

---

Subject: Re: Still missing features in IDL 8  
Posted by [JDS](#) on Sat, 06 Nov 2010 19:03:31 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

On Nov 4, 5:40 pm, Paulo Penteado <pp.pente...@gmail.com> wrote:  
> On Nov 4, 9:29 pm, JD Smith <jdtsmith.nos...@yahoo.com> wrote:  
>  
>> Lists and hashes, while very welcome (!), are somewhat cumbersome to  
>> use with out some syntactic sugar along the lines of what has been  
>> discussed. In addition to:  
>  
>> list[1,2,3]  
>  
>> for a list/array containing another list/array, containing another  
>> list/array, we need, e.g.:  
>  
>> list[3,'foo',2:5]  
>  
>> for a list containing a hash or structure, containing an array or  
>> list.  
>  
> That is just what I was doing.  
>  
>> Currently you must use  
>  
>> (list[3])['foo']  
>  
>> A better syntax, if it could be arranged, would be:  
>  
>> list[1][2]  
>  
>> or  
>  
>> list[3]['foo']  
>  
>> Even better if these could function correctly on the LHS of an  
>> assignment.  
>  
> I agree that way would be better, not just because it would not mess  
> with the 8D limits. But that would require changes to the language,  
> for the new syntax, and a very different way for the overloadbrackets  
> method to work.

I don't see how 8D limits come in. The limits are the number of possible arguments to the `_OVERLOADBRACKETSLEFTSIDE` method, which is much larger than 8. Unfortunately, the proposed method is deeply flawed, as there is a unavoidable ambiguity between array and list indices. Consider:

```
IDL> la=objarr(2,2)
IDL> for i=1b,4 do la[i-1]=list([i,2*i^2],100b+[i,2*i^2])
IDL> for i=0,3 do print,la[i]
  1      2
101    102
  2      8
102    108
  3     18
103    118
  4     32
104    132
```

Now consider the proposed syntax: `la[0,1,1]`

Which indices apply to which entity? I.e., is this:

```
IDL> print,(la[0,1])[1]
 103    118
```

or

```
IDL> print,((la[0])[1])[1]
 102
```

One option would be to say that arrays of lists gobble up as many indices as they have dimensions, but that is unnecessarily limiting and unintuitive, since you might like to index them using a single scalar as I have. You might also say each list gobbles only only a single index.

This also point out how the other syntax is superior:

`la[0,1][1]` vs. `la[0][1][1]`

would be different things. But I agree, this is not easy to accommodate at this point.

A few more comments on LIST and HASH:

We need LISTARR and HASHARR to go with OBJARR.

You should be able to assign a single scalar variable to "hash slices" to set each of the specified keys to that same value, just as you can assign a single value to a subset of lists:

```
IDL> a=hash({one:1,two:2})
```

```
IDL> a[ ['one','two'] ] = 4 ;; this should work!  
% Key and Value must have the same number of elements.  
% Error occurred at: HASH::_OVERLOADBRACKETSLEFTSIDE  
%  
% $MAIN$  
% Execution halted at: $MAIN$
```

JD

---