Subject: Re: Most Common IDL Programming Errors
Posted by Vince Hradil on Wed, 09 Apr 2008 00:27:28 GMT

View Forum Message <> Reply to Message

On Apr 8, 7:13 pm, David Fanning <n...@dfanning.com> wrote:
> Folks,
>
> I realize you don't much like to play games (I'm thinking
> back to the "Yo mama's so ..." thread), but you have no
> idea how hard it is to come up with new topics for my IDL
> Tips page every week. (Especially so because I try to
> understand what I write and make available, which
> pretty much rules out 4/5 of what I read on this newsgroup.)
>
> Anyway, I was sitting here wondering why we don't have any
> beer in the house again, and it occurred to me that I should
> have a page listing the 10 or 15 most common IDL programming
> errors with their solutions. But I can only think of three.
>
> Here are the three errors I most commonly see in IDL programs.
>
> 1. IDL programs are named incorrectly. The last program
>   module in the file should have the same name as the file.
>   Utility modules in the file should start with the name
>   of the "command" (or last) module to make clear their
>   purpose.
>
> 2. KEYWORD_SET is used to check whether a keyword is
>   "used" or "defined". This function should only be
>   used with binary keywords. (I plan to avoid all
>   nuance with this list, and just go with black and
>   white pronouncements.)
>
> 3. People draw graphics willy-nilly in widget programs
>   without having the faintest idea which window their
>   graphics might show up in.
>
> What would be on your list? Of course, typos are assumed. :-)
>
> Cheers,
>
> David
> --
> David Fanning, Ph.D.
> Fanning Software Consulting, Inc.
> Coyote's Guide to IDL Programming:http://www.dfanning.com/
> Sepore ma de ni thui. ("Perhaps thou speakest truth.")

How about:
-indexing for-loops with (short) integers.
-using indeces from where without checking count > 0

Subject: Re: Most Common IDL Programming Errors
Posted by Michael Galloy on Wed, 09 Apr 2008 00:35:53 GMT

View Forum Message <> Reply to Message

David Fanning wrote:
> Here are the three errors I most commonly see in IDL programs.
>
> 1. IDL programs are named incorrectly. The last program
>   module in the file should have the same name as the file.
>   Utility modules in the file should start with the name
>   of the "command" (or last) module to make clear their
>   purpose.
>
> 2. KEYWORD_SET is used to check whether a keyword is
>   "used" or "defined". This function should only be
>   used with binary keywords. (I plan to avoid all
>   nuance with this list, and just go with black and
>   white pronouncements.)
>
> 3. People draw graphics willy-nilly in widget programs
>   without having the faintest idea which window their
>   graphics might show up in.
>
> What would be on your list? Of course, typos are assumed. :-)

Here are a few (in no particular order, despite the numbers):

1. Thinking you are using a color table in direct graphics, but are
actually using decomposed color (shades of red problem).

2. Not setting up color in PS output correctly.

3. Confusion with direct graphics output in landscape mode in PS.

4. Not scaling object graphics into a coordinate system correctly.

Mike
--
www.michaelgalloy.com
Tech-X Corporation
Software Developer II

Subject: Re: Most Common IDL Programming Errors
Posted by Vince Hradil on Wed, 09 Apr 2008 00:40:01 GMT
View Forum Message <> Reply to Message

