Subject: GetIndex on Array
Posted by mdelfede on Sat, 22 Sep 2007 16:36:23 GMT
View Forum Message <> Reply to Message

Trying this code :

```
Array<String> a;
a << "pippo";
a << "pluto";
a << "gigi";
int i;
i = a.GetIndex("pluto");
```

i is -1, so no item is found in array.... is it a bug or do I miss something ?
I've seen that using Index and Index.Find() it does work, but Index is not sorteable.

Regards

Max

EDIT : Looking into Array code, I've seen that GetIndex looks only for address of contained element, not for the element itself. So it's useful only for pointers, not for elements for whom the operator== is defined.... A function to locate an element by value is missing.

Subject: Re: GetIndex on Array
Posted by mirek on Sun, 23 Sep 2007 08:36:31 GMT
View Forum Message <> Reply to Message

Yes, actually, "GetIndex" is an idea that looked intereseting at the time Array was first created, but turned out to be not so good idea later... (that is why it is not even documented, but once you introduce a method, it is hard to remove... but in fact, it should be removed).

Anyway, what you need you can get using FindIndex "algorithm" function, like

```
int i = FindIndex(a, "pluto");
```

anyway, be warned, this is linear search.

If you want this fast, use Index (which OTOH has very slow Remove).

Mirek

Subject: Re: GetIndex on Array
Posted by mdelfede on Sun, 23 Sep 2007 13:30:43 GMT
View Forum Message <> Reply to Message

Thank you for the answer

I'd already found the FindIndex algorithm, it works good for me.
The GetIndex of array class is really a danger, I lost 2 ours before understanding it works only on pointer value.
It's right, it's not documented, but I was thinking was a doc problem

I know that FindIndex does linear search, but I don't need speed (for now, at least), I'm making a quick and dirty console app to package OpenCascade libraries in TheIde format, separating packages and so on.... as the original version as an all-in-one include dir with 15000 files inside, it's hard to do it by hand.

I think it would not be a bad idea to have a 'real' FindIndex function in array class, it's often needed and it would be nice to have it without need of external algorithm.

A little note (again) on help system : I find it really unusable. That's the only thing I don't like on TheIde. I'm resorting with some fiferox windows opened by side, but I'm really missing a word search also on custom help files from the ide, as it has been done on codeblocks, for example.

Ciao

Massimo

Subject: Re: GetIndex on Array
Posted by mirek on Sun, 23 Sep 2007 16:57:50 GMT
View Forum Message <> Reply to Message

mdelfede wrote on Sun, 23 September 2007 09:30
A little note (again) on help system : I find it really unusable. That's the only thing I don't like on TheIde. I'm resorting with some fiferox windows opened by side, but I'm really missing a word search also on custom help files from the ide, as it has been done on codeblocks, for example.

Yep, there is a lot to be improved.

However, the basic search in help files is there ?!

Mirek

Subject: Re: GetIndex on Array
Posted by mdelfede on Sun, 23 Sep 2007 17:33:22 GMT
View Forum Message <> Reply to Message

luzr wrote on Sun, 23 September 2007 18:57
Yep, there is a lot to be improved.

However, the basic search in help files is there ?!

Uhmmmmm... where ? If I've, for example, a line like that :

```
Line = f.GetLine();
```

with f of type FileIn, I can get something with Assist++, but only the "completion", or also the source code of function, but no help. Of course, I can search for it in documentation, but it's a long way.
I see that the search part works good, as it finds all classes with GetLine (one, in this case, but more in other cases), and I think it's the good of your CParser.
What is missing is an indexed help, so with another key you can get help on item instead of the code definition of it.

I like the embedding of help system in the devel version of ide (but for now I use the official release because of problems with the devel one...); you should add the ability of embed external help files (html would be the best, as most are in this format), and a way to index them.
Codeblocks did it in a good way and, if I don't be wrong, KDevelop did it too.

Don't take it as critics, please... if I'm using UPP and TheIde and NOT CodeBlocks nor KDevelop there should be some good reasons !

Subject: Re: GetIndex on Array
Posted by mirek on Sun, 23 Sep 2007 18:00:13 GMT
View Forum Message <> Reply to Message

mdelfede wrote on Sun, 23 September 2007 13:33luzr wrote on Sun, 23 September 2007 18:57
Yep, there is a lot to be improved.

However, the basic search in help files is there ?!

