Subject: Most Common IDL Programming Errors
Posted by David Fanning on Wed, 09 Apr 2008 00:13:17 GMT
View Forum Message <> Reply to Message

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.")

Subject: Re: Most Common IDL Programming Errors
Posted by David Klassen on Wed, 09 Apr 2008 18:10:05 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 do a lot of outputting data into columns of a text file---the one
that
confused me for the longest time is that, unless otherwise stated,
with
the "width=" keyword, the line length of an opened file is 80
characters.

Subject: Re: Most Common IDL Programming Errors
Posted by Doug on Wed, 09 Apr 2008 21:33:12 GMT
View Forum Message <> Reply to Message

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

Naive use of data types:
```
IDL> print, 30000 + 30000
  -5536
IDL> print, 3 / 2
       1
```

Thinking that any non-zero value is true:
```
IDL> if (2) then print, 'True' else print, 'False'
False
```

Doug

Subject: Re: Most Common IDL Programming Errors
Posted by Tom McGlynn on Thu, 10 Apr 2008 02:18:16 GMT
View Forum Message <> Reply to Message

One problem that I used to make all the time was to assume that WHERE
returned a match, e.g.,

```
  w = where(x gt 0)
  x[w] = alog10(x[w])
```

neglecting the case where the x are all negative. My programs would
always work when I tested them, and then I'd find they failed when I
tried them in the real world. The need to check finally got beaten
into me.

Another is fogetting the inherently parallel nature of array
operations, e.g., something like expecting

```
x=[0,1,1,2,2,2]
y=indgen(3)

y[x] = y[x]+1
```

to give something other than 1,1,1.

Regards,
Tom McGlynn

Subject: Re: Most Common IDL Programming Errors
Posted by David Fanning on Thu, 10 Apr 2008 05:29:27 GMT
View Forum Message <> Reply to Message

Andrew Cool writes:

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

Oh, *that* article.

I've got a better idea than to waste time thinking up
new articles: I'm just going to recycle some old ones.
Some of us wouldn't know the difference. ;-)

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 Gaurav on Thu, 10 Apr 2008 07:01:52 GMT

View Forum Message <> Reply to Message

Boy! this is turning out to be one long list. And going through it I realized I have faced most of them and learnt them the hard way.

Since I have been working with images in different formats, I would like to mention that the reading/ display order of various formats is easy to get confused with and when working with multiple formats at the same time, I have displayed my images upside down-many a times.

I hope David extends his list beyond 10-15 to include all the common errors.

Cheers,
Gaurav

Subject: Re: Most Common IDL Programming Errors
Posted by izimine on Thu, 10 Apr 2008 08:51:01 GMT
View Forum Message <> Reply to Message

> Since I have been working with images in different formats, I would
> like to mention that the reading/ display order of various formats is
> easy to get confused with and when working with multiple formats at
> the same time, I have displayed my images upside down-many a times.

+1

I actually like having !order=1 in my defaults since most of the time my images are in top-bottom order. But I have no clue why the display coordinate system is not changed accordingly (both in direct and obj graphics).

Subject: Re: Most Common IDL Programming Errors
Posted by Erik[1] on Thu, 10 Apr 2008 09:29:27 GMT
View Forum Message <> Reply to Message

On 10 apr, 04:18, Tom McGlynn <t...@milkyway.gsfc.nasa.gov> wrote:
> One problem that I used to make all the time was to assume that WHERE
> returned a match, e.g.,
>
> w = where(x gt 0)
> x[w] = alog10(x[w])
>
> neglecting the case where the x are all negative. My programs would
> always work when I tested them, and then I'd find they failed when I
> tried them in the real world. The need to check finally got beaten

> into me.
>
> Another is fogetting the inherently parallel nature of array
> operations, e.g., something like expecting
>
> x=[0,1,1,2,2,2]
> y=indgen(3)
>
> y[x] = y[x]+1
>
> to give something other than 1,1,1.
>
> Regards,
> Tom McGlynn

Quote:
- 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 ;-)

Lol! So true this one!

