
Subject: Avoid Copy when adding a Tuple to a VectorMap

Posted by [Xemuth](#) on Sun, 06 Jun 2021 01:39:43 GMT

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

Hello,

I'm working with a `VectorMap<int, Tuple<int, A>>`

class `A` don't provide a copy constructor, only a Move constructor.

I would like to add a new Tuple to my vector map by moving a fresh class `A`.

But I don't succeed... Can someone help me ? (I would like to avoid creation of a class to replace tuple in this specific case)

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

```
using namespace Upp;
```

```
class A{
public:
    A(int e) : d_e(e){}
    A(A&& a){d_e = a.d_e;}
    int d_e;
};
```

```
CONSOLE_APP_MAIN
```

```
{
    VectorMap<int, Tuple<int, A>> myVector;
    myVector(1, pick(Tuple<int,A>(1, 5)));
    Cout() << myVector.Get(1).b.d_e << EOL;
    //Error : call to implicitly deleted copy constructor of 'A'
}
```

Also, why there is no `Create(Args...)` function in `VectorMap` ? (like the one in `ArrayMap`)