On Apr 8, 7:27 pm, Vince Hradil <hrad...@yahoo.com> wrote:
> On Apr 8, 7:13 pm, David Fanning <n...@dfanning.com> wrote:
>
>> Folks,
>
>> I realize you don't much like to play games (I'm thinking
>> back to the "Yo mama's so ..." thread), but you have no
>> idea how hard it is to come up with new topics for my IDL
>> Tips page every week. (Especially so because I try to
>> understand what I write and make available, which
>> pretty much rules out 4/5 of what I read on this newsgroup.)
>
>> Anyway, I was sitting here wondering why we don't have any
>> beer in the house again, and it occurred to me that I should
>> have a page listing the 10 or 15 most common IDL programming
>> errors with their solutions. But I can only think of three.
>
>> Here are the three errors I most commonly see in IDL programs.
>
>> 1. IDL programs are named incorrectly. The last program
>>   module in the file should have the same name as the file.
>>   Utility modules in the file should start with the name
>>   of the "command" (or last) module to make clear their
>>   purpose.
>
>> 2. KEYWORD_SET is used to check whether a keyword is
>>   "used" or "defined". This function should only be
>>   used with binary keywords. (I plan to avoid all
>>   nuance with this list, and just go with black and
>>   white pronouncements.)
>
>> 3. People draw graphics willy-nilly in widget programs
>>   without having the faintest idea which window their
>>   graphics might show up in.
>
>> What would be on your list? Of course, typos are assumed. :-)
>
>> Cheers,
>
>> David
>> --
>> David Fanning, Ph.D.
>> Fanning Software Consulting, Inc.
>> Coyote's Guide to IDL Programming:http://www.dfanning.com/
>> Sepore ma de ni thui. ("Perhaps thou speakest truth.")

>
> How about:
> -indexing for-loops with (short) integers.
> -using indeces from where without checking count > 0

-not setting /data in call to xyouts

Subject: Re: Most Common IDL Programming Errors
Posted by Peter Mason on Wed, 09 Apr 2008 01:31:05 GMT
View Forum Message <> Reply to Message

David Fanning wrote:
<...>
Anyway, I was sitting here wondering why we don't have any
beer in the house again, and it occurred to me that I should
have a page listing the 10 or 15 most common IDL programming
errors with their solutions. But I can only think of three.
<...>

Some others...

Forgetting about a "catch" statement that lies buried somewhere, hiding other bugs or making a total mystery of execution problems.

Forgetting to use "reform" - stumbling on an unwanted array dimension of 1 that's left after pulling out an array subset.   (Actually I don't know if this one's still there.   I'm a bit behind the times maybe.)

Getting "rows" mixed up with "columns" (whatever they are) when using IDL calls that involve matrices.

Cheers
Peter

Subject: Re: Most Common IDL Programming Errors
Posted by Andrew Cool on Wed, 09 Apr 2008 02:06:32 GMT
View Forum Message <> Reply to Message

On Apr 9, 9:13 am, David Fanning <n...@dfanning.com> wrote:
> Folks,
>
> I realize you don't much like to play games (I'm thinking
> back to the "Yo mama's so ..." thread), but you have no
> idea how hard it is to come up with new topics for my IDL

> Tips page every week. (Especially so because I try to
> understand what I write and make available, which
> pretty much rules out 4/5 of what I read on this newsgroup.)
>
> Anyway, I was sitting here wondering why we don't have any
> beer in the house again, and it occurred to me that I should
> have a page listing the 10 or 15 most common IDL programming
> errors with their solutions. But I can only think of three.
>
> Here are the three errors I most commonly see in IDL programs.
>
> 1. IDL programs are named incorrectly. The last program
>   module in the file should have the same name as the file.
>   Utility modules in the file should start with the name
>   of the "command" (or last) module to make clear their
>   purpose.
>
> 2. KEYWORD_SET is used to check whether a keyword is
>   "used" or "defined". This function should only be
>   used with binary keywords. (I plan to avoid all
>   nuance with this list, and just go with black and
>   white pronouncements.)
>
> 3. People draw graphics willy-nilly in widget programs
>   without having the faintest idea which window their
>   graphics might show up in.
>
> What would be on your list? Of course, typos are assumed. :-)
>
> Cheers,
>
> David
> --
> David Fanning, Ph.D.
> Fanning Software Consulting, Inc.
> Coyote's Guide to IDL Programming:http://www.dfanning.com/
> Sepore ma de ni thui. ("Perhaps thou speakest truth.")

