Subject: Re: IDL 8.0 questions
Posted by David Fanning on Sat, 24 Jul 2010 17:09:54 GMT
View Forum Message <> Reply to Message

Mike Potter writes:

> Is it me or is it bugs?

Hard to say. What kind of machine are you running this on?

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: IDL 8.0 questions
Posted by Mike Potter on Sat, 24 Jul 2010 17:50:20 GMT
View Forum Message <> Reply to Message

On Jul 24, 1:09 pm, David Fanning <n...@dfanning.com> wrote:
> Mike Potter writes:
>> Is it me or is it bugs?
>
> Hard to say. What kind of machine are you running this on?
>
> 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.")

Hi David - it's a windows computer, xp sp3 & up-to-date .

Thanks!

Subject: Re: IDL 8.0 questions
Posted by penteado on Sat, 24 Jul 2010 23:24:20 GMT
View Forum Message <> Reply to Message

On Jul 24, 1:18 pm, Mike Potter <m...@orionsound.com> wrote:
> 1) I'm trying out the new "foreach" capability. The top loop, below,
> works, the bottom one does not.
>
> for i=0,nf-1 do begin
> qfile = file_names[i]
> image = readfits(qfile)
> sky,image,sky,sky_sig,/silent
> q = dialog_message("Sky = "+string(sky)+" Sigma =
> "+string(sky_sig) ,/information)
> endfor
>
> foreach qfile, file_names do begin
> image = readfits(qfile)
> sky,image,sky_mod,sky_sig,/silent
> q = dialog_message("Sky = "+string(sky_mod)+" Sigma =
> "+string(sky_sig) ,/information)
> endforeach
>
> After running the bottom loop, the first time readfits is called it
> crashes and the input "qfile" has been changed and either contains two
> characters equal to the final two characters of what qfile was on
> input, or it returns a short string of garbage characters.
> (file_names is a string array of filenames returned by
> dialog_pickfile).

I do not see why the second loop should not work the same as the first. I can only suggest to check the values in file_names before running each loop, or during the foreach loop, to make sure they are actually the same.

Subject: Re: IDL 8.0 questions
Posted by Mike Potter on Sun, 25 Jul 2010 06:07:52 GMT
View Forum Message <> Reply to Message

On Jul 24, 7:24 pm, Paulo Penteado <pp.pente...@gmail.com> wrote:
> On Jul 24, 1:18 pm, Mike Potter <m...@orionsound.com> wrote:
>
>
>
>
>
>> 1) I'm trying out the new "foreach" capability. The top loop, below,

>> works, the bottom one does not.
>
>> for i=0,nf-1 do begin
>> qfile = file_names[i]
>> image = readfits(qfile)
>> sky,image,sky,sky_sig,/silent
>> q = dialog_message("Sky = "+string(sky)+" Sigma =
>> "+string(sky_sig) ,/information)
>> endfor
>
>> foreach qfile, file_names do begin
>> image = readfits(qfile)
>> sky,image,sky_mod,sky_sig,/silent
>> q = dialog_message("Sky = "+string(sky_mod)+" Sigma =
>> "+string(sky_sig) ,/information)
>> endforeach
>
>> After running the bottom loop, the first time readfits is called it
>> crashes and the input "qfile" has been changed and either contains two
>> characters equal to the final two characters of what qfile was on
>> input, or it returns a short string of garbage characters.
>> (file_names is a string array of filenames returned by
>> dialog_pickfile).
>
> I do not see why the second loop should not work the same as the
> first. I can only suggest to check the values in file_names before
> running each loop, or during the foreach loop, to make sure they are
> actually the same.- Hide quoted text -
>
> - Show quoted text -

Well, there's definitely something odd going on. Basically as long as there is a "foreach" loop in the code some very strange things happen. It may be some interaction with the AstroLib function READFITS - I have the latest version downloaded just a couple of days ago - shows last update as 7/16/2010. Note, though, I get similar behavior using "FITS_READ" instead. It really seems to be an issue with "FOREACH" loops.

