cplusplus.com
Search: Go
Not logged in
Reference <ostream> endl
register log in

C++
Information
Tutorials
Reference
Articles
Forum

Reference
C library:
Containers:
Input/Output:
<fstream>
<iomanip>
<ios>
<iosfwd>
<iostream>
<istream>
<ostream>
<sstream>
<streambuf>
Multi-threading:
Other:

<ostream>
class templates:
basic_ostream
classes:
ostream
wostream
manipulators:
endl
ends
flush

function

std::endl

<ostream> <iostream>

for ostream	`ostream& endl (ostream& os);`

basic template	`template <class charT, class traits>`
	`basic_ostream<charT,traits>& endl (basic_ostream<charT,traits>& os);`

Insert newline and flush

Inserts a new-line character and *flushes* the stream.

Its behavior is equivalent to calling `os.put('\n')` (or `os.put(os.widen('\n'))` for character types other than `char`), and then `os.flush()`.

Parameters

os

Output stream object affected.
Because this function is a manipulator, it is designed to be used alone with no arguments in conjunction with the *insertion* (`<<`) operations on output streams (see example below).

Return Value

Argument *os*.

Errors are signaled by modifying the *internal state flags* of *os*:

flag	error
`eofbit`	-
`failbit`	May be set if the construction of a `sentry` object failed.
`badbit`	Either the insertion on the stream (or the synchronization) failed, or some other error happened (such as when this function catches an exception thrown by an internal operation). When set, the integrity of the stream may have been affected.

Multiple flags may be set on *os* by a single operation.

If the operation sets an *internal state flag* of *os* that was registered using its member `exceptions`, the function throws an exception of type `ios_base::failure`.

Example

```
// endl example
#include <iostream>     // std::cout, std::end

int main () {

  int a=100;
  double b=3.14;

  std::cout << a;
  std::cout << std::endl;              // manipulator inserted alone
  std::cout << b << std::endl << a*b;  // manipulator in concatenated insertion
  std::endl (std::cout);               // endl called as a regular function

  return 0;
}
```

Output:

```

100
3.14
314
```

Data races

Modifies *os*.
Concurrent access to the same stream object may cause data races, except for the standard stream objects (`cout`, `cerr`, `clog`, `wcout`, `wcerr` and `wclog`) when these are *synchronized with stdio* (in this case, no data races are initiated, although no guarantees are given on the order in which characters from multiple threads are inserted).

Exception safety

Basic guarantee: if an exception is thrown, *os* is in a valid state.
It throws an exception of member type `failure` if the resulting *error state flag* of *os* is not `goodbit` and member

`exceptions` was set to throw for that state.
Any exception thrown by an internal operation is caught and handled by the function, setting *os*'s `badbit`. If `badbit` was set on the last call to `exceptions` for *os*, the function rethrows the caught exception.

See also

flush	Flush stream buffer (function)
ostream::operator<<	Insert formatted output (public member function)