1. Indexing FOR loops with a float
2. Reading undefined variables from a file.
3. Lack of comments in some other cretin's legacy code.

Andrew

Subject: Re: Most Common IDL Programming Errors

Posted by David Fanning on Wed, 09 Apr 2008 02:27:18 GMT
View Forum Message <> Reply to Message

Peter Mason writes:

> Getting "rows" mixed up with "columns" (whatever they are) when using IDL
> calls that involve matrices.

Oh, my gosh! I've been working on an EOF (Empirical
Orthogonal Functions) problem the past 3-4 days, and
I'll bet I spent half my time trying to wrap my mind
around this! "OK, does the LA_SVD routine think like
IDL or like matrix algebra? When it says the first
mode EOF is in the column of the output is that in
a linear algebra sense or an IDL sense?" One more day
of it and I'll be ready to buy Matlab. (Heck, all the
examples are in Matlab code.)

Even now, when I am *sure* I know what I'm doing,
I have momentary doubts. Why don't you write us
a primer, Peter? :-)

Cheers,

David

--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.dfanning.com/
Sepore ma de ni thui. ("Perhaps thou speakest truth.")

Subject: Re: Most Common IDL Programming Errors
Posted by David Fanning on Wed, 09 Apr 2008 02:27:18 GMT
View Forum Message <> Reply to Message

Andrew Cool writes:

> 1. Indexing FOR loops with a float

I think you missed the word "common", Andrew. :-)

I don't even know what this problem is. Can you elaborate?

Cheers,

David

--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.dfanning.com/
Sepore ma de ni thui. ("Perhaps thou speakest truth.")

Subject: Re: Most Common IDL Programming Errors
Posted by Kenneth P. Bowman on Wed, 09 Apr 2008 02:48:44 GMT
View Forum Message <> Reply to Message

In article <MPG.2265d7059ac3a84198a322@news.frii.com>,
David Fanning <news@dfanning.com> wrote:

> Peter Mason writes:
>
>> Getting "rows" mixed up with "columns" (whatever they are) when using IDL
>> calls that involve matrices.
>
> Oh, my gosh! I've been working on an EOF (Empirical
> Orthogonal Functions) problem the past 3-4 days, and
> I'll bet I spent half my time trying to wrap my mind
> around this! "OK, does the LA_SVD routine think like
> IDL or like matrix algebra? When it says the first
> mode EOF is in the column of the output is that in
> a linear algebra sense or an IDL sense?" One more day
> of it and I'll be ready to buy Matlab. (Heck, all the
> examples are in Matlab code.)
>
> Even now, when I am *sure* I know what I'm doing,
> I have momentary doubts. Why don't you write us
> a primer, Peter? :-)
>
> Cheers,
>
> David

David, I have some EOF code that is only a few lines long if you want it
(using LA_EIGENQL).

Ken

Subject: Re: Most Common IDL Programming Errors
Posted by Kenneth P. Bowman on Wed, 09 Apr 2008 02:54:26 GMT
View Forum Message <> Reply to Message

In article <MPG.2265b7fc3128704798a320@news.frii.com>,
David Fanning <news@dfanning.com> wrote:

> What would be on your list? Of course, typos are assumed. :-)

Some of these might be considered typos, but I teach beginners and I see these all the time:

1. Trying to use a function as a procedure.
2. Trying to use a procedure as a function.
3. Forgetting to put a continuation character at the end of a line.
4. Forgetting to put a comma after the last argument before a continuation character.
5. Forgetting to start X-Windows before plotting a graph.
6. File name does not match program name (not even close).
7. Unwillingness to use HELP or ?
8. Using both a / and an = sign with a keyword parameter.