Uhmmmmm... where ? If I've, for example, a line like that :

```
Line = f.GetLine();
```

with f of type FileIn, I can get something with Assist++, but only the "completion", or also the

source code of function, but no help. Of course, I can search for it in documentation, but it's a long way.
I see that the search part works good, as it finds all classes with GetLine (one, in this case, but more in other cases), and I think it's the good of your CParser.
What is missing is an indexed help, so with another key you can get help on item instead of the code definition of it.

Well, so what we need is a key that invokes help, fills in the current ID and performs the search, right?

That should be trivial... sorry that it is not already there

Quote:
Don't take it as critics, please... if I'm using UPP and TheIde and NOT CodeBlocks nor KDevelop there should be some good reasons !

What is wrong with critics?

I know that there still is a lot to be desired...

Mirek

Subject: Re: GetIndex on Array
Posted by mr_ped on Sun, 23 Sep 2007 19:23:17 GMT
View Forum Message <> Reply to Message

luzr wrote on Sun, 23 September 2007 20:00Well, so what we need is a key that invokes help, fills in the current ID and performs the search, right?

It was called "Context Help Ctrl+F1" back in times of Borland Turbo Pascal 5.0. Was that 1993 or sooner?
With the power of Assist++ I think it can work even on variable identifiers (bringing help for variable type if it is described in help), etc..

And Ctrl+F1 will again collide with KDE default desktop switching keys. You can't please everyone.

Subject: Re: GetIndex on Array
Posted by mirek on Sun, 23 Sep 2007 19:36:31 GMT
View Forum Message <> Reply to Message

mr_ped wrote on Sun, 23 September 2007 15:23luzr wrote on Sun, 23 September 2007 20:00Well, so what we need is a key that invokes help, fills in the current ID and performs the search, right?

It was called "Context Help Ctrl+F1" back in times of Borland Turbo Pascal 5.0. Was that 1993 or sooner?
With the power of Assist++ I think it can work even on variable identifiers (bringing help for variable type if it is described in help), etc..

Well, I would rather make it a brute force approach for now - just do the search of identifier in all docs... (will reveal tutorial examples too!).

Mirek

Subject: Re: GetIndex on Array
Posted by mdelfede on Sun, 23 Sep 2007 21:03:01 GMT
View Forum Message <> Reply to Message

luzr wrote on Sun, 23 September 2007 20:00

Well, so what we need is a key that invokes help, fills in the current ID and performs the search, right?

That should be trivial... sorry that it is not already there

Ehehehehehe.... maybe it's not so trivial, if you want to manage multiple (external) help files and index all of them in a smart way.... But context help is a must, I think.
Nor is trivial the html widget you'd need in order to make all that stuff.
Even worse, if you want to directly support chm files.... too many help files are in that format.

Back to container classes, another thing that I miss is some sort of 'uniform behaviour' between various containers. I know that Index is the best for fast searches and Array is better for insertions/deletions, but having for them the same set of member functions would make life easier. Why a Find() member on Index and not on Array ? Ok, on Index is faster, but what if I need a sortable container (like Array) and I need also to search inside it ? Yes, there's the FindIndex algorithm, but it's another function usage to remember.
You could have Array with linear search, Index with hashed search and SortedArray with binary search, all of them with a member function Find().

Well, enough for today !
As I said, I like TheIde, and I think it needs only some minor 'polishing' to be the best multiplatform ide out there.

Ciao

Max

Subject: Re: GetIndex on Array
Posted by mirek on Sun, 23 Sep 2007 21:32:19 GMT
View Forum Message <> Reply to Message

mdelfede wrote on Sun, 23 September 2007 17:03luzr wrote on Sun, 23 September 2007 20:00

Well, so what we need is a key that invokes help, fills in the current ID and performs the search, right?

That should be trivial... sorry that it is not already there

Ehehehehehe.... maybe it's not so trivial, if you want to manage multiple (external) help files and index all of them in a smart way.... But context help is a must, I think.

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...

Quote:
Nor is trivial the html widget you'd need in order to make all that stuff.
Even worse, if you want to directly support chm files.... too many help files are in that format.

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)

Quote:
Back to container classes, another thing that I miss is some sort of 'uniform behaviour' between various containers. I know that Index is the best for fast searches and Array is better for insertions/deletions, but having for them the same set of member functions would make life easier. Why a Find() member on Index and not on Array ? Ok, on Index is faster, but what if I need a sortable container (like Array) and I need also to search inside it ? Yes, there's the FindIndex algorithm, but it's another function usage to remember.

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...

Mirek

Subject: Re: GetIndex on Array
Posted by mdelfede on Sun, 23 Sep 2007 22:13:52 GMT
View Forum Message <> Reply to Message

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