
Subject: quirk End of file encountered before end of program
Posted by [LC's No-Spam Newsread\[1\]](#) on Thu, 11 Dec 2014 15:03:53 GMT
[View Forum Message](#) <> [Reply to Message](#)

I am trying to adapt some old code to a new task. I *occasionally* get
(IDL 8.4 on Linux)

```
return  
  ^
```

% Syntax error.
At: xxx.pro, Line 26

% End of file encountered before end of program.
At: xxx.pro, Line 28

although the program contains a perfectly valid return and end statement

Then I strip the program of almost all code, and it compiles. Then I
gradually add code until I get the error. Each time I exit and
reenter IDL.

I had a thought that it could be a sequence of begin end clauses in a
case statement (could it confuse the end of a begin end with the
compiler end ?) but then I remove entirely the case statement and go
back to the code which last time compiled, and it still gives the
error.

The code should look something like this

```
pro xxx.pro, ARGUMENTS  
  openw, 2, 'temporary.temp'  
  printf, 2, 'structure=create_struct( $'
```

CASE STATEMENT FILLING THE STRUCTURE DEFINITION in the temp file
(the default for an "empty" structure would just be)
 printf, 2, " 'data', -1 \$"

```
printf, 2, ')'  
close, 2  
; and instantiate it  
  @ temporary.temp  
;help, structure  
return  
end
```

sometimes the code with the CASE gives the error, sometimes doesn't.
Sometimes the code without the CASE gives the error, I comment out the
batch invocation of @ temporary.temp, and it doesn't ... then I reinsert

the batch invocation and then it works.

I also have the impression sometimes it finds an early version of temporary.temp or does not find it even if existing. But the file is on a local disk, not NFS (these glitches can occur if nfs client and server are out of sync), and moreover we use ntp to keep all our clocks on synch.

Any clue ? I hate non-reproducible errors ...

--

nospam@mi.iasf.cnr.it is a newsreading account used by more persons to avoid unwanted spam. Any mail returning to this address will be rejected. Users can disclose their e-mail address in the article if they wish so.

Subject: Re: quirk End of file encountered before end of program
Posted by [Jim Pendleton](#) on Thu, 11 Dec 2014 15:45:32 GMT
[View Forum Message](#) <> [Reply to Message](#)

On Thursday, December 11, 2014 8:03:55 AM UTC-7, LC's No-Spam Newsreading account wrote:

> I am trying to adapt some old code to a new task. I *occasionally* get
> (IDL 8.4 on Linux)
>
> return
> ^
> % Syntax error.
> At: xxx.pro, Line 26
>
> % End of file encountered before end of program.
> At: xxx.pro, Line 28
>
> although the program contains a perfectly valid return and end statement
>
> Then I strip the program of almost all code, and it compiles. Then I
> gradually add code until I get the error. Each time I exit and
> reenter IDL.
>
> I had a thought that it could be a sequence of begin end clauses in a
> case statement (could it confuse the end of a begin end with the
> compiler end ?) but then I remove entirely the case statement and go
> back to the code which last time compiled, and it still gives the
> error.
>
> The code should look something like this

```

>
> pro xxx.pro,ARGUMENTS
>   openw,2,'temporary.temp'
>   printf,2,'structure=create_struct( $'
>
> CASE STATEMENT FILLING THE STRUCTURE DEFINITION in the temp file
> (the default for an "empty" structure would just be)
>   printf,2," 'data',-1 $"
>
>   printf,2,')'
>   close,2
> ; and instantiate it
>   @ temporary.temp
> ;help,structure
>   return
> end
>
> sometimes the code with the CASE gives the error, sometimes doesn't.
> Sometimes the code without the CASE gives the error, I comment out the
> batch invocation of @ temporary.temp, and it doesn't ... then I reinsert
> the batch invocation and then it works.
>
> I also have the impression sometimes it finds an early version of
> temporary.temp or does not find it even if existing. But the file is on
> a local disk, not NFS (these glitches can occur if nfs client and server
> are out of sync), and moreover we use ntp to keep all our clocks on
> synch.
>
> Any clue ? I hate non-reproducible errors ...
>
>
> --
> -----
> nospam@mi.iasf.cnr.it is a newsreading account used by more persons to
> avoid unwanted spam. Any mail returning to this address will be rejected.
> Users can disclose their e-mail address in the article if they wish so.