Anyway, here's what happens:

I run the following code - selecting a group of 5 FITS files...

```
pro test_foreach,dummy
compile_opt IDL2
```

```
file_names = dialog_pickfile(filter="*.fit", path="E:\AstroData", /
read, /multiple_files )

print," "
for i=0,n_elements(file_names)-1 do begin
  qfile = file_names[i]
  print,strlen(qfile)
  help,qfile
  qim = readfits( qfile )
  print,strlen(qfile)
  help,qfile
endfor
end
```

All is well - here's the output:

```
IDL> test_foreach
% Compiled module: TEST_FOREACH.
        41
QFILE           STRING    = 'E:\AstroData\2010_03_0809B
\CR_Boo-045.fit'
% READFITS: Now reading 512 by 512 array
        41
QFILE           STRING    = 'E:\AstroData\2010_03_0809B
\CR_Boo-045.fit'

        41
QFILE           STRING    = 'E:\AstroData\2010_03_0809B
\CR_Boo-041.fit'
% READFITS: Now reading 512 by 512 array
        41
QFILE           STRING    = 'E:\AstroData\2010_03_0809B
\CR_Boo-041.fit'

        41
QFILE           STRING    = 'E:\AstroData\2010_03_0809B
\CR_Boo-042.fit'
% READFITS: Now reading 512 by 512 array
        41
QFILE           STRING    = 'E:\AstroData\2010_03_0809B
\CR_Boo-042.fit'

        41
QFILE           STRING    = 'E:\AstroData\2010_03_0809B
\CR_Boo-043.fit'
% READFITS: Now reading 512 by 512 array
        41
QFILE           STRING    = 'E:\AstroData\2010_03_0809B
```

```
\CR_Boo-043.fit'

        41
QFILE           STRING    = 'E:\AstroData\2010_03_0809B
\CR_Boo-044.fit'
% READFITS: Now reading 512 by 512 array
        41
QFILE           STRING    = 'E:\AstroData\2010_03_0809B
\CR_Boo-044.fit'

IDL>
```

But - if I put in a "foreach" loop following the "for - next" loop:

```
pro test_foreach,dummy
compile_opt IDL2

file_names = dialog_pickfile(filter="*.fit", path="E:\AstroData", /
read, /multiple_files )

print," "
for i=0,n_elements(file_names)-1 do begin
  qfile = file_names[i]
  print,strlen(qfile)
  help,qfile
  qim = readfits( qfile )
  print,strlen(qfile)
  help,qfile
  print," "
endfor

foreach qfile, file_names do begin
  qim = readfits( qfile )
endforeach
end
```

I get the following output:

```
IDL> test_foreach
% Compiled module: TEST_FOREACH.

        41
QFILE           STRING    = 'E:\AstroData\2010_03_0809B
\CR_Boo-045.fit'
% READFITS: Now reading 512 by 512 array
```

```
       41
QFILE           STRING    = 'E:\AstroData\2010_03_0809B
\CR_Boo-045.fit'

       41
QFILE           STRING    = 'E:\AstroData\2010_03_0809B
\CR_Boo-041.fit'
% READFITS: Now reading 512 by 512 array
       41
QFILE           STRING    = 'E:\AstroData\2010_03_0809B
\CR_Boo-041.fit'

       41
QFILE           STRING    = 'E:\AstroData\2010_03_0809B
\CR_Boo-042.fit'
% READFITS: Now reading 512 by 512 array
       41
QFILE           STRING    = 'E:\AstroData\2010_03_0809B
\CR_Boo-042.fit'

       41
QFILE           STRING    = 'E:\AstroData\2010_03_0809B
\CR_Boo-043.fit'
% READFITS: Now reading 512 by 512 array
       41
QFILE           STRING    = 'E:\AstroData\2010_03_0809B
\CR_Boo-043.fit'

       41
QFILE           STRING    = '8 unsigned int, 16 & 32 int, -32 & -64
re'
% READFITS:  ERROR - Unable to locate file 8 unsigned int, 16 & 32
int, -32 & -64 re
       41
QFILE           STRING    = '8 unsigned int, 16 & 32 int, -32 & -64
re'

% READFITS:  ERROR - Unable to locate file it
% READFITS:  ERROR - Unable to locate file it
% READFITS:  ERROR - Unable to locate file it
% READFITS:  ERROR - Unable to locate file it
% READFITS:  ERROR - Unable to locate file re
```

A couple of things to note:

1) the final time through the first loop filenames get corrupted somehow - and the string that results is actually some of the code

from "readfits.pro".

