
Subject: Re: VectorMap inside VectorMap
Posted by [White_Owl](#) on Tue, 03 Feb 2009 16:58:10 GMT
[View Forum Message](#) <> [Reply to Message](#)

Ok, here is the code which does exactly what I want:

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

```
using namespace Upp;
```

```
class mycontainer: Moveable<mycontainer> {
public:
    VectorMap<String, String> data;
};

VectorMap<String, mycontainer> vm;

void define_word(const String& outer_key, const String& inner_key, const String& value) {
    mycontainer *ca = vm.FindPtr(outer_key);
    if(!ca)
        ca = &(vm.Add(outer_key));

    if(ca->data.Find(inner_key)>=0)
        ca->data.RemoveKey(inner_key);
    ca->data.Add(inner_key, value);
}
```

```
CONSOLE_APP_MAIN
```

```
{
    define_word("First", "first-first", "aa");
    puts(vm.Get("First").data.Get("first-first"));
    puts("Sizeof(vm)=" + FormatInt(vm.GetCount()));
    puts("Sizeof(vm{First}.data)=" + FormatInt(vm.Get("First").data.GetCount()));

    define_word("First", "first-first", "bb");
    puts(vm.Get("First").data.Get("first-first"));
    puts("Sizeof(vm)=" + FormatInt(vm.GetCount()));
    puts("Sizeof(vm{First}.data)=" + FormatInt(vm.Get("First").data.GetCount()));
}
```

I think GetAdd()/FindPut and other "double" methods already have all this checking and removing old values. But I can not find which one of them actually does it without invalidating objects inside the container.
