
Subject: Re: Using Vector::At doesn't initialize implicit types

Posted by [mr_ped](#) on Tue, 29 Jul 2008 09:13:58 GMT

[View Forum Message](#) <> [Reply to Message](#)

cbpporter wrote on Tue, 29 July 2008 11:10 Yes, using that At works if you want to initialize it there.

I was talking about the way user defined classes get their default constructor called, while implicit types get left initialized.

It should be more uniform.

And since user types do get their constructor called, I don't think there is any performance difference between what I'm proposing and the current situation.

No, it does fix your problem, try it.

```
...  
Vector<int> v;  
v.At(0, 0) = 7000;  
v.At(2, 0) = 7000;
```

```
...  
LOG:  
v[0] = 7000  
v[1] = 0  
v[2] = 7000
```

I'm using basic types a lot, and with vectors over 1mil in size, so unwanted initialization would slow me down considerably.

Actually I think calling the user defined constructor may be inconsistent behavior, and maybe it should not happen in your example.

edit:

And the "int" nature of vector size is starting to scare me, I would rather prefer full 32b size_t (although I understand the pick behavior would interfere much more in such scheme without additional flag (increasing the memory footprint of Vector more)) :/ This new HW in recent years makes 2+GB of data to look like an easy chew, yet SW like UPP::Vector will give you some headache in such case, and you will end up probably with custom containers anyway. On 64b system the int size of Vector will become problem in future.
