Subject: Re: Proposal on *Index: access via hash value
Posted by mrjt on Wed, 01 Apr 2009 07:49:11 GMT
View Forum Message <> Reply to Message

Mindtraveller wrote on Tue, 31 March 2009 19:17mrjt, yes, I do have very close situation.

UPD: Finally U++ forced me to switch back to VectorMap. It seems like Index is not designed for such cases (but these cases happen very often IMO). Two big drawbacks:
1) Unable to find element by hash
2) Unable to change element, due to I have only (const T &) in return. But this seems too strict. Changing the object doesn`t necessary change it`s hash value. This approach forces one to declare all members as mutable.
I don't understand these problems.
1) Why don't you just use the hash as the VectorMap key (VectorMap<unsigned, CollectionElement>())? Then you can search by hash directly, although I don't really see the benefit of this for VectorMap.
2) Unless I've misunderstood then this is just wrong. both the Get() and operator[] methods have non-const variants. You can even get direct non-const access to the internal Value vector. If you want to change the Key then you use GetKey and SetKey. This just isn't a problem.

And as I stated previously if you still really want to use Index the code required to add find by hash is minimal (30-40 lines, mostly a copy of the Index class declaration.)