Ken

Subject: Re: Most Common IDL Programming Errors
Posted by Spon on Wed, 09 Apr 2008 07:06:38 GMT

View Forum Message <> Reply to Message

On Apr 9, 3:27 am, David Fanning <n...@dfanning.com> wrote:

Judging by the fact that we've got 11 IDL licences in the building, probably 5-15 users in any given week (including at least one *Real* computer scientist - with a PhD and everything) and I'm the only one who ever seems to use REBIN, REFORM, REVERSE... even READ_BINARY instead of OPENR / FSTAT / READU, I'd say some more major errors are:

- Not knowing where to find the useful built-in functions in the sea of helpfiles
- Coding up your own function when there's a perfectly useful inbuilt command that'll do what you want, and can even do it without three nested FOR loops ;-)
- Writing code that solves one specific problem (particularly using integers that don't change for this dataset but may well do so for the next one, rendering the function useless without major re-working, when a simple call to SIZE would have made it SOOO much more flexible)

So in short, I'd love to see an article entitled something like: "I've only got time to read a dozen helpfiles - which ones should I read?" This could contain the names of the commands and a brief one-line summary of what each one does. Of course, deciding which helpfiles to pick is liable to cause some, uh... heated debate ;-)

FWIW, I'd definitely throw in: SIZE, REFORM, REBIN, TRANSPOSE, WHERE, BYTSCL ... and then maybe ARRAY_INDICES; also TV, PLOT, OPLOT & TVSCL to show the wealth of keywords they accept.

I realise that I'm coming at this from a biophysics-lab-in-academia angle, which may be completely different to other people.

Regards,
Chris

Subject: Re: Most Common IDL Programming Errors
Posted by Maarten[1] on Wed, 09 Apr 2008 08:01:20 GMT
View Forum Message <> Reply to Message

On Apr 9, 12:13 am, David Fanning <n...@dfanning.com> wrote:

> Here are the three errors I most commonly see in IDL programs.

[snip]

> What would be on your list? Of course, typos are assumed. :-)

How about any code that does not use
    compile_opt defint32, strictarr, strictarrsubs
Yes, the first two can be combined with idl2, which gives you an idea how long overdue it is to make this the default. I prefer this way, because it does give you some idea of what changes. You can put this in your startup script, and avoid an awful amount of surprises.

The other errors I observe in IDL code are mostly part of the syntax itself :p

Maarten

Subject: Re: Most Common IDL Programming Errors
Posted by David Fanning on Wed, 09 Apr 2008 13:12:36 GMT
View Forum Message <> Reply to Message

Kenneth P. Bowman writes:

> David, I have some EOF code that is only a few lines long if you want it
> (using LA_EIGENQL).

Yes, thank you. We are having difficulties determining which

is the "correct" implementation around here. We are going
to take a vote later today. I could use more ammunition. :-)

Cheers,

David

--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.dfanning.com/
Sepore ma de ni thui. ("Perhaps thou speakest truth.")

Subject: Re: Most Common IDL Programming Errors
Posted by liamgumley on Wed, 09 Apr 2008 14:07:15 GMT

View Forum Message <> Reply to Message

On Apr 9, 2:06 am, Spon <christoph.b...@gmail.com> wrote:

> Judging by the fact that we've got 11 IDL licences in the building,
> probably 5-15 users in any given week (including at least one *Real*
> computer scientist - with a PhD and everything) and I'm the only one
> who ever seems to use REBIN, REFORM, REVERSE... even READ_BINARY
> instead of OPENR / FSTAT / READU, I'd say some more major errors are:

[stuff deleted]

> FWIW, I'd definitely throw in: SIZE, REFORM, REBIN, TRANSPOSE, WHERE,
> BYTSCL ... and then maybe ARRAY_INDICES; also TV, PLOT, OPLOT & TVSCL
> to show the wealth of keywords they accept.

There are a few books on IDL programming which do a decent job of
describing how to use most of these bread-and-butter procedures and
functions. Check out the usual suspects (D. Fanning, K. Bowman, and
one other guy).

