

# Re: How to Define a global const class variable in MFC?

---

*Source:* <http://www.tech-archive.net/Archive/VC/microsoft.public.vc.mfc/2008-09/msg00001.html>

---

- *From:* Joseph M. Newcomer <[newcomer@xxxxxxxxxxxxx](mailto:newcomer@xxxxxxxxxxxxx)>
  - *Date:* Sun, 31 Aug 2008 10:34:48 -0400
- 

See below..

On Sat, 30 Aug 2008 20:30:01 -0700, Electronic75 <[Electronic75@xxxxxxxxxxxxxxxxxxxxxxxxxxxxx](mailto:Electronic75@xxxxxxxxxxxxxxxxxxxxxxxxxxxxx)> wrote:

Thanks a lot Joe and doug, your posts were informative for me. one thing that I didn't understand thoroughly about header files invc++, Is in C it is supposed that all variables to be declared in a header file(declared not defined) and then we define the variable in source file. but in vc++ it seems that we have to both declare and define variables in header file. for global variables we have to declare it in header file with extern keyword and define it in source file that is perfectly consistent with c style but class member variables are both declared and defined in the header file, why is that?

\*\*\*\*

No. You are confusing declaration and definition.

If I write

```
class A {
public:
int x; // 1
static int y; // 2
static const int z; // 3
static const int w = 3; // 4
B stuff[w]; // 5
};
```

then I have declared:

1. A class instance variable x. Every instance of the class has a different address for the value x. Every reference to x is computed relative to the current this-> pointer or relative to the class instance pointer.

I can only reference this variable via a class reference, e.g.,

```
A t;
```

```
t.x;
```

```
or
```

```
void SomeFunction(A * b)
```

```
{
```

## Re: How to Define a global const class variable in MFC?

```
... b->x ...  
}  
or  
void A:OtherFunction()  
{  
... x ...  
}
```

2. A static class variable `y`. I must later, in some module, write  
`int A:y;`

Every reference to this variable uses exactly the same address, e.g.,

```
A() { y++; }
```

```
~A() { y--; }
```

counts the number of instances of `A` [this is not thread safe, but let's deal with one complication at time]. If at any time I want to see the number of instances, I could write, anywhere, something that looked like this:

```
TRACE(_T("There are %d instances of A\n"), A::y);
```

3. A static class constant `z`. I am required to define it later, as  
`const int A::z = 3;`

but I cannot use that value in my class declaration. I can use this public variable in any context I want, just by naming `A::z`.

4. A static class constant `w`. This both declares and defines it, and allows me to use it later in the class declaration (although why I would declare an array of fixed size in C++ escapes me...). I can use this public variable in any context I want, just by naming `A::w`.

5. An array of compile-time fixed size where the number of elements is a named constant that is scoped to the class namespace.

I could write

```
void A::AnyFunction()  
{  
for(int i = 0; i < w; i++)  
Process(stuff[i]);  
}
```

or even

```
void RandomClass::Anything(A * data)  
{  
for(int i = 0; i < A::w; i++)  
Process(data->stuff[i]);  
}
```

Note that except for case (4), it behaves as you expect.

The use of `extern` is used when you have a non-method declaration of a variable that is not

## Re: How to Define a global const class variable in MFC?

part of a class.  
extern A SomeData;  
which, just like C, requires that eventually some module declare  
A SomeData;  
\*\*\*\*

This is a completely scare concept. What is a static char \* variable doing here?  
Generally, as soon as you write 'char', you are writing horribly obsolete code. A fixed size array of anything is dangerous. Note that this assumes that whatever this is, it is known to all instances.  
\*\*\*\*

joe you said "a fixed size array of anything is dangerous" do you mean only pointers to arrays or arrays themselves because I used to think otherwise for defining arrays. I thought if I do not specify size of an array and define for example UCHAR c[], it will cause compiler error and danger of overwriting memory.

\*\*\*\*

It depends. If you write UCHAR c[]; (something I would never do except as the last element of a variable-length struct) it is probably syntactically illegal. If you want to define an array of unknown size, you would either define a UCHAR \* c, and allocate storage for it (primitive, 1975 C style), which would then require you to free it in the destructor, or using modern C++ methodologies you would declare CArray<UCHAR> c;  
or  
std::vector<UCHAR> c;  
or  
CByteArray c;  
and let the compiler handle all the deallocation. Since there is rarely any need to declare a fixed-size array of anything in modern C++ programming style.  
\*\*\*\*

about char being old style, I didn't understand why. it is one of standard keywords of c why should it be old style? I also program microcontrollers in plain C and there because limitation of memory using variables with smaller size are more desirable. why it is old style because computer has a lot of RAM and i could define an int even if I didn't need that?

\*\*\*\*

So what? Why does being a "standard keyword of c" make it justifiable? Modern practice says that Unicode is the way to go, and at the very least, Unicode-aware. Therefore, the use of 8-bit char data type is obsolete in modern practice. What you do in a microcontroller and what you do writing a Windows application are completely different.

It has nothing to do with the amount of RAM; it has to do with creating apps that can be

Re: How to Define a global const class variable in MFC?

## Re: How to Define a global const class variable in MFC?

localized to any language. RAM is pretty much an irrelevant consideration compared to the cost of retrofitting Unicode later. So get out of the habit of thinking 'char' is a useful data type in writing WIndows apps. It is used in incredibly rare and exotic situations, most typically, talking to embedded controllers and handling 8-bit data streams over network connections (which are often encoded 8-bit data in UTF-8). It is not a good habit to use in writing modern code.

\*\*\*\*

If GetCollectionName is returning the m\_sCollectionName, then m\_sCollectionName should probably be protected.

You don't need (void), () says the same thing.

\*\*\*\*

you are absolutely write, thanks

Not knowing what is in it makes it hard to guess what its relationship to anything else might be.

\*\*\*\*

Second You can see in this line  
CArray <CCommandCollection> m\_aCommandCollect;  
I have defined an array of the same type of class. Compiler didn't get me error for that(if I remove global definition of CCommandCollection variable program compiles smoothly) but I suspect if it can cause trouble.

\*\*\*\*

Why? It is perfectly valid syntax and has perfectly valid semantics which are well-defined. Why would it cause trouble?

\*\*\*\*

thanks, I learned that.

Since you did not include the contents of the other header file, there's no way to tell.

You did not show what is there, or give the actual compiler error messages.

I can write the whole header files and source files but I thought it will be very big and boring to readers. I will try to change some of my code in light

## Re: How to Define a global const class variable in MFC?

of what I've learn in this thread and if I still couldn't define a global class variable I will post all of the codes.  
thank you for your kindness and being helpful.  
best wishes,

\*\*\*\*

Key here is that you **HAVE** to show the error messages **AS ISSUED BY THE COMPILER**, and show us exactly what lines they occur on (some people show a message that says "error on line 282" but only show us a dozen lines, and we have to guess which one is line 282). In addition, you must show all declarations of variables involved in the line that has the error, which means you must show us the header files (you often do not need to show all the details of the class contents, but we need enough to be able to understand what the declarations are so we can figure out why the error message, that **SPECIFIC** error message, came out). You need to show several lines before the error message as well (it is surprising how many people make a syntax error on the previous line which isn't caught until the next line, but they show us the line in question with no context)

joe

\*\*\*\*

Joseph M. Newcomer [MVP]  
email: newcomer@xxxxxxxxxxxxx  
Web: <http://www.flounder.com>  
MVP Tips: [http://www.flounder.com/mvp\\_tips.htm](http://www.flounder.com/mvp_tips.htm)

.