
Subject: Re: Object programming with data...
Posted by [merlecorp](#) on Wed, 22 May 2002 21:52:38 GMT
[View Forum Message](#) <> [Reply to Message](#)

Craig Markwardt <craigmnet@cow.physics.wisc.edu> wrote in message
news:<onptzrj0zd.fsf@cow.physics.wisc.edu>...

> Randall Skelton <rskelto@atm.ox.ac.uk> writes:
>> On a slightly different topic, is it possible to define a function that
>> takes an arbitrary number of parameters? i.e. how do I write a function
>> 'sum' that takes 'n' variables and sums them? (yes, in this case I could
>> use 'total' but that's not the point...)
>
> No fair slipping this at the end of an unrelated post! I usually
> don't read David's "gosh golly" articles :-)
>
> The answer to your question is no, and yes. No, there is no construct
> in IDL that makes handling an arbitrary number of arguments easy. On
> the other hand, yes, it is possible to parse them if you specify all
> the parameters explicitly, as in,
>
> PRO MYTOTAL, X1, X2, X3, X4, X5, X6, X7, X8, X9, X10
>
> and so on up to the maximum of 64 (?). Then you access them using the
> EXECUTE() function.
>
> for i = 0, n_params()-1 do begin
> dummy = execute('x = x + x'+strtrim(i,2))
> endfor
>
>
> Craig

Hey,

Here's another approach using the `_extra` keyword. It's not my idea--I
pilfered it from R. Kling's InformationPanel--but I liked his approach
to arbitrary inputs & thought it was worth posting in this simplified
form.

One catch ... I'm not sure if there are limits on the number of
parameters accepted by the `_extra` keyword.

cheers,
merle

; -----

```
FUNCTION Sum, _extra=extra
; -- borrowed this approach from R. Kling's InformationPanel.pro

nNumEntries = N_Tags(extra)

nSum = 0
FOR i = 0, nNumEntries-1 DO nSum = nSum + extra.(i)

Return, nSum

END
```

```
; -----
PRO eg
; -- example calling program

x1 = 10
x2 = 20
x3 = 30
x4 = 40
x5 = 50

Print, Sum(x1=x1, x2=x2, x3=x3)
Print, Sum(x1=x1, x2=x2, x3=x3, x4=x4)
Print, Sum(x1=x1, x2=x2, x3=x3, x4=x4, x5=x5)

END
```