2) If I re-compile the exact same code for "test_foreach" I can get
different results -

```
IDL> test_foreach
% Compiled module: TEST_FOREACH.

        41
QFILE           STRING    = 'E:\AstroData\2010_03_0809B
\CR_Boo-045.fit'
% READFITS:  ERROR - Unable to locate file fit
        41
QFILE           STRING    = 'fit'

        41
QFILE           STRING    = 'E:\AstroData\2010_03_0809B
\CR_Boo-041.fit'
% READFITS:  ERROR - Unable to locate file fit
        41
QFILE           STRING    = 'fit'

        41
QFILE           STRING    = 'E:\AstroData\2010_03_0809B
\CR_Boo-042.fit'
% READFITS:  ERROR - Unable to locate file fit
        41
QFILE           STRING    = 'fit'

        41
QFILE           STRING    = 'E:\AstroData\2010_03_0809B
\CR_Boo-043.fit'
% READFITS:  ERROR - Unable to locate file fit
        41
QFILE           STRING    = 'fit'

        41
QFILE           STRING    = 'E:\AstroData\2010_03_0809B
\CR_Boo-044.fit'
% READFITS:  ERROR - Unable to locate file fit
        41
QFILE           STRING    = 'fit'

% READFITS:  ERROR - Unable to locate file it
% READFITS:  ERROR - Unable to locate file it
% READFITS:  ERROR - Unable to locate file it
```

% READFITS: ERROR - Unable to locate file it
% READFITS: ERROR - Unable to locate file it

And - in this case, note that in the "FOR" loop, the "help,qfile" line returns:

QFILE   STRING   = 'fit'

despite the fact that the line before that, "print,strlen(qfile)", insists the string length is 41 characters.

Sorry this is such a long post - I guess maybe I should send this to ITT VIS support? While editing I'm also getting frequent "freezes" of the IDLDE and must use the task manager in windows to exit idl - or the IDLDE just vanishes when I hit "compile". Is there any help in the help files describing the IDLDE?

Thanks again for any insight..

Mike Potter

Subject: Re: IDL 8.0 questions
Posted by wlandsman on Sun, 25 Jul 2010 11:12:03 GMT
View Forum Message <> Reply to Message

On Jul 25, 2:07 am, Mike Potter <m...@orionsound.com> wrote:

> QFILE   STRING   = '8 unsigned int, 16 & 32 int, -32 & -64

> A couple of things to note:
>
> 1) the final time through the first loop filenames get corrupted
> somehow - and the string that results is actually some of the code
> from "readfits.pro".

1. No - there is no line in readfits.pro that reads "8 unsigned int....". This looks instead to me like something typically written in the comment field of the FITS header (for the keyword BITPIX).

2. It looks like the error is occurring in the FOR-ENDFOR loop -- before the FOREACH loop is even reached! Is that right?

I have no idea what is going on but if forced to choose I would say
that this is a string descriptor problem rather than a FOREACH
problem. In particular, I would suspect that there is an IDL 8.0
Windows bug in the processing of empty strings. --Wayne

Subject: Re: IDL 8.0 questions
Posted by Mike Potter on Sun, 25 Jul 2010 14:22:32 GMT
View Forum Message <> Reply to Message

On Jul 25, 7:12 am, wlandsman <wlands...@gmail.com> wrote:
> On Jul 25, 2:07 am, Mike Potter <m...@orionsound.com> wrote:
>
>> QFILE STRING = '8 unsigned int, 16 & 32 int, -32 & -64
>> A couple of things to note:
>
>> 1) the final time through the first loop filenames get corrupted
>> somehow - and the string that results is actually some of the code
>> from "readfits.pro".
>
> 1. No - there is no line in readfits.pro that reads "8 unsigned
> int....". This looks instead to me like something typically
> written in the comment field of the FITS header (for the keyword
> BITPIX).
>
> 2. It looks like the error is occurring in the FOR-ENDFOR loop --
> before the FOREACH loop is even reached! Is that right?
>
> I have no idea what is going on but if forced to choose I would say
> that this is a string descriptor problem rather than a FOREACH
> problem. In particular, I would suspect that there is an IDL 8.0
> Windows bug in the processing of empty strings. --Wayne

Wayne:

My appologoes - I was getting sloppy! It was getting late and I was
fairly exasperated - the "bitpix 16 & 32 int " etc, just reminded me
of some of the code around line 530 in READFITS - but you're totally
right, it isn't from there - but it is also not from the image
headers. As another note - after dispensing with any "FOREACH"
statements from my code I've had no further "freezes" or spontaneous
exits of the IDLDE.

