

---

Subject: Re: Error: Array has a corrupted descriptor  
Posted by [David Fanning](#) on Tue, 28 Nov 2006 03:24:55 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Gongqin Shen writes:

```
> I have this problem when I have to initialized a same-name array
> multiple times. It goes like this:
> FOR i = 1, N_ELEMENTS(list) - 1 DO BEGIN
>   arr = BYTARR(list[i], list[i-1])
>   Do Something here....
> END
>
> After a few, somewhere around 5, runs, the program crashes and reports
> this error:
> ERROR: Array has a corrupted descriptor.
>
> I searched through the archives, and there is one thread about DICOM
> image disposal addressed the similar question. I am wondering is there
> anybody who knows how to get around this?
```

This is usually a LHS problem. I think the secret lies  
in the "Do Something here..." part of the code. :-)

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: Error: Array has a corrupted descriptor  
Posted by [gqshen2008@gmail.com](#) on Tue, 28 Nov 2006 15:49:17 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

David,

Thank you for paying attention to my question. Bascially, "do  
something here" is just some basic array manipulation like:

```
arr = arr3D[1:2, *]
arrInd[0, *] -= a
arrInd[1, *] -= b
arr[arrInd] = 1B
aLabeled = Label_Region(arr)
```

```
FOR j = 1, MAX(aLabeled) DO BEGIN
```

```
ithIdx = (WHERE(aLabeled EQ j))[0]
outputList = [outputList, ithIdx]
ENDFOR
```

I hope your eagle eye can find out where the glitch comes from. :-).

Really Grateful,

Gongqin

David Fanning wrote:

> Gongqin Shen writes:

>

>> I have this problem when I have to initialize a same-name array

>> multiple times. It goes like this:

>> FOR i = 1, N\_ELEMENTS(list) - 1 DO BEGIN

>> arr = BYTARR(list[i], list[i-1])

>> Do Something here....

>> END

>>

>> After a few, somewhere around 5, runs, the program crashes and reports

>> this error:

>> ERROR: Array has a corrupted descriptor.

>>

>> I searched through the archives, and there is one thread about DICOM

>> image disposal addressed the similar question. I am wondering is there

>> anybody who knows how to get around this?

>

> This is usually a LHS problem. I think the secret lies

> in the "Do Something here..." part of the code. :-)

>

> 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: Error: Array has a corrupted descriptor

Posted by [David Fanning](#) on Tue, 28 Nov 2006 16:06:11 GMT

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

---

Gongqin Shen writes:

> Thank you for paying attention to my question. Basically, "do

> something here" is just some basic array manipulation like:

> arr = arr3D[1:2, \*]

```
> arrInd[0, *] -= a
> arrInd[1, *] -= b
> arr[arrInd] = 1B
> aLabeled = Label_Region(arr)
>
> FOR j = 1, MAX(aLabeled) DO BEGIN
>   ithIdx = (WHERE(aLabeled EQ j))[0]
>   outputList = [outputList, ithIdx]
> ENDFOR
>
> I hope your eagle eye can find out where the glitch comes from. :-).
```

I think the likely suspect is this line:

```
outputList = [outputList, ithIdx]
```

At the *very* least, I would write this line like this:

```
outputList = [Temporary(outputList), ithIdx]
```

But I think you might be better off making output list as big as you need it (or bigger) and then filling it. I suspect the constant recreating of this array is fragmenting memory like crazy and resulting in your problems.

```
void = Where(aLabeled GT 0, count)
outputList = Lindgen(count)
```

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: Error: Array has a corrupted descriptor  
Posted by [Karl Schultz](#) on Tue, 28 Nov 2006 17:12:57 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

