Subject: Inconsistency between Vector and Index
Posted by Novo on Tue, 23 Jul 2019 01:34:44 GMT
View Forum Message <> Reply to Message

```
 Vector<int> vi = {1, 2, 3};
 Vector<Vector<int>> vvi;
 Index<Vector<int>> ivi;
 vvi.Add(vi); // clones vi.
 ivi.Add(vi); // fails to compile
```

Vector::Add implicitly clones its argument. This is what Index::Add used to do in the past.
New Index will now ask you what exactly you want to do with the argument (pick or clone).
IMHO, this is inconsistent.
IMHO, nothing is supposed to be "secretly" cloned behind the scene.
I would say that current behavior of Vector::Add is incorrect.

Just my two cents.

Subject: Re: Inconsistency between Vector and Index
Posted by mirek on Tue, 23 Jul 2019 07:21:18 GMT
View Forum Message <> Reply to Message

You are right, fixed.

Subject: Re: Inconsistency between Vector and Index
Posted by Novo on Wed, 24 Jul 2019 03:20:54 GMT
View Forum Message <> Reply to Message

mirek wrote on Tue, 23 July 2019 03:21You are right, fixed.
Thanks!

It looks like the issue is still not fixed completely because GrowAdd(const T& x) still calls clone(x).
Also, Array::Add(const T& x) still makes a clone.
Vector::Insert(int i, const T& x) also makes a clone.

IMHO, this issue with clone is a fundamental change of a design. It is, probably, easier to change Index to make clone instead.
Changing semantics of XXX(const T& x) calls will break a lot of code.
It is up to you to decide how it will work in the future.