
Subject: Re: Modifying Arrays and Structures in HASH's (hint: you can't)

Posted by [JDS](#) on Fri, 23 Mar 2012 23:07:00 GMT

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

> The problems shows up in structures and array indices if they are used in the way written above:

>

> IDL> print,(h['a'])[1]

> -1

> IDL> (h['a'])[1]=99

> % Expression must be named variable in this context: <INT Array[3]>.

> % Execution halted at: \$MAIN\$

>

> And that is all fault of the parser's weird use of values instead of references on qualified names. Which, for that reason, and for breaking the least surprise principle, should be changed in a future version (a compile_opt, or a new file extension, to keep compatibility).

>

> But I think that the multiple indices for arrays approach could easily be applied to structures, just making a small change to the overloadbracketsleftside method. I may write a small derived class for that purpose. In that way, it would be possible to do

>

> h['a','b']=2

Interesting. I wonder if this would this be fully as flexible as the structure dereference, which can retrieve and assign arrays:

```
IDL> a=replicate({b:randomu(sd)},100)
```

```
IDL> help,a.b
```

```
<Expression>  FLOAT  = Array[100]
```

```
IDL> a.b=findgen(100)
```

It would need to do so for the POLS. It certainly loses points for lack of semantic consistency. And it creates another more extreme category of, e.g. the "structure member variables cannot be passed by reference" rule.

Thanks,

JD