I'm running a text search through all of the files on my computer
right now - it has to be coming from somewhere. It does indeed look
like some sort of comment or piece of a help file.

Thanks!
Mike Potter

Subject: Re: IDL 8.0 questions
Posted by Mike Potter on Sun, 25 Jul 2010 15:22:52 GMT
View Forum Message <> Reply to Message

On Jul 25, 7:12 am, wlandsman <wlands...@gmail.com> wrote:
> On Jul 25, 2:07 am, Mike Potter <m...@orionsound.com> wrote:
>
>> QFILE STRING = '8 unsigned int, 16 & 32 int, -32 & -64
>> A couple of things to note:
>
>> 1) the final time through the first loop filenames get corrupted
>> somehow - and the string that results is actually some of the code
>> from "readfits.pro".
>
> 1. No - there is no line in readfits.pro that reads "8 unsigned
> int....". This looks instead to me like something typically
> written in the comment field of the FITS header (for the keyword
> BITPIX).
>
> 2. It looks like the error is occurring in the FOR-ENDFOR loop --
> before the FOREACH loop is even reached! Is that right?
>
> I have no idea what is going on but if forced to choose I would say
> that this is a string descriptor problem rather than a FOREACH
> problem. In particular, I would suspect that there is an IDL 8.0
> Windows bug in the processing of empty strings. --Wayne

Wayne - as to your second point - yes, the error shows no matter where the "FOREACH" loop is located. I tried the same code with the "FOREACH" loop first and got essentially the same results. I say "essentially" because the results do vary from consecutive running of the program, even without re-compiling. And while it doesn't happen every time it SEEMS as though the spontaneous exits of the IDLDE occur while compiling code containing a FOREACH loop. Also note I tried a FOREACH loop that interated through an array of integers and got essentially the same result.

Mike

Subject: Re: IDL 8.0 questions
Posted by penteado on Sun, 25 Jul 2010 16:27:06 GMT

View Forum Message <> Reply to Message

On Jul 25, 12:22 pm, Mike Potter <m...@orionsound.com> wrote:
> Wayne - as to your second point - yes, the error shows no matter where
> the "FOREACH" loop is located. I tried the same code with the
> "FOREACH" loop first and got essentially the same results. I say
> "essentially" because the results do vary from consecutive running of
> the program, even without re-compiling. And while it doesn't happen
> every time it SEEMS as though the spontaneous exits of the IDLDE occur
> while compiling code containing a FOREACH loop. Also note I tried a
> FOREACH loop that interated through an array of integers and got
> essentially the same result.

It does seem like a bug, so it would be good to know exactly on what conditions it happens. Two things that could be tried:

1) Run the program from the command line, to find out if it may be the effect of the Workbench.

2) Remove the calls to readfits(), to see if the problem appears when using only intrinsic routines.

I am not saying that there is something wrong with readfits(). It could just be that it is in it the action that triggers the bug. For instance, like the bug in older versions of the Workbench that caused it to crash on some calls of readcol.

Subject: Re: IDL 8.0 questions
Posted by penteado on Sun, 25 Jul 2010 17:20:26 GMT
View Forum Message <> Reply to Message

I just did some tests, and I can confirm that this is a bug, it is completely reproducible, and it is not limited to Windows. I tried on Vista 32 with similar results. In Linux 64, besides the problem as reported (the string contents getting changed), IDL crashes. Same on command line and Workbench.