On Tue, 28 Nov 2006 09:06:11 -0700, David Fanning wrote:

```
> Gongqin Shen writes:
```

```
>
```

```
>> Thank you for paying attention to my question. Bascially, "do
```

```

>> something here" is just some basic array manipulation like:
>> arr = arr3D[1:2, *]
>> arrInd[0, *] -= a
>> arrInd[1, *] -= b
>> arr[arrInd] = 1B
>> aLabeled = Label_Region(arr)
>>
>> FOR j = 1, MAX(aLabeled) DO BEGIN
>>   ithIdx = (WHERE(aLabeled EQ j))[0]
>>   outputList = [outputList, ithIdx]
>> ENDFOR
>>
>> I hope your eagle eye can find out where the glitch comes from. :-).
>
> I think the likely suspect is this line:
>
>   outputList = [outputList, ithIdx]
>
> At the *very* least, I would write this line like this:
>
>   outputList = [Temporary(outputList), ithIdx]
>
> But I think you might be better off making output list as big
> as you need it (or bigger) and then filling it. I suspect
> the constant recreating of this array is fragmenting memory
> like crazy and resulting in your problems.
>
>   void = Where(aLabeled GT 0, count)
>   outputList = Lindgen(count)
>
> Cheers,
>
> David

```

The corrupted array descriptor error message is almost always caused by a programming error in an IDL system routine or user-written DLM function. You generally cannot cause this error by just moving array elements around since IDL performs bounds checking. This code is also pretty mature and we would have fixed any problems by now. Likewise, a storage fragmentation issue that leads the inability to allocate a large block of storage would result in a message about storage allocation, not a corrupted array descriptor.

Unless there is a user-written DLM in the mix, I suspect a bug in LABEL\_REGION. It would be really good if Gongqin could create a reproduce case and get it to ITTVIS Tech Support.

Karl

---



---

Subject: Re: Error: Array has a corrupted descriptor  
Posted by [gqshen2008@gmail.com](mailto:gqshen2008@gmail.com) on Tue, 28 Nov 2006 17:16:48 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

I agree, that does make sense since IDL must use MALLOC routine internally (I assume IDL is written in C) to claim the memory any time the size of array changes, and constantly changing it will fragment the memory like crazy as you said. However, I have modified my code to get rid of the BYTARR line and keep the dynamically modifying array line, and the program doesn't crash now.

I am really confused here and hope someone can be the myth-buster.

P.S. IDL gives me the impression that it is not a big fan of the idea of constantly changing a complex structure like array or an unsigned long which are over 1,000,000 since it will screw up the content of the variable, sometimes. My suggestion is to make a temporary variable to avert the burden. I would like to hear a better idea to deal with this kind of situation.

Best regards,  
Gongqin

David Fanning wrote:

> Gongqin Shen writes:

>

>> Thank you for paying attention to my question. Basically, "do something here" is just some basic array manipulation like:

>> arr = arr3D[1:2, \*]

>> arrInd[0, \*] -= a

>> arrInd[1, \*] -= b

>> arr[arrInd] = 1B

>> aLabeled = Label\_Region(arr)

>>

>> FOR j = 1, MAX(aLabeled) DO BEGIN

>>   ithIdx = (WHERE(aLabeled EQ j))[0]

>>   outputList = [outputList, ithIdx]

>> ENDFOR

>>

>> I hope your eagle eye can find out where the glitch comes from. :-).

>

> I think the likely suspect is this line:

>

>   outputList = [outputList, ithIdx]

>

> At the \*very\* least, I would write this line like this:

>

>   outputList = [Temporary(outputList), ithIdx]

>

> But I think you might be better off making output list as big

> as you need it (or bigger) and then filling it. I suspect  
> the constant recreating of this array is fragmenting memory  
> like crazy and resulting in your problems.  
>  
> void = Where(aLabeled GT 0, count)  
> outputList = Lindgen(count)  
>  
> 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: Error: Array has a corrupted descriptor  
Posted by [gqshen2008@gmail.com](mailto:gqshen2008@gmail.com) on Tue, 28 Nov 2006 17:20:10 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Karl,

Thank you for your comment and I can make a zip file containing only the pertaining code and it should not be too complicated. Which email address I should send to? The one you listed here?

Much appreciated.

