Subject: warning: (all(part1))(part2) = part2*0.
Posted by R. Bauer on Thu, 04 Jun 1998 07:00:00 GMT
View Forum Message <> Reply to Message

This will gave an error like:

```
print,(all(part1))(part2)
      0.00000      1.00000      2.00000
% Temporary variables are still checked out - cleaning up...
```

I know there are many solutions to get around of this malfunctional
code.
I was really surprising me that's it gaves this message.

```
PRO ind_err

all = findgen(100)
part1 = findgen(10)
part2 = findgen(3)

(all(part1))(part2) = part2*0.

print,(all(part1))(part2)


END
```

--
R.Bauer

Institut fuer Stratosphaerische Chemie (ICG-1)
Forschungszentrum Juelich
email: R.Bauer@fz-juelich.de

Subject: Re: warning: (all(part1))(part2) = part2*0.
Posted by R. Bauer on Fri, 05 Jun 1998 07:00:00 GMT
View Forum Message <> Reply to Message

Stein Vidar Hagfors Haugan wrote:

> R. Bauer wrote:
>
>> (all(part1))(part2) = part2*0.

>
> What you're doing here is to assign values to parts of a
> temporary value, almost as if you're saying
>
>   5*all(part1) = 1
>
> since (all(part1)) is a temporary value (actually, an rvalue)
> like any other expression.
>
> I think the error here is that IDL's compiler does not protest,
> but instead generates code (seemingly) to do assignments to
> a temporary value.
>

On idl 4 the compiler is protesting if something like above is defined.I like to have these feature back again.

Are there other differences known in compiler errors which won't be detected itself by idl 5.x ?

> What you should do (with the intended (?) effect) is:
>
>   all(part1(part2)) = part2*0

We have done it in the same way.

--
R.Bauer

Institut fuer Stratosphaerische Chemie (ICG-1)
Forschungszentrum Juelich
email: R.Bauer@fz-juelich.de

Subject: Re: warning: (all(part1))(part2) = part2*0.
Posted by steinhh on Fri, 05 Jun 1998 07:00:00 GMT
View Forum Message <> Reply to Message

R. Bauer wrote:

> (all(part1))(part2) = part2*0.

What you're doing here is to assign values to parts of a
temporary value, almost as if you're saying

5*all(part1) = 1

since (all(part1)) is a temporary value (actually, an rvalue)
like any other expression.

I think the error here is that IDL's compiler does not protest,
but instead generates code (seemingly) to do assignments to
a temporary value.

What you should do (with the intended (?) effect) is:

all(part1(part2)) = part2*0

Regards,

Stein Vidar