Cheers,
Liam.
Practical IDL Programming
http://www.gumley.com/
PS: Thanks to everyone who helped PIP recently surpass 5000 copies in
print.

Subject: Re: Most Common IDL Programming Errors
Posted by ben.bighair on Wed, 09 Apr 2008 14:22:17 GMT

View Forum Message <> Reply to Message

On Apr 8, 8:13 pm, David Fanning <n...@dfanning.com> wrote:

>
> What would be on your list? Of course, typos are assumed. :-)
>

I add to all-of-the-above...

(a) Not searching www.dfanning.com and comp.lang.idl-pvwave first.

(b) Assuming that object-oriented design is only for graphics and assuming that object-oriented design is not suitable for command line interactivity.

(c) Not buying the great third-party books on IDL.

Cheers,
Ben

Subject: Re: Most Common IDL Programming Errors
Posted by rigby on Wed, 09 Apr 2008 15:26:08 GMT

View Forum Message <> Reply to Message

> What would be on your list?

These are my top gotchas:

1) IDL silently truncates array operations to the shorter array.

```
help, [1,2,3,4] - [1,2]
   [0, 0]
```

(I wish there were a compile_opt switch to prohibit this!0

2) Where() returns a one-element array when there is only one match.

```
help, where([1,2,3,4] EQ 4)
<Expression>   LONG = ARRAY[1]
```

The combination (1) and (2) is deadly:

```
array = [1, 2, 3, 4]
j = where(array EQ max(array))
array -= array[j]
print, array
-3
```

My solution is to use a wrapper around where() which will return a
scalar.

3) Operations which mix signed and unsigned integers can't be trusted.

```
maxuint = 'ffff'xu
help, maxuint
MAXUINT     UINT  = 65535
print, fix(1) GT maxuint
   1
print, long(1) GT maxuint
   0
```

These results can be understood using IDL's promotion rules for mixed
expressions,
but it's just not worth it. My solution: Never use unsigned integers.

(Why would one use unsigned integers, you might ask? I've done a few
bit-level
simulations of digital hardware using IDL. It would be conceptually
simpler to
use unsigned integer variables to represent unsigned quantities or
signed quantities
where the sign bit is not the left-most bit.)

Subject: Re: Most Common IDL Programming Errors
Posted by Kenneth P. Bowman on Wed, 09 Apr 2008 16:31:15 GMT
View Forum Message <> Reply to Message

In article <MPG.22666eb3ce01c4fd98a324@news.frii.com>,
 David Fanning <news@dfanning.com> wrote:

> Kenneth P. Bowman writes:
>
>> David, I have some EOF code that is only a few lines long if you want it
>> (using LA_EIGENQL).
>
> Yes, thank you. We are having difficulties determining which
> is the "correct" implementation around here. We are going
> to take a vote later today. I could use more ammunition. :-)
>
> Cheers,
>
> David

The input array 'values' is an nvar x nt (number of times) observation matrix.

This code first calculates the nvar x nvar covariance matrix 'cov' by using MATRIX_MULTIPLY. For large data sets you may want to compute the covariance matrix out of memory.

The eigenvalues and eigenvectors are found using LA_EIGENQL. The code calculates the first 'nev' eigenvalues and eigenvectors of the covariance matrix. (I say first because we normally sort the eigenvalues from largest to smallest. LA_EIGENQL does the reverse, so actually it is the last 'nev' eigenvalues, hence the calls to REVERSE.)

```
cov      = MATRIX_MULTIPLY(values, values, /BTRANSPOSE)/nt ;Compute covariance matrix
eigenval = LA_EIGENQL(cov, EIGENVECTORS = eigenvec, $      ;Compute eigenvalues and
eigenvectors
   RANGE = [nvar - nev, nvar-1])
eigenval = REVERSE(eigenval)                            ;Sort eigenvalues from largest to smallest
eigenvec = REVERSE(eigenvec, 2)                           ;Sort eigenvectors as eigenvalues
```