This is what I used:

```
pro test_foreach
compile_opt IDL2

file_names=(file_search('*.fits'))[0:1]

for i=0,n_elements(file_names)-1 do begin
  qfile = file_names[i]
  help,qfile,i
```

```
  qim = readfits( qfile )
  help,qfile
endfor

foreach qfile, file_names,i do begin
  print,i
  qim = readfits( qfile )
endforeach

print,file_names
end
```

Then it crashes on the first call to readfits() in the foreach loop:

```
IDL> test_foreach
QFILE           STRING    = 'dec18i0001.fits'
I               LONG      =           0
% READFITS: Now reading 256 by 256 array
QFILE           STRING    = 'dec18i0001.fits'
QFILE           STRING    = 'dec18i0002.fits'
I               LONG      =           1
% READFITS: Now reading 256 by 256 array
QFILE           STRING    = 'dec18i0002.fits'
           0
% READFITS:  ERROR - Unable to locate file ts
*** glibc detected *** /usr/local/itt/idl/idl80/bin/bin.linux.x86_64/
idl: double free or corruption (fasttop): 0x0000000001ff7a80 ***
```

Even more strange, if I change the line before that to

```
print,i,' ',qfile
```

All goes as it should:

```
IDL> test_foreach
QFILE           STRING    = 'dec18i0001.fits'
I               LONG      =           0
% READFITS: Now reading 256 by 256 array
QFILE           STRING    = 'dec18i0001.fits'
QFILE           STRING    = 'dec18i0002.fits'
I               LONG      =           1
% READFITS: Now reading 256 by 256 array
QFILE           STRING    = 'dec18i0002.fits'
           0 dec18i0001.fits
% READFITS: Now reading 256 by 256 array
           1 dec18i0002.fits
% READFITS: Now reading 256 by 256 array
dec18i0001.fits dec18i0002.fits
```

In the few months I had been using the test releases for IDL 8, I never saw something like this. I tested on Tech Preview 3, and found it also crashes, though in a different way.

I am going to look into it in more detail, to see if I can tell where the value of qfile gets changed. And I am reporting this to ITTVIS.

Subject: Re: IDL 8.0 questions
Posted by Mike Potter on Sun, 25 Jul 2010 18:15:10 GMT
View Forum Message <> Reply to Message

On Jul 25, 1:20 pm, Paulo Penteado <pp.pente...@gmail.com> wrote:
> I just did some tests, and I can confirm that this is a bug, it is
> completely reproducible, and it is not limited to Windows. I tried on
> Vista 32 with similar results. In Linux 64, besides the problem as
> reported (the string contents getting changed), IDL crashes. Same on
> command line and Workbench.
>
> This is what I used:
>
> pro test_foreach
> compile_opt IDL2
>
> file_names=(file_search('*.fits'))[0:1]
>
> for i=0,n_elements(file_names)-1 do begin
> qfile = file_names[i]
> help,qfile,i
> qim = readfits(qfile)
> help,qfile
> endfor
>
> foreach qfile, file_names,i do begin
> print,i
> qim = readfits(qfile)
> endforeach
>
> print,file_names
> end
>
> Then it crashes on the first call to readfits() in the foreach loop:
>
> IDL> test_foreach
> QFILE STRING = 'dec18i0001.fits'
> I LONG = 0
> % READFITS: Now reading 256 by 256 array

> QFILE STRING = 'dec18i0001.fits'
> QFILE STRING = 'dec18i0002.fits'
> I LONG = 1
> % READFITS: Now reading 256 by 256 array
> QFILE STRING = 'dec18i0002.fits'
> 0
> % READFITS: ERROR - Unable to locate file ts
> *** glibc detected *** /usr/local/itt/idl/idl80/bin/bin.linux.x86_64/
> idl: double free or corruption (fasttop): 0x0000000001ff7a80 ***
>
> Even more strange, if I change the line before that to
>
> print,i,' ',qfile
>
> All goes as it should:
>
> IDL> test_foreach
> QFILE STRING = 'dec18i0001.fits'
> I LONG = 0
> % READFITS: Now reading 256 by 256 array
> QFILE STRING = 'dec18i0001.fits'
> QFILE STRING = 'dec18i0002.fits'
> I LONG = 1
> % READFITS: Now reading 256 by 256 array
> QFILE STRING = 'dec18i0002.fits'
> 0 dec18i0001.fits
> % READFITS: Now reading 256 by 256 array
> 1 dec18i0002.fits
> % READFITS: Now reading 256 by 256 array
> dec18i0001.fits dec18i0002.fits
>
> In the few months I had been using the test releases for IDL 8, I
> never saw something like this. I tested on Tech Preview 3, and found
> it also crashes, though in a different way.
>
> I am going to look into it in more detail, to see if I can tell where
> the value of qfile gets changed. And I am reporting this to ITTVIS.