Subject: Re: Most Common IDL Programming Errors
Posted by Maarten[1] on Thu, 10 Apr 2008 11:35:43 GMT
View Forum Message <> Reply to Message

On Apr 9, 9:33 pm, Doug <doug.lindh...@gmail.com> wrote:
> David Fanning wrote:
>> What would be on your list? Of course, typos are assumed. :-)
>
> Naive use of data types:
> IDL> print, 30000 + 30000
> -5536
> IDL> print, 3 / 2
> 1

compile_opt defint32 in your startup script, and at the beginning of every function/procedure you write.

Maarten

Subject: Re: Most Common IDL Programming Errors
Posted by mmiller3 on Thu, 10 Apr 2008 13:22:03 GMT
View Forum Message <> Reply to Message

>>>> > "Carsten" == Carsten Lechte <chl@toppoint.de> writes:

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

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

By far my most common error is writing "print x" in IDL after
I've been using python for a while and writing "print, x" in
python after using IDL for a while.

Mike

Subject: Re: Most Common IDL Programming Errors
Posted by Vince Hradil on Thu, 10 Apr 2008 14:29:06 GMT
View Forum Message <> Reply to Message

On Apr 10, 12:29 am, David Fanning <n...@dfanning.com> wrote:
> Andrew Cool writes:
>> May I refer you to your posts of the previous Millennium on Dec 1,
>> 2000, under
>> "How Computers Represent Floats"
>
> Oh, *that* article.
>
> I've got a better idea than to waste time thinking up
> new articles: I'm just going to recycle some old ones.
> Some of us wouldn't know the difference. ;-)
>
> 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.")

David,

May I suggest SciGen http://pdos.csail.mit.edu/scigen/ ?

Cheers,

Vince

Subject: Re: Most Common IDL Programming Errors
Posted by Vince Hradil on Thu, 10 Apr 2008 14:31:58 GMT
View Forum Message <> Reply to Message

On Apr 10, 8:22 am, mmill...@iupui.edu (Michael A. Miller) wrote:
>>>> >> "Carsten" == Carsten Lechte <c...@toppoint.de> writes:
>
>> PRINT, BYTE('test') eq BYTE('e')
>
>> PRINT, BYTE('test') eq (BYTE('e'))[0]
>
> By far my most common error is writing "print x" in IDL after
> I've been using python for a while and writing "print, x" in
> python after using IDL for a while.
>
> Mike

Ha - I find myself typing %> cat, file.txt in unix...

Subject: Re: Most Common IDL Programming Errors
Posted by R.G. Stockwell on Thu, 10 Apr 2008 18:16:11 GMT
View Forum Message <> Reply to Message

"David Fanning" <news@dfanning.com> wrote in message
news:MPG.2265b7fc3128704798a320@news.frii.com...
...
> Here are the three errors I most commonly see in IDL programs.

I apologize for not having read all the replies - sorry if this is
duplicating others.

1) multiple copies of a routine.
I have done this a few times, with older "utility" programs that somehow
get duplicated in different parts of the path. Which is fine cause IDL
always
uses the "first" one - until edits make the two copies branch independently
and
then function calls don't do what i think they are doing.

2) running off arrays - (depending on system settings)
two cases, a) one where IDL repeats the last element
and doesn't issue an error. b) where you multiply two arrays
and the longer one gets truncated.

Cheers,
bob

Subject: Re: Most Common IDL Programming Errors
Posted by Tom McGlynn on Fri, 11 Apr 2008 12:55:06 GMT
View Forum Message <> Reply to Message

On Apr 9, 10:18 pm, Tom McGlynn <t...@milkyway.gsfc.nasa.gov> wrote:
...
> Another is fogetting the inherently parallel nature of array
> operations, e.g., something like expecting
>
> x=[0,1,1,2,2,2]
> y=indgen(3)
>
> y[x] = y[x]+1
>
> to give something other than 1,1,1.
>
And perhaps I should include using the XXXgen functions when one
means XXXarr!
Tom

Subject: Re: Most Common IDL Programming Errors
Posted by David Fanning on Fri, 11 Apr 2008 12:58:25 GMT
View Forum Message <> Reply to Message

Tom McGlynn writes:

> And perhaps I should include using the XXXgen functions when one
> means XXXarr!