Cheers, Ken

Subject: Re: Most Common IDL Programming Errors
Posted by Vince Hradil on Wed, 09 Apr 2008 16:39:55 GMT

View Forum Message <> Reply to Message

```
On Apr 9, 10:26 am, ri...@crd.ge.com wrote:
>>  What would be on your list?
>
>  These are my top gotchas:
>
>  1) IDL silently truncates array operations to the shorter array.
>
>  help, [1,2,3,4] - [1,2]
>     [0, 0]
>
>  (I wish there were a compile_opt switch to prohibit this!0
>
>  2) Where() returns a one-element array when there is only one match.
>
>  help, where([1,2,3,4] EQ 4)
>  <Expression>   LONG = ARRAY[1]
>
>  The combination (1) and (2) is deadly:
>
>  array = [1, 2, 3, 4]
>  j = where(array EQ max(array))
>  array -= array[j]
>  print, array
>  -3
>
>  My solution is to use a wrapper around where() which will return a
>  scalar.
>
>  3) Operations which mix signed and unsigned integers can't be trusted.
>
>  maxuint = 'ffff'xu
>  help, maxuint
>  MAXUINT     UINT  = 65535
>  print, fix(1) GT maxuint
>     1
>  print, long(1) GT maxuint
>     0
>
>  These results can be understood using IDL's promotion rules for mixed
>  expressions,
>  but it's just not worth it.  My solution: Never use unsigned integers.
>
>  (Why would one use unsigned integers, you might ask?  I've done a few
>  bit-level
>  simulations of digital hardware using IDL. It would be conceptually
>  simpler to
>  use unsigned integer variables to represent unsigned quantities or
>  signed quantities
>  where the sign bit is not the left-most bit.)
```

I just got stung with this one:
Changing a passed variable in a function, then assuming it has
changed:

```
function foo, bar
  bar = bar/2.0
  foobar = 7.0 * bar
  return, foobar
end

pro testit
  bar = 8.0
  fb = foo(bar)
  print, fb, bar
end
```

Subject: Re: Most Common IDL Programming Errors
Posted by brodzik@nsidc.org on Wed, 09 Apr 2008 17:11:40 GMT
View Forum Message <> Reply to Message

On Apr 8, 6:13 pm, David Fanning <n...@dfanning.com> wrote:

> beer in the house again, and it occurred to me that I should
> have a page listing the 10 or 15 most common IDL programming
> errors with their solutions. But I can only think of three.
>
> Here are the three errors I most commonly see in IDL programs.

Using the assignment operator '=' instead of comparison 'eq' in a conditional test.

One way to be defensive about this is to be in the habit of putting
the constant value
on the left side, so the compiler catches the syntax error long before
runtime.
For example, this will fail the conditional, but happily reassign the
value of a to 0:

```
IDL>
a=5
IDL> if ( a = 0 ) then print,
'OK'
IDL>
print,a
      0
```

Also a bit of a bear to debug, depending on the conditional
consequences, and how
important it is for a to stay 5...

Whereas this will let the compiler find the problem long before you
run it:

```
IDL>
a=5
IDL> if ( 0 = a ) then
print,'OK'

if ( 0 = a ) then
print,'OK'

^
% Syntax
error.
```

IDL>

On the other hand, my coworkers *hate* reading my
"backwards comparison" style, so if you adopt this style,
you'll get a lot of odd looks at walkthroughs.

MJ

Subject: Re: Most Common IDL Programming Errors
Posted by David Klassen on Wed, 09 Apr 2008 18:05:33 GMT

View Forum Message <> Reply to Message

On Apr 8, 9:31 pm, "Peter Mason" <peter.ma...@deleteme.csiro.au>
wrote:
>
> Getting "rows" mixed up with "columns" (whatever they are) when using IDL
> calls that involve matrices.