Paulo:

Thanks for the confirmation. You know, I've been trying the things you suggested and in the past hour or so have been unable to get it to misbehave! That is, it's working fine right now. I have no clue as to why. But note that in the line "% READFITS: ERROR - Unable to locate file ts" note that, like in my case, the two characters remaining are the last two in the file name. My files are ".fit" so I was getting "it" as the remaining characters after going to

readfits().

Mike Potter

Subject: Re: IDL 8.0 questions
Posted by penteado on Sun, 25 Jul 2010 19:11:18 GMT
View Forum Message <> Reply to Message

On Jul 25, 3:15 pm, Mike Potter <m...@orionsound.com> wrote:
> Thanks for the confirmation. You know, I've been trying the things
> you suggested and in the past hour or so have been unable to get it to
> misbehave! That is, it's working fine right now. I have no clue as
> to why. But note that in the line "% READFITS: ERROR - Unable to
> locate file ts" note that, like in my case, the two characters
> remaining are the last two in the file name. My files are ".fit" so I
> was getting "it" as the remaining characters after going to
> readfits().

Indeed. And I confirmed it is not readfits()'s fault, as I found where filename gets changed. In my version, it is in line 251 of readfits:

ext = strlowcase(strmid(filename,len-3,3))

I found it to be strmid that causes the problem. When I put a breakpoint in that line:

```
% Breakpoint at: READFITS         251 /software/idl/idlastro/pro/
readfits.pro
IDL> help,filename
FILENAME       STRING    = 'dec18i0001.fits'
IDL> help,filename,strmid(filename,len-3,3)
FILENAME       STRING    = 'its'
<Expression>   STRING    = 'its'
```

So I made a much simpler test case that manifests the bug:

```
IDL> s='abc_'+strtrim(sindgen(2),2)
IDL> for i=0,n_elements(s)-1 do help,i,s[i],strmid(s[i],0,1)
I              INT       =        0
<Expression>   STRING    = 'abc_0'
<Expression>   STRING    = 'a'
I              INT       =        1
<Expression>   STRING    = 'abc_1'
<Expression>   STRING    = 'a'
```

So far, all is well. But:

```
IDL> foreach el,s,i do help,i,el,strmid(el,0,1)
I               LONG      =           0
EL              STRING    = 'a'
<Expression>    STRING    = 'a'
I               LONG      =           1
EL              STRING    = 'a'
<Expression>    STRING    = 'a'
```

So el is getting changed, it is receiving the output of strmid.

To make things more confusing, a print to el makes the problem go away:

```
IDL> foreach el,s,i do begin & print,i,':',el & help,i,strmid(el,
0,1),el & endforeach
          0:abc_0
I               LONG      =           0
<Expression>    STRING    = 'a'
EL              STRING    = 'abc_0'
          1:abc_1
I               LONG      =           1
<Expression>    STRING    = 'a'
EL              STRING    = 'abc_1'
```

And to show that is is not just a case of strmid putting its output into its input variable (which could not happen in the array element in the for loop, since it is passed by value):

```
IDL> a='abcd'
IDL> help,strmid(a,0,1)
<Expression>    STRING    = 'a'
IDL> help,a
A               STRING    = 'abcd'
```

Maybe el is getting one of its metadata fields wrong when it is created?

Subject: Re: IDL 8.0 questions
Posted by David Fanning on Sun, 25 Jul 2010 19:33:59 GMT
View Forum Message <> Reply to Message

Paulo Penteado writes:

> Maybe el is getting one of its metadata fields wrong when it is
> created?

Good detective work, Paulo! :-)

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: IDL 8.0 questions
Posted by penteado on Mon, 26 Jul 2010 15:57:53 GMT
View Forum Message <> Reply to Message

On Jul 25, 4:11 pm, Paulo Penteado <pp.pente...@gmail.com> wrote:
> Maybe el is getting one of its metadata fields wrong when it is
> created?

I was curious about this, so I checked what flags were being set in a foreach element variable. I found that when just the loop is used, the variable has no flags. When a print is used inside the loop, the print call sets the IDL_V_DYNAMIC flag. Which, I thought should have been always set, as the element is a string.
