
Subject: modern syntax

Posted by [greg.addr](#) on Wed, 18 Nov 2015 11:47:10 GMT

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

> If you're running 8.4 or later, I encourage more modern syntax, for example
> IDL> x='Run0'+variable1.toString()+'_Seq0001_Scan'+variable2.tost ring()
> IDL> x
> Run012345_Seq0001_Scan345
>
> The .toString() static method on IDL_Variable types implies trimming of both leading and trailing
blanks.
>
> For more on static methods, see http://www.exelisvis.com/docs/IDL_Variable.html
>
> Jim P.

I've been wondering about this recently. I admit it's rare that I use any language but IDL, so I'm sure I'm behind the times. Consistency is a good thing in programming, so I'm uneasy with a pick and mix approach to the new syntax: it can only leave code less readable. I understand the sense of the object approach, so I'd go with the new, but the notation - to my eyes - is getting worse:

variable1.toString() vs string(variable1)

- the parentheses have become a relict, symbolically enclosing an argument which has gone elsewhere. To compensate the loss of meaning, we need to tack on a new preposition, 'to'. Ugh.

In some cases, though, the new syntax is elegant:

variable.tname vs size(variable,/tname)

although the old notation was unnecessarily blighted by the need to access this through the size function. I haven't yet grasped why variable.tname doesn't need (or allow) the following parentheses, since it appears to me to return a value and therefore qualify as a function. But I like it better without.

```
IDL> a={b:0,tname:4}
```

```
IDL> a.tname
```

```
4
```

```
IDL> (a).tname
```

```
% Object reference type required in this context: A.
```

I don't know what to do here, though. Isn't 'a' also an object now?

Then there's the grotesque,

```
PRINT, var1.Equals(var2)
```

I don't want to write that.

I'd be interested to know how others choose which to use.

cheers,
Greg
