Subject: Are there any plans supporting Vector/Array/Map::operator[] during debug?
Posted by Mindtraveller on Fri, 08 May 2009 10:43:50 GMT
View Forum Message <> Reply to Message

A understand of course that it is very untrivial and must be very hard to code, but viewing something like myVector[i] would be very handy.

Subject: Re: Are there any plans supporting Vector/Array/Map::operator[] during debug?
Posted by unodgs on Fri, 08 May 2009 12:09:40 GMT
View Forum Message <> Reply to Message

Mindtraveller wrote on Fri, 08 May 2009 06:43A understand of course that it is very untrivial and must be very hard to code, but viewing something like myVector[i] would be very handy.
Yes, I miss that too

Subject: Re: Are there any plans supporting Vector/Array/Map::operator[] during debug?
Posted by mirek on Sat, 09 May 2009 15:16:05 GMT
View Forum Message <> Reply to Message

Well, maybe we could use dirty trick...

Mirek

Subject: Re: Are there any plans supporting Vector/Array/Map::operator[] during debug?
Posted by Mindtraveller on Sat, 09 May 2009 20:09:49 GMT
View Forum Message <> Reply to Message

As far as I know C++ Builder authors solved this problem by actual execution of child code in child process' address space (set IP to actual function, execute, and get the result, restore IP).
They realized that executing such a code could (in some cases) lead to changing internal program state. But in most cases calls like myvector[i] didn`t change anything important. To make their solution more flexible they even had Configuration checkbox like "[x] Allow side-effects" which was checked by default.
The rule is: when I type something into watches, I really want to know what it is and I implicitly agree with any side effects.
It seems good solution to me: in most cases it is more important to have all visible during debug, and it make no problem. But if it is crucial, one may disable this any time he wants.

Subject: Re: Are there any plans supporting Vector/Array/Map::operator[] during

debug?
Posted by mr_ped on Sun, 10 May 2009 09:49:19 GMT
View Forum Message <> Reply to Message

If the [] operator is const, it should not change anything internally, i.e. if the whole chain of such needed functions are const, you are quite safe to use such hack.

Although I have a *feeling* (didn't even think about it) this may be hard to write, hard to maintain and hard to make crossplatform.

Maybe Mirek will come with better (edit: I meant more like "simpler") hack, after all as long as we are speaking about built in NTL classes, nobody knows them better then him.