

Re: access violation

Source: <http://www.tech-archive.net/Archive/VC/microsoft.public.vc.language/2004-08/0048.html>

From: Ram Kumar (*efextra_at_newsgroups.nospam*)

Date: 08/02/04

Date: Mon, 2 Aug 2004 11:35:32 +0530

Hi Carolyn,

I didn't find any access violation on running your code in VC 6.0 compiler.
code is given below –

```
#include "stdafx.h"
```

```
typedef struct tagTIMESTAMP_STRUCT  
{  
    short year;  
    unsigned short month;  
    unsigned short day;  
    unsigned short hour;  
    unsigned short minute;  
    unsigned short second;  
    unsigned int fraction;  
} TIMESTAMP_STRUCT;
```

```
typedef struct  
{  
    int nCustomINDSeqNum;  
    TIMESTAMP_STRUCT nCustomLastUpdateTime;  
    char szField1[257];  
    //...  
    char szField255[257];  
    unsigned int nNODE_ID;  
    unsigned int nCUSTOM_APPLICATION_ID;  
} ASPT_EDS_IND_CUSTOM;
```

```
##define DS_C_TYPE(s,m) SetCType(((s *)&DS_SPACE::uDsTypes)->m)
```

```
inline short SetCType(unsigned int nValue)  
{  
    return 0;  
}
```

```
typedef union  
{
```

```
unsigned __int64 u64;
__int64 n64;
unsigned int u32;
int n32;
unsigned short u16;
short n16;
unsigned char u8;
char c8;
TIMESTAMP_STRUCT ts;
double d;
} DS_TYPESu;

namespace DS_SPACE
{
static DS_TYPESu uDsTypes;
};

int main(int argc, char* argv[])
{
printf("Hello World!\n");

//short foo = DS_C_TYPE(ASPT_EDS_IND_CUSTOM, nNODE_ID); // access violation

//Alternative is:
short Raj_short = SetCType(((ASPT_EDS_IND_CUSTOM
*)&DS_SPACE::uDsTypes)->nNODE_ID);

return 0;
}
```

Best Regards,
Ram Kumar

"Carolyn Sano" <CarolynSano@discussions.microsoft.com> wrote in message
news:A57C3918-6B34-4506-997C-8412FB96128B@microsoft.com...

> Hi,

>

> *I have hit an access violation in my code. I know exactly where it's*
happening and what code is causing this. But I'd like someone to explain to
me why this doesn't work. If the following is too long and confusing, I can
cut and paste the code in an email (it's a pretty small test program that I
wrote). It's built in VS C++ 6.0 on Win 2K. Thanks.

>

> *I have this large structure (~65568 bytes) that I wrote.*

>

> *My structure is defined as:*

> *typedef struct*

> {

> *int nCustomINDSeqNum;*

> *TIMESTAMP_STRUCT nCustomLastUpdateTime;*

```
> char szField1[257];
> ...
> char szField255[257];
> unsigned int nNODE_ID;
> unsigned int nCUSTOM_APPLICATION_ID;
> } ASPT_EDS_IND_CUSTOM;
>
> My colleague's code is this macro:
> #define DS_C_TYPE(s,m) SetCType(((s *)&DS_SPACE::uDsTypes)->m)
>
> I call the macro like this:
> short foo = DS_C_TYPE(ASPT_EDS_IND_CUSTOM, nNODE_ID); // access violation
>
> The expansion of the macro is this: SetCType(((ASPT_EDS_IND_CUSTOM
*)&DS_SPACE::uDsTypes)->nNODE_ID)
>
> The method is this:
> inline short SetCType(unsigned int nValue)
> {
> return 0;
> }
>
> other stuff:
> typedef struct tagTIMESTAMP_STRUCT
> {
> short year;
> unsigned short month;
> unsigned short day;
> unsigned short hour;
> unsigned short minute;
> unsigned short second;
> unsigned int fraction;
> } TIMESTAMP_STRUCT;
>
> typedef union
> {
> unsigned __int64 u64;
> __int64 n64;
> unsigned int u32;
> int n32;
> unsigned short u16;
> short n16;
> unsigned char u8;
> char c8;
> TIMESTAMP_STRUCT ts;
> double d;
> } DS_TYPESu;
>
> namespace DS_SPACE { static DS_TYPESu uDsTypes; };
>
>
```