

Re: C++ Compiler behavior regarding struct constructors

Source:

<http://www.tech-archive.net/Archive/DotNet/microsoft.public.dotnet.languages.vc/2004-12/0399.html>

From: Vladimir Nesterovsky (*vladimir_at_nesterovsky-bros.com*)

Date: 12/19/04

Date: Sun, 19 Dec 2004 18:33:33 +0200

C++ guarantees zero initialization for static variables of primitive type, or of primitive structure, or of array of primitive types (structures) at program start up.

Some heaps provide guarantee for zero allocated memory.

Managed stack also provides zero initialized stack frame guarantee.

In all other cases developer should take care of memory initialization.

I believe it's an extension that:

```
MyStruct m_MyStruct;
```

and

```
MyStruct m_MyStruct = MyStruct();
```

generate different code.

--

Vladimir Nesterovsky

e-mail: vladimirn@multiconn.com

> I just notice some strange behaviors on the MS C++ compiler regarding struct

> default constructor, see the example bellow:

> struct MyStruct

> {

> int a[5];

> };

> class MyClass

> {

> public:

> MyClass();

> ~MyClass();

> protected:

> MyStruct m_MyStruct;

> };

> // #1:

> MyClass::MyClass() //The default ctor for the class does not init the struct

> with NULL(maybe there is no default ctor for struct yet?)

> {

microsoft.public.dotnet.languages.vc: Re: C++ Compiler behavior regarding struct constructors

```
> }
> // #2 if I define the class ctor like this:
> MyClass::MyClass() : m_MyStruct() // The default ctor for the class does
init
> the struct with NULL (call m_MyStruct() default ctor, now there is one!)
> {
> }
> // #3 or like this:
> MyClass::MyClass()
> {
>     m_MyStruct = MyStruct(); // does init the struct with NULL using a
local
> stack var init with NULL
> }
> // Now if I define the struct like this:
> struct MyStruct
> {
>     MyStruct()
>     {
>         int i;
>         for(i=0; i<5; i++)
>             a[i] = NULL;
>     }
>     int a[5];
> };
> // #1 will init the struct with NULL because uses the defined struct ctor
but
> generate almost the same bin (asm) code as case #2 and #3 (the init stack
> var) without the struct ctor defined!
> I guess I don't understand when the compiler knows about the struct
default
> constructor and when does not (and expect for one defined).
```