I usually do exactly the opposite. :-(

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 R.G. Stockwell on Fri, 11 Apr 2008 16:02:54 GMT
View Forum Message <> Reply to Message

"Tom McGlynn" <tam@milkyway.gsfc.nasa.gov> wrote in message
news:08513f0c-9af3-4613-900c-2449d1306fc5@l42g2000hsc.google groups.com...
> On Apr 9, 10:18 pm, Tom McGlynn <t...@milkyway.gsfc.nasa.gov> wrote:
> ...
>> Another is fogetting the inherently parallel nature of array
>> operations, e.g., something like expecting
>>
>> x=[0,1,1,2,2,2]
>> y=indgen(3)
>>
>> y[x] = y[x]+1
>>
>> to give something other than 1,1,1.
>>
> And perhaps I should include using the XXXgen functions when one
> means XXXarr!
> Tom

XXXarr needs a /nan keyword (to fill it with nans).
I always use it like

a = fltarr(len) + !values.f_nan

-bob

Subject: Re: Most Common IDL Programming Errors
Posted by Matt[2] on Fri, 11 Apr 2008 16:30:02 GMT
View Forum Message <> Reply to Message

"R.G. Stockwell" <nothanks@noemail.com> writes:

> a = fltarr(len) + !values.f_nan

What about
a = make_array(len, value=!values.f_nan)

That's what I use. I'm sure it saves me some computational power by not having
to compute those values I'm just going to overwrite anyway. :)

Matt

--
Matthew Savoie - Scientific Programmer
National Snow and Ice Data Center
(303) 735-0785 http://nsidc.org

Subject: Re: Most Common IDL Programming Errors
Posted by R.G. Stockwell on Fri, 11 Apr 2008 17:41:25 GMT
View Forum Message <> Reply to Message

<savoie@nsidc.org> wrote in message
news:ywkufxtss6b9.fsf@snowblow.colorado.edu...
> "R.G. Stockwell" <nothanks@noemail.com> writes:
>
>
>> a = fltarr(len) + !values.f_nan
>
> What about
> a = make_array(len, value=!values.f_nan)
>
> That's what I use. I'm sure it saves me some computational power by not
> having
> to compute those values I'm just going to overwrite anyway. :)

You know, I am just the kind of person who will end up time testing
those variations. And seeing what operation is faster: fltarr(len) +
!values.f_nan,
fltarr(len) - !values.f_nan, fltarr(len) * !values.f_nan, or
fltarr(len)/!values.f_nan

:)

Cheers,
bob

Subject: Re: Most Common IDL Programming Errors
Posted by Vince Hradil on Fri, 11 Apr 2008 17:54:02 GMT
View Forum Message <> Reply to Message

On Apr 11, 12:41 pm, "R.G. Stockwell" <notha...@noemail.com> wrote:
> <sav...@nsidc.org> wrote in message
>

> news:ywkufxtss6b9.fsf@snowblow.colorado.edu...
>
>> "R.G. Stockwell" <notha...@noemail.com> writes:
>
>>> a = fltarr(len) + !values.f_nan
>
>> What about
>> a = make_array(len, value=!values.f_nan)
>
>> That's what I use. I'm sure it saves me some computational power by not
>> having
>> to compute those values I'm just going to overwrite anyway. :)
>
> You know, I am just the kind of person who will end up time testing
> those variations. And seeing what operation is faster: fltarr(len) +
> !values.f_nan,
> fltarr(len) - !values.f_nan, fltarr(len) * !values.f_nan, or
> fltarr(len)/!values.f_nan
>
> :)
>
> Cheers,
> bob

I'm waiting with bated breath. 8^)

Subject: Re: Most Common IDL Programming Errors
Posted by Kenneth P. Bowman on Fri, 11 Apr 2008 18:37:51 GMT
View Forum Message <> Reply to Message

In article <fto23e$fk5$1@aioe.org>,
"R.G. Stockwell" <nothanks@noemail.com> wrote:

> "Tom McGlynn" <tam@milkyway.gsfc.nasa.gov> wrote in message
> news:08513f0c-9af3-4613-900c-2449d1306fc5@l42g2000hsc.google groups.com...
>> On Apr 9, 10:18 pm, Tom McGlynn <t...@milkyway.gsfc.nasa.gov> wrote:
>> ...
>>> Another is fogetting the inherently parallel nature of array
>>> operations, e.g., something like expecting
>>>
>>> x=[0,1,1,2,2,2]
>>> y=indgen(3)
>>>
>>> y[x] = y[x]+1
>>>
>>> to give something other than 1,1,1.
>>>

>> And perhaps I should include using the XXXgen functions when one
>> means XXXarr!
>> Tom
>
>
> XXXarr needs a /nan keyword (to fill it with nans).
> I always use it like
>
> a = fltarr(len) + !values.f_nan
>
> -bob

Or

a = REPLICATE(!VALUES.F_NAN, len)

Ken Bowman

Subject: Re: Most Common IDL Programming Errors
Posted by R.G. Stockwell on Fri, 11 Apr 2008 20:52:56 GMT
View Forum Message <> Reply to Message

"Vince Hradil" <hradilv@yahoo.com> wrote in message
news:b4e816fa-1c96-44a1-96bb-1a4e7395d376@24g2000hsh.googleg roups.com...
> On Apr 11, 12:41 pm, "R.G. Stockwell" <notha...@noemail.com> wrote:
>> <sav...@nsidc.org> wrote in message
>>
>> news:ywkufxtss6b9.fsf@snowblow.colorado.edu...
>>
>>> "R.G. Stockwell" <notha...@noemail.com> writes:
>>
>>>> a = fltarr(len) + !values.f_nan
>>
>>> What about
>>> a = make_array(len, value=!values.f_nan)
>>
>>> That's what I use. I'm sure it saves me some computational power by not
>>> having
>>> to compute those values I'm just going to overwrite anyway. :)
>>
>> You know, I am just the kind of person who will end up time testing
>> those variations. And seeing what operation is faster: fltarr(len) +
>> !values.f_nan,
>> fltarr(len) - !values.f_nan, fltarr(len) * !values.f_nan, or
>> fltarr(len)/!values.f_nan
>>
>> :)

>>
>> Cheers,
>> bob
>
> I'm waiting with bated breath. 8^)

Here they are (10k iterations, ran loops over the different commands, repeated them in different orders, all functions and arrays were previously created):

(and by the way WOW! - make array wins)

plus 89.172000 Seconds.

minus 89.188000 Seconds.

multiply 90.531000 Seconds.

divide 89.485000 Seconds.

makearray 2.0000000 Seconds.

makearray 1.9840000 Seconds.

divide 90.250000 Seconds.

multiply 90.219000 Seconds.

minus 88.797000 Seconds.

plus 88.125000 Seconds.

Subject: Re: Most Common IDL Programming Errors
Posted by Vince Hradil on Fri, 11 Apr 2008 21:23:19 GMT
View Forum Message <> Reply to Message

On Apr 11, 3:52 pm, "R.G. Stockwell" <notha...@noemail.com> wrote:
> "Vince Hradil" <hrad...@yahoo.com> wrote in message
>
> news:b4e816fa-1c96-44a1-96bb-1a4e7395d376@24g2000hsh.googleg roups.com...
>
>> On Apr 11, 12:41 pm, "R.G. Stockwell" <notha...@noemail.com> wrote:
>>> <sav...@nsidc.org> wrote in message
>
>>> news:ywkufxtss6b9.fsf@snowblow.colorado.edu...
>

