Subject: Vector<int>::At does not return a default constructed value
Posted by cbpporter on Mon, 15 Dec 2014 13:00:24 GMT
View Forum Message <> Reply to Message

I understand why this would happen with built in types, but is there no fix for this?

Subject: Re: Vector<int>::At does not return a default constructed value
Posted by mirek on Tue, 16 Dec 2014 06:48:41 GMT
View Forum Message <> Reply to Message

cbpporter wrote on Mon, 15 December 2014 14:00I understand why this would happen with built in types, but is there no fix for this?

Here I do not understand what you mean by "fix"?

As it is now, if you need new elements added to Vector<int> to be initialized to some specific value, e.g. zero, you can specify it as second parameter of .At.

Now by fix you can also mean that int() 'constructor' should be used and initialize value to zero, like std::vector does. Here I am not so sure this is such a good thing; but I am open to debate...

Mirek

Subject: Re: Vector<int>::At does not return a default constructed value
Posted by cbpporter on Tue, 16 Dec 2014 11:54:20 GMT
View Forum Message <> Reply to Message

mirek wrote on Tue, 16 December 2014 08:48

Now by fix you can also mean that int() 'constructor' should be used and initialize value to zero, like std::vector does. Here I am not so sure this is such a good thing; but I am open to debate...

Mirek

Yes, that's what I meant. I know this is a touchy subject maybe with no real best answer and I was wondering what your stance is on this.

I ask for two reasons: one I needed At and forgot about the two parameter version so I ended up writing more complex code than needed to handle this (fixed now with two param version) and I'm also designing my own mini-library where I'm hitting the same problems with the container classes.

Subject: Re: Vector<int>::At does not return a default constructed value
Posted by mirek on Tue, 16 Dec 2014 14:16:47 GMT
View Forum Message <> Reply to Message

Well, the disadvantage of such "involuntary" initialization is fact that in some cases, compiler is not able to optimize out the zero assignment, thus resulting in negligibly lower performance. IMO, same reasoning is true for local variables...

And then, estetically, I really dislike the anomaly that

int x;

and

int x = int();

are different things...

Mirek