Extend this to "planes" for 3-d arrays. I have to admit to writing
programs
with arrays as [z,x,y] and others where they are [x,y,z].
Fortunately, none
of these programs has to interact with the others (directly...)

Subject: Re: Most Common IDL Programming Errors
Posted by bjelley on Wed, 09 Apr 2008 18:05:54 GMT

View Forum Message <> Reply to Message

Frankly I am shocked this hasn't been stated by now, but...

Not explicitly declaring a string before attempting to read a
character string from stdio or from file. The ability of IDL to read
short integers and floats sometimes allows us to take the non-fortran
route and read a variable without any declaration of its type. Not so
with strings. This is one I catch myself doing from time to time.

Cheers,
-Ben

Subject: Re: Most Common IDL Programming Errors
Posted by MarioIncandenza on Wed, 09 Apr 2008 18:34:27 GMT

View Forum Message <> Reply to Message

On Apr 8, 6:31 pm, "Peter Mason" <peter.ma...@deleteme.csiro.au>

wrote:

> Forgetting to use "reform" - stumbling on an unwanted array dimension of 1
> that's left after pulling out an array subset.   (Actually I don't know if
> this one's still there.   I'm a bit behind the times maybe.)

I'm going to second this one, this one still bites me in the rear
every couple of weeks, it seems. And it can be really hard to
diagnose.

Other things that have cost me time recently are:
--remembering the /EXTRACT keyword in STRMATCH() and STREGEX() (BTW,
has anyone implemented RegExp substitution in IDL?)
--syntax for calls to ARRAY_INDICES(), as well as use of
REVERSE_INDICES.

The other ones I recognize from back when I was new at this....

Subject: Re: Most Common IDL Programming Errors
Posted by Andrew Cool on Thu, 10 Apr 2008 02:56:51 GMT
View Forum Message <> Reply to Message

On Apr 9, 11:27 am, David Fanning <n...@dfanning.com> wrote:

> Andrew Cool writes:
>> 1. Indexing FOR loops with a float
>
> I think you missed the word "common", Andrew. :-)
>
> I don't even know what this problem is. Can you elaborate?
>

```
 for i = 0.0,10.000,0.1 do begin
   print,i
 end
```

C'mon David, If you've written about the limitations in representation
of floating point variables,
you've writen about it a hundred times in the past 20 years...

The loop above doesn't ever get to 10.000, under Windows at least.

May I refer you to your posts of the previous Millennium on Dec 1,
2000, under
"How Computers Represent Floats"

It's right after your post on "The Joy of Common Blocks", and
immediately before "Fun with VMS." ;-)

Andrew

> Cheers,
>
> David
> --
> David Fanning, Ph.D.
> Fanning Software Consulting, Inc.
> Coyote's Guide to IDL Programming:http://www.dfanning.com/
> Sepore ma de ni thui. ("Perhaps thou speakest truth.")

Subject: Re: Most Common IDL Programming Errors
Posted by Carsten Lechte on Thu, 10 Apr 2008 09:22:07 GMT
View Forum Message <> Reply to Message

Peter Mason wrote:
> Forgetting to use "reform" - stumbling on an unwanted array dimension of 1
> that's left after pulling out an array subset. (Actually I don't know if
> this one's still there. I'm a bit behind the times maybe.)

Also array related: scalars are not the same as arrays of size one, e.g.

```
PRINT, BYTE( 'test') eq BYTE( 'e')
```

vs what you really wanted:

```
PRINT, BYTE( 'test') eq (BYTE( 'e'))[0]
```

chl

Subject: Re: Most Common IDL Programming Errors
Posted by mmiller3 on Thu, 10 Apr 2008 13:23:25 GMT
View Forum Message <> Reply to Message

>>>> > "David" == David Fanning <news@dfanning.com> writes:

> I think you missed the word "common", Andrew. :-)

That reminds me - using common blocks to pass around vast numbers
of "global" variables and driving those who have to try to

decrypt the code completely nuts.