Doc. No.: WG21/N1023
X3J16/96-0205
Date: November 12, 1996
Project: C++ Standard Library
Reply to: Pete Becker
pbecker@oec.com

Clause 23 (Containers Library) Motions

Motion (to close various issues without action)

Move we close the following clause 23 issues without taking any action: 23-043, -063, -069, -074, -077, -079, -080 in N1014 = 96-0196[1].

Motion (to adopt various changes to clause 23) :

Amend the WP as follows, thus closing issues 23-064 and 23-066:
-- add the following sentence to the end of clause 23.1.2 [lib.associative.reqmts], following paragraph 9:

> When an associative container is constructed by passing a comparison object the container shall not store a pointer or reference to the passed comparison object, even if that object is passed by reference. When an associative container is copied, either through a copy constructor or an assignment operator, the target container shall then use the comparison object from the container being copied, as if that comparison object had been passed to the target container in its constructor.

Amend the WP as follows, thus closing issue 23-065:
-- add the following sentence to the end of paragraph 2 of clause 23.1.2 [lib.associative.reqmts]:

> This comparison object may be a pointer to function or an object of a type with an appropriate function call operator.

-- add the following sentence to the end of paragraph 7 of the introductory text of clause 25 [lib.algorithms]:

> This function object may be a pointer to function or an object of a type with an appropriate function call operator.

Amend the WP as follows, thus closing issue 23-067:
-- strike the Returns section in the definition of operator<< in clause 23.3.5.3 [lib.bitset.operators] and replace it with the following:

> Returns: os << x.to_string<charT,traits,allocator<charT> >()

Amend the WP as follows, thus closing issue 23-068:

---

[1] Note also that 23-071 has been closed. It was approved at a previous meeting and should have been closed at that time.

-- strike the explicit bitset constructor from paragraph 1 of clause 23.3.5 [lib.template.bitset] and replace it with the following:

```
explicit bitset(const basic_string<charT,traits,Allocator>& str,
        basic_string<charT,traits,Allocator>::size_type pos = 0,
        basic_string<charT,traits,Allocator>::size_type n =
                basic_string<charT,traits,Allocator>::npos);
```

Amend the WP as follows, thus closing issue 23-070:

-- strike the Returns section in the definition of capacity() in clause 23.2.4.2 [lib.vector.capacity] and replace it with the following:

> Returns: the total number of elements that the vector can hold without requiring reallocation.

-- strike the last sentence of paragraph 4 of clause 23.2.4.2 [lib.vector.capacity] and replace it with the following:

> It is guaranteed that no reallocation takes place during insertions that happen after a call to reserve() until the time when an insertion would make the size of the vector greater than the size specified in the most recent call to reserve().

Amend the WP as follows, thus closing issue 23-072:

-- replace all occurrences of the formal argument 'T' with 'U' in the definition of the second assign() member template in clause 23.2.4 [lib.vector].

-- replace all occurrences of the formal argument 'T' with 'U' in the definition of the second assign() member template in clause 23.2.5 [lib.vector.bool].

Amend the WP as follows, thus closing issue 23-073:

-- modify the definition of std::map::value_compare::operator() in clause 23.3.1 [lib.map] as indicated by italics:

```
bool operator()(const value_type& x, const value_type& y) const {
        return comp(x.first, y.first);
}
```

-- modify the definition of std::multimap::value_compare::operator() in clause 23.3.2 [lib.multimap] as indicated by italics:

```
bool operator()(const value_type& x, const value_type& y) const {
        return comp(x.first, y.first);
}
```

Amend the WP as follows, thus closing issue 23-075:

-- strike the following text from the Effects section of clause 23.2.1.2 [lib.deque.capacity]

```
erase(begin()+sz, s.end());
```

and replace it with the following:

```
erase(begin()+sz, end());
```

-- strike the following text from the Effects section of clause 23.2.2.2 [lib.list.capacity]

```
erase(begin()+sz, s.end());
```

and replace it with the following:

```
erase(begin()+sz, end());
```

-- strike the following text from the Effects section of clause 23.2.4.2 [lib.vector.capacity]

```
erase(begin()+sz, s.end());
```

and replace it with the following:

erase(begin()+sz, end());

Amend the WP as follows, thus closing issue 23-076:
-- modify both occurrences of the following text in the definition of deque in clause 23.2.1 [lib.deque], as indicated by italics:
typedef *std*::reverse_iterator
-- modify both occurrences of the following text in the definition of vector in clause 23.2.4 [lib.vector], as indicated by italics:
typedef *std*::reverse_iterator
-- modify both occurrences of the following text in the definition of vector<bool> in clause 23.2.5 [lib.vector.bool], as indicated by italics:
typedef *std*::reverse_iterator
-- modify both occurrences of the following text in the definition of basic_string in clause 21.3 [lib.basic.string], as indicated by italics:
typedef *std*::reverse_iterator

Amend the WP by replacing all occurrences of “distance_type” with “difference_type” in clauses 24.1.6 [lib.iterator.tags], 24.2 [lib.iterator.synopsis], 24.3 [lib.iterator.primitives], 24.3.7 [lib.iterator.operations], 24.4.1.1 [lib.reverse.bidir.iter], 25 [lib.algorithms], and 25.1.6 [lib.alg.count], thus closing issue 23-078.