

## Re: \_\_VA\_ARGS\_\_ and the DDK compiler

---

*Source:*

<http://www.tech-archive.net/Archive/Development/microsoft.public.development.device.drivers/2005-10/msg00123.html>

---

- *From:* "Norman Diamond" <[ndiamond@xxxxxxxxxxxxxxxxxxxx](mailto:ndiamond@xxxxxxxxxxxxxxxxxxxx)>
  - *Date:* Thu, 6 Oct 2005 09:16:38 +0900
- 

No, his questions are related more to the difference between MS and the ISO standard. If his questions concerned a Visual Studio product then microsoft.public.vc.language would be the place for them, but it looked to me like he might be using a compiler that came in a DDK.

(For Visual Studio, the MSDN feedback site might also be a suitable place, but one particularly obvious nonconformance to the ISO C++ and C standards was already marked "postponed".)

"Pavel A." <[pavel\\_a@xxxxxxxxxxxxxxxxxxxx](mailto:pavel_a@xxxxxxxxxxxxxxxxxxxx)> wrote in message <news:uzteZPayFHA.3696@xxxxxxxxxxxxxxxxxxxxxxxxxxxx>

Hi,  
Since your questions are related more to the difference between MS and GNU compilers than to driver development, try to ask in microsoft.public.vc.language

Regarding the lack of vararg macros: see how KdPrint macro is defined in the DDK.  
The usage has double braces like KdPrint(("format", arg, arg)) - it looks weird but works.  
Another way to deal with debug prints is WMI trace and WPP - it allows printf like statements.

Regards,  
--PA

"Laurents C. R. Meyer"  
<[LaurentsCRMeyer@xxxxxxxxxxxxxxxxxxxxxxxxxxxx](mailto:LaurentsCRMeyer@xxxxxxxxxxxxxxxxxxxxxxxxxxxx)> wrote in message <news:EDC3673C-9577-427D-A9DC-A29666975FCD@xxxxxxxxxxxxxxxxxxxx>

I see. Well in this case there will be no way to accomplish this, because I wanted to do something like the following:

Re: \_\_VA\_ARGS\_\_ and the DDK compiler

```
#define OUT( format, ... ) DbgPrint( __FILE__  
"%4.4d) - " __FUNCTION__ ": "  
format, __LINE__, __VA_ARGS__ )
```

My attempts I did before posting to use as `__inline` or `__forceinline` declared functions with `__LINE__` etc. macros inside the inlined function scope resulted in very odd Outputs - negativ file numbers and other stuff.

So I guess it is impossible.

Another question on my mind is, why does the

```
#line LineNum "FileName"
```

directive uses an offset of 2 for LineNum?  
Since I could not figure out a way to deactivate what would be the (I think) /FC switch in VC++ do reduce the `__FILE__` output from `path\filename` to `filename` (which is the default in VC++), I am using the `#line` directive to set the filename manually. Consulting the MSDN I expected the following to work properly (assuming it has been written in the first line of the source code file):

```
#line 1 "driver.cpp"
```

What I figured out with `DbgView` was, that this would actually move the line by -1. I solved that now in writing:

```
#line 2 "driver.cpp"
```

in the first source code line.  
I would be interested in knowing why its working like that and if there is

Re: \_\_VA\_ARGS\_\_ and the DDK compiler

some kind of hidden source code line which is inserted by the build tools internally?

Thanks.

Laurents C. R. Meyer  
SEALOG

"Mark Roddy" wrote:

Laurents C. R. Meyer wrote:  
> I find using \_\_VA\_ARGS\_\_ in macros very  
> powerful. I am working some > debugging  
> code over and would like to use  
> \_\_VA\_ARGS\_\_ but until now I was not >  
> able to  
> make the build environment compile it  
> without errors.  
>  
> Does anybody know if there is a  
> compiler switch I have to turn on?  
>  
> Laurents C. R. Meyer  
> SEALOG  
Visual C does not support the C99  
standard. Try using inline functions  
instead.

--

=====  
Mark Roddy DDK MVP  
Windows 2003/XP/2000 Consulting  
Hollis Technology Solutions 603-321-1032  
www.hollistech.com

Re: \_\_VA\_ARGS\_\_ and the DDK compiler