
Subject: Proposal on *Index: access via hash value
Posted by [Mindtraveller](#) on Tue, 31 Mar 2009 11:37:52 GMT
[View Forum Message](#) <> [Reply to Message](#)

Recently I've rewritten a number of classes from VectorMap towards heavily Index usage. But IMO this class lacks access by hash value. To show why it is important I'll give an example of typical situation.

Let's imagine I have a number of CollectionElement class objects. Each object is identified by its unique value. Hash value is derived from this unique id. I do also have a number of another objects which contain information based on CollectionElement-s. To keep this system serializable, I refer to CollectionElements by their unique ids. The problem arises when I try to get requested object by this id. *Index class has no means to reach object by its hash. Once found, I would certainly use FindNext() - so multi-keying isn't problem here.

You may say that there's a *Map class, and it gives me access by id value. But IMO in this situation it is more than needed. Using *Map will make me duplicate keys: internal CollectElement object id is duplicated as the *Map key. It doesn't seem quite well.

May be it will be really useful to add something like

```
bool AIndex::Check(unsigned _hash);
```

```
int AIndex::FindFirst(_hash);
```

```
?
```
