
Subject: [SOLVED] Vector Bug
Posted by [Xemuth](#) on Tue, 22 Oct 2019 11:47:40 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hello community,

let's assume this code :
#include <Core/Core.h>

using namespace Upp;

```
CONSOLE_APP_MAIN
{
    Vector<String> vector1;

    String& r = vector1.Create("hello");
    Cout() << r << EOL;

    String& r2 = vector1.Create("hello2");
    Cout() << r2 << EOL;

    r = "test";//Crashing here
    Cout() << r << EOL;
}
```

my code crash at r="test" statement.
Is it a bug ?!

this kind of thing works :
Array<String> vector1;

```
String& r = vector1.Create<String>("hello");
Cout() << r << EOL;
```

```
String& r2 = vector1.Create<String>("hello2");
Cout() << r2 << EOL;
```

```
r = "test";
Cout() << r << EOL;
```

Thanks in advance
Best regard.

Subject: Re: Vector Bug

Posted by [cbpporter](#) on Tue, 22 Oct 2019 12:06:47 GMT

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

Vector, when it reallocates memory, it invalidates references.

[https://www.ultimatepp.org/srcdoc\\$Core\\$Caveats_en-us.html](https://www.ultimatepp.org/srcdoc$Core$Caveats_en-us.html)

I don't think there is a workaround for this and you are not supposed to long term store references gotten from vector, just store them for a short amount of processing, until the next operation on Vector.

I believe if you need to store references the correct way is to use Array.

Subject: Re: Vector Bug

Posted by [Xemuth](#) on Tue, 22 Oct 2019 12:52:10 GMT

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

cbpporter wrote on Tue, 22 October 2019 14:06Vector, when it reallocates memory, it invalidates references.

[https://www.ultimatepp.org/srcdoc\\$Core\\$Caveats_en-us.html](https://www.ultimatepp.org/srcdoc$Core$Caveats_en-us.html)

I don't think there is a workaround for this and you are not supposed to long term store references gotten from vector, just store them for a short amount of processing, until the next operation on Vector.

I believe if you need to store references the correct way is to use Array.

Ahhhhhh ! seems logique thanks
