex.h>** (deprecated) | behaves as if each name from [<ccomplex>](http://en.cppreference.com/w/cpp/header/ccomplex) is placed in global namespace |
| **<ctype.h>** (deprecated) | behaves as if each name from [<cctype>](http://en.cppreference.com/w/cpp/header/cctype) is placed in global namespace |
| **<errno.h>** (deprecated) | behaves as if each name from [<cerrno>](http://en.cppreference.com/w/cpp/header/cerrno) is placed in global namespace |
| **<fenv.h>** (deprecated) | behaves as if each name from [<cfenv>](http://en.cppreference.com/w/cpp/header/cfenv) is placed in global namespace |
| **<float.h>** (deprecated) | behaves as if each name from [<cfloat>](http://en.cppreference.com/w/cpp/header/cfloat) is placed in global namespace |
| **<inttypes.h>** (deprecated) | behaves as if each name from [<cinttypes>](http://en.cppreference.com/w/cpp/header/cinttypes) is placed in global namespace |
| **<iso646.h>** (deprecated) | behaves as if each name from [<ciso646>](http://en.cppreference.com/w/cpp/header/ciso646) is placed in global namespace |
| **<limits.h>** (deprecated) | behaves as if each name from [<climits>](http://en.cppreference.com/w/cpp/header/climits) is placed in global namespace |
| **<locale.h>** (deprecated) | behaves as if each name from [<clocale>](http://en.cppreference.com/w/cpp/header/clocale) is placed in global namespace |
| **<math.h>** (deprecated) | behaves as if each name from [<cmath>](http://en.cppreference.com/w/cpp/header/cmath) is placed in global namespace |
| **<setjmp.h>** (deprecated) | behaves as if each name from [<csetjmp>](http://en.cppreference.com/w/cpp/header/csetjmp) is placed in global namespace |
| **<signal.h>** (deprecated) | behaves as if each name from [<csignal>](http://en.cppreference.com/w/cpp/header/csignal) is placed in global namespace |
| **<stdalign.h>** (deprecated) | behaves as if each name from [<cstdalign>](http://en.cppreference.com/w/cpp/header/cstdalign) is placed in global namespace |
| **<stdarg.h>** (deprecated) | behaves as if each name from [<cstdarg>](http://en.cppreference.com/w/cpp/header/cstdarg) is placed in global namespace |
| **<stdbool.h>** (deprecated) | behaves as if each name from [<cstdbool>](http://en.cppreference.com/w/cpp/header/cstdbool) is placed in global namespace |
| **<stddef.h>** (deprecated) | behaves as if each name from [<cstddef>](http://en.cppreference.com/w/cpp/header/cstddef) is placed in global namespace |
| **<stdint.h>** (deprecated) | behaves as if each name from [<cstdint>](http://en.cppreference.com/w/cpp/header/cstdint) is placed in global namespace |
| **<stdio.h>** (deprecated) | behaves as if each name from [<cstdio>](http://en.cppreference.com/w/cpp/header/cstdio) is placed in global namespace |
| **<stdlib.h>** (deprecated) | behaves as if each name from [<cstdlib>](http://en.cppreference.com/w/cpp/header/cstdlib) is placed in global namespace |
| **<string.h>** (deprecated) | behaves as if each name from [<cstring>](http://en.cppreference.com/w/cpp/header/cstring) is placed in global namespace |
| **<tgmath.h>** (deprecated) | behaves as if each name from [<ctgmath>](http://en.cppreference.com/w/cpp/header/ctgmath) is placed in global namespace |
| **<time.h>** (deprecated) | behaves as if each name from [<ctime>](http://en.cppreference.com/w/cpp/header/ctime) is placed in global namespace |
| **<uchar.h>** (deprecated) | behaves as if each name from [<cuchar>](http://en.cppreference.com/w/cpp/header/cuchar) is placed in global namespace |
| **<wchar.h>** (deprecated) | behaves as if each name from [<cwchar>](http://en.cppreference.com/w/cpp/header/cwchar) is placed in global namespace |
| **<wctype.h>** (deprecated) | behaves as if each name from [<cwctype>](http://en.cppreference.com/w/cpp/header/cwctype) is placed in global namespace |

**See also**

|  |
| --- |
| [**C documentation**](http://en.cppreference.com/w/c/header) for **C Standard Library header files** |

* [Support us](http://www.cppreference.com/support)
* [Recent changes](http://en.cppreference.com/w/Special:RecentChanges)
* [FAQ](http://en.cppreference.com/w/Cppreference:FAQ)
* [Offline version](http://en.cppreference.com/w/Cppreference:Archives)
* [What links here](http://en.cppreference.com/w/Special:WhatLinksHere/cpp/header)
* [Related changes](http://en.cppreference.com/w/Special:RecentChangesLinked/cpp/header)
* [Upload file](http://upload.cppreference.com/w/Special:Upload)
* [Special pages](http://en.cppreference.com/w/Special:SpecialPages)
* [Printable version](http://en.cppreference.com/mwiki/index.php?title=cpp/header&printable=yes)
* [Permanent link](http://en.cppreference.com/mwiki/index.php?title=cpp/header&oldid=77315)
* [Page information](http://en.cppreference.com/mwiki/index.php?title=cpp/header&action=info)
* In other languages
* [العربية](http://ar.cppreference.com/w/cpp/header)
* [Deutsch](http://de.cppreference.com/w/cpp/header)
* [Español](http://es.cppreference.com/w/cpp/header)
* [Français](http://fr.cppreference.com/w/cpp/header)
* [Italiano](http://it.cppreference.com/w/cpp/header)
* [日本語](http://ja.cppreference.com/w/cpp/header)
* [Português](http://pt.cppreference.com/w/cpp/header)
* [Русский](http://ru.cppreference.com/w/cpp/header)
* [中文](http://zh.cppreference.com/w/cpp/header)
* This page was last modified on 21 March 2015, at 08:41.
* [Privacy policy](http://en.cppreference.com/w/Cppreference:Privacy_policy)

* [About cppreference.com](http://en.cppreference.com/w/Cppreference:About)

* [Disclaimers](http://en.cppreference.com/w/Cppreference:General_disclaimer)
* [Powered by MediaWiki](http://www.mediawiki.org/) [Powered by GeSHi](http://qbnz.com/highlighter/) [Hosted by Tiger Technologies](http://www.tigertech.net/referral/cppreference.com)