>>>> "R.G. Stockwell" <notha...@noemail.com> writes:
>
>>>> > a = fltarr(len) + !values.f_nan
>
>>>> What about
>>>> a = make_array(len, value=!values.f_nan)
>
>>>> That's what I use. I'm sure it saves me some computational power by not
>>>> having
>>>> to compute those values I'm just going to overwrite anyway. :)
>
>>> You know, I am just the kind of person who will end up time testing
>>> those variations. And seeing what operation is faster: fltarr(len) +
>>> !values.f_nan,
>>> fltarr(len) - !values.f_nan, fltarr(len) * !values.f_nan, or
>>> fltarr(len)/!values.f_nan
>
>>> :)
>
>>> Cheers,
>>> bob
>
>> I'm waiting with bated breath. 8^)
>
> Here they are (10k iterations, ran loops over the different
> commands, repeated them in different orders, all functions and arrays
> were previously created):
>
> (and by the way WOW! - make array wins)
>
> plus 89.172000 Seconds.
>
> minus 89.188000 Seconds.
>
> multiply 90.531000 Seconds.
>
> divide 89.485000 Seconds.
>
> makearray 2.0000000 Seconds.
>
> makearray 1.9840000 Seconds.
>
> divide 90.250000 Seconds.
>
> multiply 90.219000 Seconds.
>
> minus 88.797000 Seconds.
>

> plus 88.125000 Seconds.

"Wow!" is right. I was being facetious originally, but I think I
really learned something. Thanks for the effort.

Subject: Re: Most Common IDL Programming Errors
Posted by David Fanning on Sat, 12 Apr 2008 18:59:50 GMT
View Forum Message <> Reply to Message

Folks,

Yikes! Forty-six articles in this topic and counting...

I spent the entire morning writing. My hand is cramped
and I still need to do my taxes and get ready for a trip
to Las Vegas next week. (Coyote is already planning his
report!)

Anyway, I'm not even half-way through the list, so I'll
work on it more as I have time, but at least the preliminary
stuff is there. If I missed your favorite, I'm sorry. I'll
try to get to it in version 2. :-)

http://www.dfanning.com/code_tips/mostcommon.html

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 R.Bauer on Sun, 13 Apr 2008 17:28:34 GMT
View Forum Message <> Reply to Message

David Fanning schrieb:
> 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

Hi David

have to apologize too that thread is too long for a quick review. ;)

1. IDL programs are named incorrectly. ...

IDL programs are saved using uppercase letters (or mixed) by a windows users who wonders then why noone found his program.

b) Fun with pointers. missing to use ptr_free

c) fun with file io. using /get_lun but doing close, lun

3. People draw graphics willy-nilly in widget programs
without having the faintest idea which window their
graphics might show up in.

that can gain more fun if they do use several draw windows and try to use oplot later without storing the right sysvars for each draw window.

cheers
Reimar

Subject: Re: Most Common IDL Programming Errors
Posted by pgrigis on Wed, 16 Apr 2008 21:39:32 GMT
View Forum Message <> Reply to Message

> David Fanning 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
>

Well, I must admit that I mistype "print" as "pritn" so often
that at some point I just decided to define a procedure called
"pritn.pro" that passes its arguments to print...

```
IDL> pritn,2+2
% Compiled module: PRITN.
      4
```

Ciao,
Paolo

Subject: Re: Most Common IDL Programming Errors
Posted by rlayberry on Thu, 17 Apr 2008 15:52:25 GMT
View Forum Message <> Reply to Message

On 9 Apr, 01:13, 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. :-)

remembering (sometimes) to use /swap_endian when reading in binary
data across platforms

not deleting large arrays when the program no longer needs them

not commenting at the top of my code what the program does and also
giving short but useless filenames (a1.pro etc)

Subject: Re: Most Common IDL Programming Errors
Posted by gerapagi on Tue, 10 Jul 2012 08:33:44 GMT
View Forum Message <> Reply to Message

On Wednesday, April 9, 2008 5:43:17 AM UTC+5:30, David Fanning 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.")

I just want know where are the references for the formula or algorithm used in IDL?

Subject: Re: Most Common IDL Programming Errors
Posted by David Fanning on Tue, 10 Jul 2012 12:50:35 GMT
View Forum Message <> Reply to Message

gerapagi@gmail.com writes:

> I just want know where are the references for the formula or algorithm used in IDL?

Usually you find this in the description of the
command in the on-line help.

Cheers,

David

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