
Subject: Re: Operator precedence flipflop?
Posted by [penteado](#) on Sat, 02 Jun 2012 16:32:46 GMT
[View Forum Message](#) <> [Reply to Message](#)

That makes sense. Since there were no objects involved, I did not think of that.

This seems to confirm it:

IDL Version 8.1 (linux x86_64 m64). (c) 2011, ITT Visual Information Solutions

```
IDL> s={str:[ptr_new(['1','2']),ptr_new(['a','b'])]}
IDL> print,*s.str(0)
% Expression must be a scalar in this context: S.
% Execution halted at: $MAIN$
IDL> print,*s->str(0)
% Expression must be a scalar in this context: S.
% Execution halted at: $MAIN$
IDL> print,*(s->str(0))
1 2
```

IDL Version 8.2 (linux x86_64 m64). (c) 2012, Exelis Visual Information Solutions, Inc.

```
IDL> s={str:[ptr_new(['1','2']),ptr_new(['a','b'])]}
IDL> print,*s.str(0)
1 2
IDL> print,*s->str(0)
% Expression must be a scalar in this context: S.
% Execution halted at: $MAIN$
IDL> print,*(s->str(0))
% Object reference type required in this context: S.
% Execution halted at: $MAIN$
```

(groups.google.com/group/comp.lang.idl-pvwave/browse_thread/thread/fe6ac2540db82f3b)

On Jun 2, 12:02 pm, fawltylangu...@gmail.com wrote:

> On Saturday, June 2, 2012 12:23:48 AM UTC+2, Paulo Penteado wrote:
> My guess is that the cause is the dot operator ambiguity introduced in IDL8. Probably `s.str(0)` in `*s.str(0)` is interpreted as a method call for which `s` must be a scalar. But I can not check, I could not find a list of fixed bugs in IDL 8.2.