luzr wrote on Sun, 23 September 2007 23:32

Actually, this part is already done

You can search the files now. Invoke help, enter the symbols you want to search and hit a button... (That is why I was wondering why you miss existing feature...

hemmm... yes, I must admit it, I did miss the feature
OTOH, invoke-help-type-word-press-button is far from a confortable CTRL-F1 (or whatever should be) on the word under cursor.
Yes, I'm lazy
And, it works only on built-in help files.

Quote:
Ahh, sorry, I thought we are speaking about U++ Topic++ help files... Yes, this would be a bit more complex issue (but not that complex if somebody would write html and chm RichText parsers)

Those should be already done, somewhere... I'll give it a search next days. Here you really don't need to reinvent the wheel !

Quote:
Actually, I stand behind the design here. In fact, you can use FindIndex for Index too (with the same performance implications).

FindIndex is universal code that works with any U++ container. Why should you repeat it in each one of them?

Also, if you decide that FindIndex is so important that it grants it a method status, what else is important? We do not want to end with hundreds of methods per class, do we?

IMO it is a good practice to limit methods to things that cannot be implemented using container's interface (and, ok, maybe break this rule for convenice in case of very often or very typical methods). But FindIndex is far from frequently used or typical...

I didn't say that you must have hundred methods in container classes (even they'd dont disturb too much, if you don't need them you don't use them....), but I think there should be a sort of 'common set of methods', for all containers.
On my example, you have :

- a FindIndex algorithm, that works for all containers, linear search.
- a Find() method for Index class, that does the same as FindIndex, but in hashed mode.

- NO Find() method for Array and Vector, even no linear search
- a GetIndex() method for Array that is misleading

Uniformity would be :
- a FindIndex algorithm that behaves differently for Array (linear search) than for Index (hashed) and maybe would do a binary search on a SortedArray future class.
--OR--
Find() methods for both containers, behaving differently (this I think would be better...)
--OR--
Both of them (the best, IMHO)

Of course, this is only a small example. You should be able to change containers in code with less effort possible. Now, to change from Index to Array, you must change variable declaration AND change method to locate an element.

Ciao

Max