```

I suspect this isn't behaving as you wish it to behave, due to a misinterpretation of how "@" is used.

The "@" in your code is interpreted at compile time, not at run time.

Make a simple example

```

pro abc
@def
print, 'here'
end

```

Attempt to compile it, without executing it. You'll notice that you get an error "%Error opening file. File: def". But the routine still compiles. Try

IDL> abc

Create a file "def" and put a single line in it

print, 'there'

Do not recompile abc. Simply execute it.

The output does not change. Recompile abc.pro and execute and this time the "def" file is included and executed, as expected.

In your example, you will always be out of synch by one execution with respect to your "@temporary.temp". It's including the version of the file from your previous execution, not the version of the file you are generating on the fly. When you get syntax errors, it's highly likely this is because there's an error in the syntax of the "temporary.temp" file from the previous execution.

I expect what you really want here (short of a logical redesign involving hashes) will be one or more calls to CREATE_STRUCT rather than what you have now.

Jim P

"I work for Exelis"

Subject: Re: quirk End of file encountered before end of program
Posted by [LC's No-Spam Newsread\[1\]](#) on Fri, 12 Dec 2014 13:38:50 GMT
[View Forum Message](#) <> [Reply to Message](#)

On Thu, 11 Dec 2014, Jim P wrote:

```
>> The code should look something like this
>>
>> pro xxx.pro, ARGUMENTS
>>   openw, 2, 'temporary.temp'
>>   printf, 2, 'structure=create_struct( $'
>> CASE STATEMENT FILLING THE STRUCTURE DEFINITION in the temp file
>> printf, 2, ')'
>> close, 2
>> ; and instantiate it
>>   @ temporary.temp
>> return
>> end
```

```
> I suspect this isn't behaving as you wish it to behave, due to a
> misinterpretation of how "@" is used.
```

>
> The "@" in your code is interpreted at compile time, not at run time.

I did not realize that.

The original code I had (dating to several years ago, before anonymous structures were introduced) used in fact a much more complex mechanism.

It wrote the structure definition (without use of CREATE_STRUCT which did not exist) to the temporary.temp file

then wrote a xxxtmpnnn.pro file with nnn increasing each time, which contained the invocation of @temporary.temp (and passed back the created structure as argument)

then built a string with the invocation of xxxtmpnnn.pro and used execute to invoke it

When I did now some simple tests, I thought that the above arrangement was become unnecessary now with create_struct and anonymous structures. I wanted to simplify it and hoped it were NOW possible.

Apparently I cannot do execute('@temporary.temp')

Is there any limitation to the string length in execute(string) ?

I used a temporary file because the structure definition is rather long, and is constructed in steps (driven from a data file) ... for legibility temporary.temp has the statement on several lines (terminated by the \$ continuation marker but last one)

If there is no limitation I could concatenate the pieces of the statement instead of writing them to temporary.temp and just execute the resulting string.

Subject: Re: quirk End of file encountered before end of program
Posted by [lecacheux.alain](#) on Fri, 12 Dec 2014 14:38:53 GMT
[View Forum Message](#) <> [Reply to Message](#)

On Friday, December 12, 2014 2:38:52 PM UTC+1, LC's No-Spam Newsreading account wrote:

> On Thu, 11 Dec 2014, Jim P wrote:

>
>>> The code should look something like this
>>>
>>> pro xxx.pro, ARGUMENTS
>>> openw, 2, 'temporary.temp'

```

>>> printf,2,'structure=create_struct( $'
>>> CASE STATEMENT FILLING THE STRUCTURE DEFINITION in the temp file
>>> printf,2,')'
>>> close,2
>>> ; and instantiate it
>>> @ temporary.temp
>>> return
>>> end
>
>> I suspect this isn't behaving as you wish it to behave, due to a
>> misinterpretation of how "@" is used.
>>
>> The "@" in your code is interpreted at compile time, not at run time.
>
> I did not realize that.
>
> The original code I had (dating to several years ago, before anonymous
> structures were introduced) used in fact a much more complex mechanism.
>
> It wrote the structure definition (without use of CREATE_STRUCT which
> did not exist) to the temporary.temp file
>
> then wrote a xxtmpnnn.pro file with nnn increasing each time, which
> contained the invocation of @temporary.temp (and passed back the
> created structure as argument)
>
> then built a string with the invocation of xxtmpnnn.pro
> and used execute to invoke it
>
> When I did now some simple tests, I thought that the above arrangement
> was become unnecessary now with create_struct and anonymous structures.
> I wanted to simplify it and hoped it were NOW possible.
>
>
> Apparently I cannot do execute('@temporary.temp')
>
> Is there any limitation to the string length in execute(string) ?
>
> I used a temporary file because the structure definition is rather long,
> and is constructed in steps (driven from a data file) ... for legibility
> temporary.temp has the statement on several lines (terminated by the $
> continuation marker but last one)
>
> If there is no limitation I could concatenate the pieces of the
> statement instead of writing them to temporary.temp and just execute the
> resulting string.

```

if you do not want to initialize your structure directly in the main procedure (by `x = {...}`), you could

create an external function 'fun.pro' like:

```
function fun
  x = {...}
  return, x
end
```

and call it, from the main procedure by `x = call_function('fun')`.

alx.

Subject: Re: quirk End of file encountered before end of program
Posted by [LC's No-Spam Newsread\[1\]](#) on Mon, 15 Dec 2014 11:42:34 GMT
[View Forum Message](#) <> [Reply to Message](#)

On Fri, 12 Dec 2014, alx wrote:

> LC's No-Spam Newsreading account wrote:

>> On Thu, 11 Dec 2014, Jim P wrote:

>>

>>>> The code should look something like this

>>>>

>>>> pro xxx.pro, ARGUMENTS

>>>> openw, 2, 'temporary.temp'

>>>> printf, 2, 'structure=create_struct(\$'

>>>> CASE STATEMENT FILLING THE STRUCTURE DEFINITION in the temp file

>>>> printf, 2, ')

>>>> close, 2

>>>> ; and instantiate it

>>>> @ temporary.temp

>>>> return

>>>> end

>>

>>> I suspect this isn't behaving as you wish it to behave, due to a

>>> misinterpretation of how "@" is used.

>>>

>>> The "@" in your code is interpreted at compile time, not at run time.

>>

>> I did not realize that.

>>

>> The original code I had (dating to several years ago, before anonymous

>> structures were introduced) used in fact a much more complex mechanism.

and even had a comment I wrote saying what Jim says 4 lines above !

>> Is there any limitation to the string length in `execute(string)` ?

>>

>> I used a temporary file because the structure definition is rather long,

>> and is constructed in steps (driven from a data file) ... for legibility

>> temporary.temp has the statement on several lines (terminated by the \$

>> continuation marker but last one)
>>
>> If there is no limitation I could concatenate the pieces of the
>> statement instead of writing them to temporary.temp and just execute the
>> resulting string.

Apparently I tried that, and did not encounter any problem with at least about 200 structure elements.

> if you do not want to initialize your structure directly in the main
> procedure (by x = {...}), you could create an external function

Sorry ALX this comment is out of scope. I may or may not do it in the main procedure or in a second level function (actually the original code did both for two different cases).

