
Subject: Re: strictarrsubs in IDL 8.1

Posted by [kdpenner](#) on Fri, 17 May 2013 23:17:00 GMT

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

On Friday, May 17, 2013 3:50:39 PM UTC-7, kdpe...@gmail.com wrote:

> Hi,

>

>

>

> I have the following in my .idlstartup:

>

>

>

> compile_opt strictarrsubs

>

>

>

Grr. WHERE returns an array when the condition is met, and when it's not, it returns -1 as an integer. strictarrsubs only catches bad subscripts when passed in arrays, not integers.

How unpleasant.

> However, I can subscript arrays in 8.1 with [-1], for instance. If I remember correctly, the compile_opt statement in earlier versions of IDL meant I would get an error for a [-1] subscript. Am I misremembering, or has something changed?

>

>

>

> Thanks,

>

>

>

> Kyle

Subject: Re: strictarrsubs in IDL 8.1

Posted by [Jeremy Bailin](#) on Mon, 20 May 2013 15:30:28 GMT

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

On 5/17/13 7:17 PM, kdpenner@gmail.com wrote:

> On Friday, May 17, 2013 3:50:39 PM UTC-7, kdpe...@gmail.com wrote:

>> Hi,

>>

>>

>>

>> I have the following in my .idlstartup:

>>

>>

```
>>
>> compile_opt strictarrsubs
>>
>>
>>
> Grr. WHERE returns an array when the condition is met, and when it's not, it returns -1 as an
integer. strictarrsubs only catches bad subscripts when passed in arrays, not integers.
>
> How unpleasant.
>
>> However, I can subscript arrays in 8.1 with [-1], for instance. If I remember correctly, the
compile_opt statement in earlier versions of IDL meant I would get an error for a [-1] subscript.
Am I misremembering, or has something changed?
>>
>>
>>
>> Thanks,
>>
>>
>>
>> Kyle
>
```

That's because for-reasons-that-still-baffle-me, subscripting with a scalar negative index has a new legitimate meaning in IDL8 (counting from the end instead).

-Jeremy.

Subject: Re: strictarrsubs in IDL 8.1
Posted by [Chip Helms](#) on Mon, 20 May 2013 18:02:54 GMT
[View Forum Message](#) <> [Reply to Message](#)

I use the /NULL keyword in WHERE to avoid that issue. Although the only way to be completely safe is to check the count before using the results of WHERE.
