

---

Subject: VectorMap.Get

Posted by [deep](#) on Tue, 13 Oct 2015 12:31:06 GMT

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

---

Hi,

When I want to get the content of VecotorMap second time I am getting Assert Failed error.

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

using namespace Upp;

CONSOLE_APP_MAIN
{
    Vector<int> v1;
    VectorMap<String,Vector<int>> Vm ;

    v1.Clear();
    v1.Add(0);
    v1.Add(3);
    Vm.Add("key1",v1);

    v1.Clear();
    v1.Add(10);
    v1.Add(31);
    v1.Add(41);
    Vm.Add("key2",v1);

    Vector<int> v2 ;
    v2 = Vm.Get("key1");
    DUMP(v2);

    v2.Clear();
    v2 = Vm.Get("key2");
    DUMP(v2);

    v2.Clear();
    v2 = Vm.Get("key1");
    DUMP(v2);
}
```

Log content

```
v2 = [0, 3]
```

```
v2 = [10, 31, 41]
v2 = ***** ASSERT FAILED: Assertion failed in
c:\devtools\upp_git\uppsrc\core\Vcont.h, line 34
Broken rval_ semantics
```

It is working whith VectorMap<String,String>

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

using namespace Upp;

CONSOLE_APP_MAIN
{
  VectorMap<String,String> v1;

  v1.Add("key1","val1");
  v1.Add("key2","val2");
  v1.Add("key3","val3");

  String s;

  s.Clear();
  s= v1.Get("key1");
  DUMP(s);

  s.Clear();
  s= v1.Get("key2");
  DUMP(s);

  s.Clear();
  s= v1.Get("key3");
  DUMP(s);

  s.Clear();
  s= v1.Get("key1");
  DUMP(s);
}
```

Log content

```
s = val1
s = val2
s = val3
s = val1
```



Subject: Re: VectorMap.Get

Posted by [Klugier](#) on Tue, 13 Oct 2015 14:38:05 GMT

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

---

Hello Deep,

Try to clone your vector. I think in your case there is move operation for some reason. Maybe someone that knows better that mechanism can describe it for you. On the other hand I have got compilation error of your example in the latest upp. Code that should work:

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

using namespace Upp;

CONSOLE_APP_MAIN
{
    Vector<int> v1;
    VectorMap<String, Vector<int>> Vm;

    v1.Clear();
    v1.Add(0);
    v1.Add(3);
    Vm.Add("key1", v1); // <- Vector was copied

    v1.Clear();
    v1.Add(10);
    v1.Add(31);
    v1.Add(41);
    Vm.Add("key2", v1); // <- Vector was copied

    Vector<int> v2;
    v2 = clone(Vm.Get("key1")); // <- Without clone, probably move operation
    DUMP(v2);

    v2.Clear();
    v2 = clone(Vm.Get("key2"));
    DUMP(v2);

    v2.Clear();
    v2 = clone(Vm.Get("key1"));
    DUMP(v2);
}
```

Here is my log values:

```
v2 = [0, 3]
v2 = [10, 31, 41]
v2 = [0, 3]
```

-----  
P.S.

Maybe you should put space after comma. In my opinion declaration like: `VectorMap<String, Vector<int>>` is more readable than `VectorMap<String,Vector<int>>`.

Sincerely,  
Klugier

---

Subject: Re: VectorMap.Get

Posted by [mirek](#) on Thu, 15 Oct 2015 09:02:37 GMT

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

---

[quote title=Klugier wrote on Tue, 13 October 2015 16:38]Hello Deep,

Try to clone your vector. I think in your case there is move operation for some reason. Maybe someone that knows better that mechanism can describe it for you. [code]

Yes, it is 'pick' (U++ variant of move).

In C++11 mode, the code would not even compile. Unfortunately, before c++11, there are no means to detect these errors at compile time.

Note that the optimal variant to using 'clone' is to use const reference (unless you need to change the element you are getting).

```
const Vector<int>& v2 = Vm.Get("key1");
```

Mirek

---

Subject: Re: VectorMap.Get

Posted by [deep](#) on Thu, 15 Oct 2015 10:01:48 GMT

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

---

Hi Mirek and Klugier,

Thanks for response.

Both working.

Will use

```
const Vector<int>& v2 = Vm.Get("key1");
```

Need to only use the returned values.

---