
Subject: Random windows VM errors

Posted by [Joost Aan de Brugh](#) on Thu, 10 Jul 2008 16:28:04 GMT

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

Hello all,

I have been programming in IDL for a while. And now I have quite a substantial program. It is made in a Linux environment where we have IDL. But it also has to work in a Windows environment where we have just the IDL virtual machine.

I start with generating some constants. I read in three ini-files and I set some values. Those values are used all over the program. Sometimes the value is different for windows (Those values have mostly to do with widget layout). It looks like:

```
Function Main::Init, windows=windows
```

```
    w = Keyword_Set(windows)

    self.consts = Ptr_New({TAG1:0, TAG2:"", TAG3:[0,0], ... Big
structure}); This has to be carried around
...
    (*self.consts).TAG34 = w?Value1:Value2 ; Both values are of equal
type and equal logic.
    (*self.consts).TAG67 = Value ; This value is independent of
platform.
...
    openr, lun, "file.ini", /Get_Lun
    ; Read in more variables (independent of w). This is done without
"Execute", because it has to work in a virtual machine. I wrote a
simple parser for reading KEY = VALUE
    close, lun
    Free_Lun, lun
... ; More ini-files are read

    ; Create the program
    void = Obj_New("Program", self, ...)

    return, 1
```

```
End
```

```
In Linux I do
    mnk = Obj_New("Main")
```

```
The Windows IDL VM will do
    mnk = Obj_New("Main", /windows)
```

and I get rude errors like "Array has a corrupted descriptor" and/or instruction 0xbladibla tries to acces 0xfffff... and horribly fails.

Whether or not these errors occur looks completely random, but sometimes it even depends on whether you opened a .sav-file with the IDL virtual machine (open with) or you first open the VM and then open the .sav-file.

I tried to locate the problem by constructing and realizing empty top level base widgets at in the middle of the code to see whether the error occurs before or after the construction of that widget:

```
if w then begin
  windowsfix = Widget_Base(Title="Monkey")
  Widget_Control, windowsfix, /realize
end
```

```
; At end of constuctor:
if w then Widget_Control, windowsfix, /destroy
```

Somehow the error desappeared when I did this. And later it came back and it disappeared when I created and destroyed another one. In Linux, everything is okay.

Does anyone know what causes this randomness. I use a huge struct, pointers, objects and ini-files, but no Execute or Call_Whatever

Thanks in advance,
Joost Aan de Brugh
