

---

Subject: Re: About Nuller and Null

Posted by [mirek](#) on Sat, 10 Oct 2020 23:39:01 GMT

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

---

Tom1 wrote on Sat, 10 October 2020 20:25Hi Mirek,

Thanks for looking into this. I really have trouble and feel insecure about returning Null references. The access to Array and Vector containers comes as references. So, when I create a function returning those references, I need to be able to return Null if the container does not have a suitable object to return for a request.

However, returning a Null reference is not trivial. And possibly also forbidden in C++. Then, I looked at using pointers instead and found that C++ references have the following limitation:

"There shall be no references to references, no arrays of references, and no pointers to references. " (ISO C++)

Finally (after quite a few hours) I came up with the following solution: Using: " return (A&)Null; " to return a Null reference. How dangerous is this? (I also added the check: " this==&(classname&)Null " to IsNullInstance() in order to cover this case.

In contrast to the previous code the following compiles with CLANG too and seems to work as expected:

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

```
using namespace Upp;
```

```
#define NULLSUPPORT(classname, variable)\
classname(const Nuller&) { variable=NULL; }\
void SetNull() { variable=NULL; }\
bool IsNullInstance() const { return this==&(classname&)Null || IsNull(variable); }
```

```
class A{\
public:\
    int a;\
    int b;
```

```
    NULLSUPPORT(A,a)
```

```
    void Clear(){ a=b=0; }
```

```
    A(){\
        a=1;\
        b=2;\
    }
```

```
    void Serialize(Stream &s){
```

```
s % a % b;  
}
```

```
String ToString() const { return IsNullInstance() ? String("Null") : String("A[") << a << ", " << b <<  
"]"; }  
};
```

// Testing:

```
Array<A> av;
```

```
A& GetA1(int x){  
    if((x<0)||((x>=av.GetCount()))) return (A&)Null;  
    return av[x];  
}
```

```
CONSOLE_APP_MAIN{  
    av.Add().a=1;  
    av.Add().a=2;  
    av.Add().a=3;  
    av.Add().a=4;  
  
    for(int i=-1;i<6;i++){ A &a=GetA1(i); Cout() << a << "\n"; }  
    return;  
}
```

But is this safe? If not, is there a decent way to do it?

Best regards,

Tom

I am totally confused what are you trying to achieve here...

Both Null and Nuller are never supposed to be used outside of "assigning Null syntax sugar" context.

I think you might be overthinking something here.

Mirek

Mirek