

## Re: STL Vector: Unexpected behavior

---

*Source:* <http://www.tech-archive.net/Archive/VC/microsoft.public.vc.stl/2005-08/msg00120.html>

---

- *From:* Ulrich Eckhardt <[eckhardt@xxxxxxxxxxxxxxxx](mailto:eckhardt@xxxxxxxxxxxxxxxx)>
  - *Date:* Thu, 25 Aug 2005 17:41:16 +0200
- 

bob wrote:

```
> for (localIterator = localVector.begin(); localIterator !=
> localVector.end(); ++localIterator) {
> MyClass* obj_ptr = *localIterator; // dereference to get element
> if(obj_ptr->some_field == true) do_stuff();
> else do_different_stuff();
> // now remove element from list
> localVector.erase(remove(localVector.begin(), localVector.end(),
> *localIterator), localVector.end());
> } // end of for loop
```

Erasing an element from a container always invalidates the iterator used for the deletion, i.e. you can't use it afterwards, not even increment it to the next element. For a vector, you should expect it to invalidate every iterator of that container. Also, removing elements from the middle of a vector is a slow operation, you should use `list<>` instead, then you can use this simple algorithm:

```
iterator it = the_list.begin();
while(it!=the_list.end)
{
if(some_condition(*it))
the_list.erase(it++);
else
++it;
}
```

Other than that, you could move all pointers you want to keep to a new vector and then swap the former one with it.

```
> This results in an error since there is nothing
> in the vector to dereference!
```

Yes, you also omit checking if the pointer is zero in your loop. I'd consider using `boost::shared_ptr<>`, btw.

```
> for (unsigned index = 0; index < localVector.size; ++index)
>
> Whilst that _may_ solve the problem, it won't allow me to _understand_
```

Re: STL Vector: Unexpected behavior

> why the other approach doesn't work.

No, this also doesn't work, because when you erase an element from the middle and then increment 'index' you skipped an element.

Uli

---

• *Follow-Ups:*

- ◆ **Re: STL Vector: Unexpected behavior**  
◇ From: David D
- ◆ **Re: STL Vector: Unexpected behavior**  
◇ From: Stephen Howe
- ◆ **Re: STL Vector: Unexpected behavior**  
◇ From: bob

• *References:*

- ◆ **STL Vector: Unexpected behavior**  
◇ From: bob

- Prev by Date: **STL Vector: Unexpected behavior**
- Next by Date: **Re: STL Vector: Unexpected behavior**
- Previous by thread: **STL Vector: Unexpected behavior**
- Next by thread: **Re: STL Vector: Unexpected behavior**
- Index(es):
  - ◆ **Date**
  - ◆ **Thread**