
Subject: Using Vector::At doesn't initialize implicit types
Posted by [cbpporter](#) on Tue, 29 Jul 2008 08:51:02 GMT
[View Forum Message](#) <> [Reply to Message](#)

If I use At or SetLength, Vecor will call the default constructor for the new items created which are not manually initialized for user classes, but will leave implicit types uninitialized.

Example:

```
#include <Core/Core.h>

using namespace Upp;

struct Foo: public Moveable<Foo>
{
    int x;

    Foo()      { x = 0; }
    Foo(int xx) { x = xx; }
};

CONSOLE_APP_MAIN
{
    Vector<int> v;
    v.At(0) = 7000;
    v.At(2) = 7000;

    DUMP(v[0]);
    DUMP(v[1]);
    DUMP(v[2]);

    Vector<Foo> w;
    w.At(0) = Foo(7000);
    w.At(2) = Foo(7000);

    DUMP(w[0].x);
    DUMP(w[1].x);
    DUMP(w[2].x);
}
```

I think we should change this. It can lead to problems and nasty crashes if you have a Vector<Foo>, where Foo is a value type class, and for some reason you need to start using Vector<Foo*>. If you Vector is not initialized in order, which is not a requirement for this class, you will most likely get memory corruption when you try to access the gaps and you cant test for NULL either.