My point was about building the structure definition at runtime (essentially, the structure tag names are driven by the data file being read)

Subject: Re: quirk End of file encountered before end of program
Posted by [Yngvar Larsen](#) on Mon, 15 Dec 2014 15:05:19 GMT
[View Forum Message](#) <> [Reply to Message](#)

On Friday, 12 December 2014 14:38:52 UTC+1, LC's No-Spam Newsreading account wrote:
> On Thu, 11 Dec 2014, Jim P wrote:

>
> The original code I had (dating to several years ago, before anonymous
> structures were introduced) used in fact a much more complex mechanism.
>

Wow. Are you sure? Both named and anonymous structures have been there at least since IDL 4.0 (maybe even from the very beginning?), released several _decades_ ago.

It is usually more fruitful if you describe also what you are trying to achieve instead of only what you tried that didn't work.

<http://xyproblem.info>

In your case, I really don't see why you cannot use CREATE_STRUCT(tag1, value1, tag2, value2, ...) directly. After all, you must have all tags and values available as variables in your xxx.pro to do what you tried.

I don't know what the contents of your datafile is. Binary or text? But assuming text, one struct tag/value per line, one struct per datafile, and the existence of a procedure PARSELINE that takes a string (line) as input and returns TAG and VALUE, you can do something like this slightly old fashioned code:

```
datafile = '/some/file'
openr, unit, datafile, /get_lun
line = ""
init = 0
while ~eof(unit) do begin
    readf, unit, line
    parseline, line, tag, value
    if init eq 0 then begin
        struct = create_struct(tag, value)
        init = 1
    endif else begin
        struct = create_struct(tag, value, struct)
    endelse
endwhile
free_lun, unit
```

Should work for very old IDL versions.

--
Yngvar

Subject: Re: quirk End of file encountered before end of program
Posted by [LC's No-Spam Newsread\[1\]](#) on Wed, 17 Dec 2014 10:28:36 GMT
[View Forum Message](#) <> [Reply to Message](#)

On Mon, 15 Dec 2014, Yngvar Larsen wrote:

```
> ! LC's No-Spam Newsreading account wrote:
>>
>> The original code I had (dating to several years ago, before anonymous
>> structures were introduced) used in fact a much more complex mechanism.
>
> Wow. Are you sure? Both named and anonymous structures have been there
> at least since IDL 4.0 (maybe even from the very beginning?), released
> several _decades_ ago.
```

Actually the original scripts are dated 1996 (they could be older, possibly I wasn't using rsync to inherit things when I moved machine before then :-)). I am a sporadic but regular IDL user ... after all IDL was used by Giotto at the time of the last Halley comet pass (1986) ... although I started a bit later.

> In your case, I really don't see why you cannot use
> CREATE_STRUCT(tag1, value1, tag2, value2, ...) directly.

In fact I tested it, and now do it creating a long statement and using
execute statement at the end.

```
> if init eq 0 then begin
>   struct = create_struct(tag, value)
>   init = 1
> endif else begin
>   struct = create_struct(tag, value, struct)
> endelse
```

Hmm ... this way you are PREpending new tags ... and possibly creating a
nested structure ?

I want a sort-of flat structure where tags are APpended at the end.
I do something like this:

```
stmt='structure=create_struct( '

stmt=stmt+"'data', ...."

loop on "keywords"
  stmt=stmt+" '"+name+"', "+value
end loop

stmt=stmt+')'
test=execute(stmt)
```

The first tag is special, it can be a data array or a data substructure.
All other tags come later, and are keywords read from the data file, and
assigned heuristically a data type.

So far I haven't encountered problems in building long statements
(structure with about 200 tags)

For me the issue is closed (happily)

Subject: Re: quirk End of file encountered before end of program
Posted by [lecacheux.alain](#) on Wed, 17 Dec 2014 16:49:14 GMT
[View Forum Message](#) <> [Reply to Message](#)

On Wednesday, December 17, 2014 11:28:38 AM UTC+1, LC's No-Spam Newsreading account
wrote:

> On Mon, 15 Dec 2014, Yngvar Larsen wrote:
>

```

>> ! LC's No-Spam Newsreading account wrote:
>>>
>>> The original code I had (dating to several years ago, before anonymous
>>> structures were introduced) used in fact a much more complex mechanism.
>>
>> Wow. Are you sure? Both named and anonymous structures have been there
>> at least since IDL 4.0 (maybe even from the very beginning?), released
>> several _decades_ ago.
>
> Actually the original scripts are dated 1996 (they could be older,
> possibly I wasn't using rsync to inherit things when I moved machine
> before then :-)). I am a sporadic but regular IDL user ... after all IDL
> was used by Giotto at the time of the last Halley comet pass (1986) ...
> although I started a bit later.
>
>> In your case, I really don't see why you cannot use
>> CREATE_STRUCT(tag1, value1, tag2, value2, ...) directly.
>
> In fact I tested it, and now do it creating a long statement and using
> execute statement at the end.
>
>> if init eq 0 then begin
>>   struct = create_struct(tag, value)
>>   init = 1
>> endif else begin
>>   struct = create_struct(tag, value, struct)
>> endelse
>
> Hmm ... this way you are PREpending new tags ... and possibly creating a
> nested structure ?
>
> I want a sort-of flat structure where tags are APpended at the end.
> I do something like this:
>
> stmt='structure=create_struct( '
>
> stmt=stmt+"data', ...."
>
> loop on "keywords"
>   stmt=stmt+"','"+name+"','"+value
> end loop
>
> stmt=stmt+')'
> test=execute(stmt)
>
> The first tag is special, it can be a data array or a data substructure.
> All other tags come later, and are keywords read from the data file, and
> assigned heuristically a data type.

```

>
> So far I haven't encountered problems in building long statements
> (structure with about 200 tags)
>
> For me the issue is closed (happily)

> For me the issue is closed (happily)

But you would have saved your time by reading the documentation more carefully.

Usage of CREATE_STRUCT is very flexible:

new_struct = CREATE_STRUCT(..., old_structure, ...)

where ... can be i) others structures, ii) tag-value pairs, iii) array of tags, array of values. In any order.

alx.

Subject: Re: quirk End of file encountered before end of program

Posted by [Yngvar Larsen](#) on Thu, 18 Dec 2014 07:57:11 GMT

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

On Wednesday, 17 December 2014 11:28:38 UTC+1, LC's No-Spam Newsreading account wrote:

> On Mon, 15 Dec 2014, Yngvar Larsen wrote:

>

>> ! LC's No-Spam Newsreading account wrote:

>>>

>>> The original code I had (dating to several years ago, before anonymous
>>> structures were introduced) used in fact a much more complex mechanism.

>>

>> Wow. Are you sure? Both named and anonymous structures have been there

>> at least since IDL 4.0 (maybe even from the very beginning?), released

>> several _decades_ ago.

> Actually the original scripts are dated 1996 (they could be older,

> possibly I wasn't using rsync to inherit things when I moved machine

> before then :-)). I am a sporadic but regular IDL user ... after all IDL

> was used by Giotto at the time of the last Halley comet pass (1986) ...

> although I started a bit later.

Old indeed! But in 1996, anonymous structures were definitely there. I started using IDL around that time myself.

Automatic structure definition via CREATE_STRUCT(name=mystruct) or {mystruct}, using a file mystruct__define.pro came later, I think.

>> In your case, I really don't see why you cannot use

```
>> CREATE_STRUCT(tag1, value1, tag2, value2, ...) directly.
>
> In fact I tested it, and now do it creating a long statement and using
> execute statement at the end.
>
>> if init eq 0 then begin
>>   struct = create_struct(tag, value)
>>   init = 1
>> endif else begin
>>   struct = create_struct(tag, value, struct)
>> endelse
>
> Hmm ... this way you are PREpending new tags ...
```

```
struct = create_struct(struct, tag, value)
```

> and possibly creating a nested structure ?

No nesting here, unless one of your VALUES happen to be a structure.

> For me the issue is closed (happily)

Good for you. However, I will strongly advise to always reconsider your approach whenever you feel you need to use EXECUTE. This is almost always a sign that you are doing it wrong, or at least reinventing some of IDLs wheels.

--

Yngvar