Gongqin

Karl Schultz wrote:

> On Tue, 28 Nov 2006 09:06:11 -0700, David Fanning wrote:

>

>> Gongqin Shen writes:

>>

>>> Thank you for paying attention to my question. Bascially, "do something here" is just some basic array manipulation like:

>>> arr = arr3D[1:2, \*]

>>> arrInd[0, \*] -= a

>>> arrInd[1, \*] -= b

>>> arr[arrInd] = 1B

>>> aLabeled = Label\_Region(arr)

>>>

>>> FOR j = 1, MAX(aLabeled) DO BEGIN

>>> ithIdx = (WHERE(aLabeled EQ j))[0]

>>> outputList = [outputList, ithIdx]

>>> ENDFOR

>>>

>>> I hope your eagle eye can find out where the glitch comes from. :-).

```
>>
>> I think the likely suspect is this line:
>>
>>   outputList = [outputList, ithIdx]
>>
>> At the *very* least, I would write this line like this:
>>
>>   outputList = [Temporary(outputList), ithIdx]
>>
>> But I think you might be better off making output list as big
>> as you need it (or bigger) and then filling it. I suspect
>> the constant recreating of this array is fragmenting memory
>> like crazy and resulting in your problems.
>>
>>   void = Where(aLabeled GT 0, count)
>>   outputList = Lindgen(count)
>>
>> Cheers,
>>
>> David
>
> The corrupted array descriptor error message is almost always caused by a
> programming error in an IDL system routine or user-written DLM function.
> You generally cannot cause this error by just moving array elements
> around since IDL performs bounds checking. This code is also pretty mature
> and we would have fixed any problems by now. Likewise, a storage
> fragmentation issue that leads the inability to allocate a large block of
> storage would result in a message about storage allocation, not a
> corrupted array descriptor.
>
> Unless there is a user-written DLM in the mix, I suspect a bug in
> LABEL_REGION. It would be really good if Gongqin could create a reproduce
> case and get it to ITTVIS Tech Support.
>
> Karl
```

---

---

Subject: Re: Error: Array has a corrupted descriptor  
Posted by [David Fanning](#) on Tue, 28 Nov 2006 17:20:54 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Karl Schultz writes:

```
> The corrupted array descriptor error message is almost always caused by a
> programming error in an IDL system routine or user-written DLM function.
> You generally cannot cause this error by just moving array elements
> around since IDL performs bounds checking. This code is also pretty mature
> and we would have fixed any problems by now. Likewise, a storage
```

- > fragmentation issue that leads the inability to allocate a large block of
- > storage would result in a message about storage allocation, not a
- > corrupted array descriptor.
- >
- > Unless there is a user-written DLM in the mix, I suspect a bug in
- > LABEL\_REGION. It would be really good if Gongqin could create a reproduce
- > case and get it to ITTVIS Tech Support.

You are probably right. I think I was thinking about the  
"temporary variables still checked out" error.

And while they have the hood open, maybe they can fix the  
boundary problems with this function. :-)

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: Error: Array has a corrupted descriptor  
Posted by [David Fanning](#) on Tue, 28 Nov 2006 17:24:50 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Gongqin Shen writes:

- > I am really confused here and hope someone can be the myth-buster.

You are hoping for WAY too much. We are working with  
computers here. Confusion is pretty much a way of life  
in this field.

- > P.S. IDL gives me the impression that it is not a big fan of the idea
- > of constantly changing a complex structure like array or an unsigned
- > long which are over 1,000,000 since it will screw up the content of the
- > variable, sometimes. My suggestion is to make a temporary variable to
- > avert the burden. I would like to hear a better idea to deal with this
- > kind of situation.

I've never run into anything remotely like this. I'd have  
to see it to believe it. :-)

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: Error: Array has a corrupted descriptor

Posted by [Karl Schultz](#) on Tue, 28 Nov 2006 17:49:24 GMT

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

---

On Tue, 28 Nov 2006 09:16:48 -0800, Gongqin Shen wrote:

- > I agree, that does make sense since IDL must use MALLOC routine
- > internally (I assume IDL is written in C) to claim the memory any time
- > the size of array changes, and constantly changing it will fragment the
- > memory like crazy as you said. However, I have modified my code to get
- > rid of the BYTARR line and keep the dynamically modifying array line,
- > and the program doesn't crash now.
- > I am really confused here and hope someone can be the myth-buster.

This is surprising. My best guess at the moment is that you reduced the memory fragmentation problem which allows LABEL\_REGION to proceed without encountering storage allocation errors. LABEL\_REGION may still have a bug when recovering from a storage allocation problem that resulted in the corrupted array descriptor. The code paths (yes, written in C) that recover from storage allocation problems are a bit hard to exercise under various storage allocation conditions.

It would be best if you could wrap up your (failing) code and data and send it to support at [itvvis dot com](mailto:itvvis@dot.com).

---