C++
Information
Tutorials
Reference
Articles
Forum

Reference
C library:
Containers:
<array>
<deque>
<forward_list>
<list>
<map>
<queue>
<set>
<stack>
<unordered_map>
<unordered_set>
<vector>
Input/Output:
Multi-threading:
Other:

<vector>
vector
vector<bool>

vector
vector::vector
vector::~vector
member functions:
vector::assign
vector::at
vector::back
vector::begin
vector::capacity
vector::cbegin
vector::cend
vector::clear
vector::crbegin
vector::crend
vector::data
vector::emplace
vector::emplace_back
vector::empty
vector::end
vector::erase
vector::front
vector::get_allocator
vector::insert
vector::max_size
vector::operator=
vector::operator[]
vector::pop_back
vector::push_back
vector::rbegin
vector::rend
vector::reserve
vector::resize
vector::shrink_to_fit
vector::size
vector::swap
non-member overloads:
relational operators (vector)
swap (vector)

public member function

std::vector::max_size

<vector>

C++98 C++11

```
size_type max_size() const noexcept;
```

Return maximum size

Returns the maximum number of elements that the vector can hold.

This is the maximum potential size the container can reach due to known system or library implementation limitations, but the container is by no means guaranteed to be able to reach that size: it can still fail to allocate storage at any point before that size is reached.

Parameters

none

Return Value

The maximum number of elements a vector container can hold as content.

Member type `size_type` is an unsigned integral type.

Example

```
// comparing size, capacity and max_size
#include <iostream>
#include <vector>

int main ()
{
  std::vector<int> myvector;

  // set some content in the vector:
  for (int i=0; i<100; i++) myvector.push_back(i);

  std::cout << "size: " << myvector.size() << "\n";
  std::cout << "capacity: " << myvector.capacity() << "\n";
  std::cout << "max_size: " << myvector.max_size() << "\n";
  return 0;
}
```

A possible output for this program could be:

```
size: 100
capacity: 128
max_size: 1073741823
```

Complexity

Constant.

Iterator validity

No changes.

Data races

The container is accessed.
No contained elements are accessed: concurrently accessing or modifying them is safe.

Exception safety

No-throw guarantee: this member function never throws exceptions.

See also

vector::capacity	Return size of allocated storage capacity (public member function)
vector::size	Return size (public member function)
vector::resize	Change size (public member function)