C++
Information
Tutorials
Reference
Articles
Forum

Reference
C library:
Containers:
Input/Output:
Multi-threading:
Other:
<algorithm>
<bitset>
<chrono>
<codecvt>
<complex>
<exception>
<functional>
<initializer_list>
<iterator>
<limits>
<locale>
<memory>
<new>
<numeric>
<random>
<ratio>
<regex>
<stdexcept>
<string>
<system_error>
<tuple>
<typeindex>
<typeinfo>
<type_traits>
<utility>
<valarray>

<string>
class templates:
basic_string
char_traits
classes:
string
u16string
u32string
wstring
functions:
stod
stof
stoi
stol
stold
stoll
stoul
stoull
to_string
to_wstring

string
string::string
string::~string
member functions:
string::append
string::assign
string::at
string::back
string::begin
string::capacity
string::cbegin
string::cend
string::clear
string::compare
string::copy
string::crbegin

function

std::getline (string)

<string>

C++98 C++11

```
(1)  istream& getline (istream& is, string& str, char delim);
(2)  istream& getline (istream& is, string& str);
```

Get line from stream into string

Extracts characters from *is* and stores them into *str* until the delimitation character *delim* is found (or the newline character, `'\n'`, for *(2)*).

The extraction also stops if the end of file is reached in *is* or if some other error occurs during the input operation.

If the delimiter is found, it is extracted and discarded (i.e. it is not stored and the next input operation will begin after it).

Note that any content in *str* before the call is replaced by the newly extracted sequence.

Each extracted character is appended to the string as if its member push_back was called.

Parameters

`is`
: istream object from which characters are extracted.

`str`
: string object where the extracted line is stored.
The contents in the string before the call (if any) are discarded and replaced by the extracted line.

Return Value

The same as parameter *is*.

A call to this function may set any of the internal state flags of *is* if:

flag	error
`eofbit`	The end of the source of characters is reached during its operations.
`failbit`	The input obtained could not be interpreted as a valid textual representation of an object of this type. In this case, *distr* preserves the parameters and internal data it had before the call. Notice that some `eofbit` cases will also set `failbit`.
`badbit`	An error other than the above happened.

(see ios_base::iostate for more info on these)

Additionally, in any of these cases, if the appropriate flag has been set with *is*'s member function ios::exceptions, an exception of type ios_base::failure is thrown.

Example

```
1  // extract to string
2  #include <iostream>
3  #include <string>
4
5  int main ()
6  {
7    std::string name;
8
9    std::cout << "Please, enter your full name: ";
10   std::getline (std::cin,name);
11   std::cout << "Hello, " << name << "!\n";
12
13   return 0;
14 }
```

Edit & Run

Complexity

Unspecified, but generally linear in the resulting length of *str*.

Iterator validity

Any iterators, pointers and references related to *str* may be invalidated.

Data races

Both objects, *is* and *str*, are modified.

Exception safety

Basic guarantee: if an exception is thrown, both *is* and *str* end up in a valid state.

See also

istream::getline	Get line (public member function)
operator>> (string)	Extract string from stream (function)