Subject: Minor obscure error message
Posted by wlandsman on Tue, 01 Sep 2015 19:12:33 GMT

View Forum Message <> Reply to Message

Suppose one wants to write !p.thick=2 but forgets the equals sign

```
IDL> !p.thick2
% Tag name THICK2 is undefined for structure !PLT.
% Execution halted at: $MAIN$
```

and one gets a very logical error message.

But if one precedes this with the syntactically correct statement !p.background=0

```
IDL> !p.background=0 & !p.thick2
% Object reference type required in this context: <STRUCT    Array[1]>.
% Execution halted at: $MAIN$
```

one gets an obscure reference about requiring an object type.   I'm not sure how IDL is getting confused here.   It only happens when the two statements are on the same line.

```
IDL> print,!version
{ x86_64 linux unix linux 8.4 Sep 27 2014      64      64}
```

--Wayne

Subject: Re: Minor obscure error message
Posted by Dick Jackson on Tue, 01 Sep 2015 21:50:05 GMT

View Forum Message <> Reply to Message

Hi Wayne,

I think this is why: if there is an "&" indicating multiple statements, then "implied print" is not attempted at all on any of them. In your first example, IDL tried to interpret it as a method call on !p (and failed, with no message), then tried to interpret it as an implied print, and failed. Second time, IDL tried as a method call on !p, and that failed. Due to the "&", it did not try implied print, so it reported the first failure.

Here's a simpler example to show a similar effect:

```
IDL> !Pi
      3.1415927
IDL> !Pi & !DPi
% Attempt to call undefined procedure: '!PI'.
```

Cheers,

-Dick

Dick Jackson Software Consulting Inc.
Victoria, BC, Canada --- http://www.d-jackson.com

On Tuesday, 1 September 2015 12:12:35 UTC-7, wlandsman wrote:
> Suppose one wants to write !p.thick=2 but forgets the equals sign
>
> IDL> !p.thick2
> % Tag name THICK2 is undefined for structure !PLT.
> % Execution halted at: $MAIN$
>
> and one gets a very logical error message.
>
> But if one precedes this with the syntactically correct statement !p.background=0
>
> IDL> !p.background=0 & !p.thick2
> % Object reference type required in this context: <STRUCT   Array[1]>.
> % Execution halted at: $MAIN$
>
> one gets an obscure reference about requiring an object type.   I'm not sure how IDL is getting confused here.   It only happens when the two statements are on the same line.
>
> IDL> print,!version
> { x86_64 linux unix linux 8.4 Sep 27 2014   64   64}
>
>
> --Wayne

Subject: Re: Minor obscure error message
Posted by wlandsman on Wed, 02 Sep 2015 02:01:08 GMT

View Forum Message <> Reply to Message

Dick,

Thanks for the explanation!   I was still confused for a bit about why IDL does not know that !P is a structure, and instead tries to interpret it as an object.   The answer is likely that

IDL> structure.tag

can never be a valid syntax (if you don't allow implied print), but

IDL> object.method

can be a valid syntax.   --Wayne

On Tuesday, September 1, 2015 at 5:50:09 PM UTC-4, Dick Jackson wrote:

> Hi Wayne,
>
> I think this is why: if there is an "&" indicating multiple statements, then "implied print" is not attempted at all on any of them. In your first example, IDL tried to interpret it as a method call on !p (and failed, with no message), then tried to interpret it as an implied print, and failed. Second time, IDL tried as a method call on !p, and that failed. Due to the "&", it did not try implied print, so it reported the first failure.
>
> Here's a simpler example to show a similar effect:
>
> IDL> !Pi
>   3.1415927
> IDL> !Pi & !DPi
> % Attempt to call undefined procedure: '!PI'.
>
> Cheers,
> -Dick
>
> Dick Jackson Software Consulting Inc.
> Victoria, BC, Canada --- http://www.d-jackson.com
>
> On Tuesday, 1 September 2015 12:12:35 UTC-7, wlandsman wrote:
>> Suppose one wants to write !p.thick=2 but forgets the equals sign
>>
>> IDL> !p.thick2
>> % Tag name THICK2 is undefined for structure !PLT.
>> % Execution halted at: $MAIN$
>>
>> and one gets a very logical error message.
>>
>> But if one precedes this with the syntactically correct statement !p.background=0
>>
>> IDL> !p.background=0 & !p.thick2
>> % Object reference type required in this context: <STRUCT   Array[1]>.
>> % Execution halted at: $MAIN$
>>
>> one gets an obscure reference about requiring an object type.   I'm not sure how IDL is getting confused here.   It only happens when the two statements are on the same line.
>>
>> IDL> print,!version
>> { x86_64 linux unix linux 8.4 Sep 27 2014   64   64}
>>
>>
>> --Wayne

Subject: Re: Minor obscure error message

Posted by Yngvar Larsen on Wed, 02 Sep 2015 07:36:37 GMT
View Forum Message <> Reply to Message

On Tuesday, 1 September 2015 23:50:09 UTC+2, Dick Jackson wrote:
> Hi Wayne,
>
> I think this is why: if there is an "&" indicating multiple statements, then "implied print" is not attempted at all on any of them.

Apparently you are right. But then the question is: why not? Is there a compelling reason why expressions on a single line separated by "&" can't treated separately, with implied print where relevant? E.g. something like this:

```
IDL> !pi & a = 1 & !dpi & b = 0
      3.1415927
      3.1415926535897931
IDL> a, b
      1
      0
```

--
Yngvar