
Subject: Re: objects, crashes, and negative memory oh my
Posted by [Pete Warner](#) on Fri, 01 Dec 2006 19:32:59 GMT
[View Forum Message](#) <> [Reply to Message](#)

The whole problem started when I replaced file_test() with file_search() and then forgot to change the checking line from
if file_a_good and file_b_good then...
to
if (file_a ne "") and (file_b ne "") then...

I got obsessed with the thought that it must be an object problem because I don't have much experience there and it looked like I must be leaking memory somewhere. When stepping through my code the crash never happened immediately after the bad "and", but maybe that was just luck.

Thanks everybody. Karl, I have been in contact with RSI support. How can I make sure there aren't two different people working on this separately? I'll send them the latest from the group here.

On Dec 1, 2:19 pm, "Karl Schultz" <k_remove_schu...@ittvis.com> wrote:

> On Fri, 01 Dec 2006 11:41:08 -0700, Jean H. wrote:

>> Pete Warner wrote:

>>> Here's a program that crashes me every time. The key part seems to be
>>> the "file_search" followed by the "and" statement. Take either out and
>>> it doesn't cause me trouble. I'm creating the .txt file because I don't
>>> know what to search for on your computers, and file_search needs to
>>> find something. I run between 1 and 10 loops before IDL dies on my
>>> computer. Tested on 6.1 and 6.3.

>

>> even a 1 line

>> IDL> a = file_search() & print, 'test ' and a

>> makes IDL crash...

>> IDL> print, lversion

>> { x86 Win32 Windows Microsoft Windows 6.3 Mar 23 2006 32 64}

>

>> JeanOuch. Ok I'll open a bug report and we'll try to fix it for next release.

>

> I've spent a little time looking for a workaround and didn't have much
> luck.

>

> About all I can suggest is looking at how the AND operator works on
> non-numeric values and maybe changing your code to do the same thing
> without using AND.

>

> Docs say:

>

> For operations on other types, the result is equal to the second
> operand if the first operand is not equal to zero or the null string;

```
> otherwise, the result is zero or the null string.
>
> So you might replace:
>
> print, 'anystring' and b
>
> with
>
> print, 'anystring' ne '' ? b : ''
>
> Now 'anystring' ne '' doesn't make sense unless 'anystring' is another
> variable.
>
> Entire program, with a workaround:
>
> pro testfailure
>
> a = 'test_file.txt'
> openw, 1, a
> close, 1
> free_lun, 1
>
> key = 'b'
> any = 'anystring'
> count = 0
> print, 'Starting Memory: ', memory(/current)
> while (key ne 'a') do begin
>   count++
>   b = file_search(a)
>   print, any ne '' ? b : ''
>   print, 'Memory after iteration ', count, ', ', memory(/current)
>   print, 'Hit "a" to exit, anykey to continue'
>   key = get_kbrd(1)
> endwhile
> print, 'ending'
>
> end
```
