Subject: Re: Checking datatypes (datatype.pro)
Posted by meron on Thu, 13 Jul 1995 07:00:00 GMT
View Forum Message <> Reply to Message

In article <Pine.SOL.3.91.950713122559.14751A-100000@chroma>, Russ Welti
<rwelti@chroma.mbt.washington.edu> writes:
>
> Someone posted about the danger of IDL variables and
> parameters being of unexpected datatypes. There is
> a little function, 'datatype' (probably based on HELP)
> which is handy for verifying that the type is what
> you expect it is at critical times, such as upon
> entering a routine.
>
> For example:
>
> IDL> j=1L
> IDL> if datatype(j) NE 'LON' then message,'Invalid datatype for j.'
> IDL>
> IDL> j=1.0
> IDL> if datatype(j) NE 'LON' then message,'Invalid datatype for j.'
> % $MAIN$: Invalid datatype for j.
>
>
> I find it very helpful.
> Thank you, Ray Sterner, and here it is:

... routine body omitted ...

Alternatively, one can use the following quick and dirty thingy

_____ _____

Function Type, x

```
;+
; NAME:
; TYPE
; PURPOSE:
; Finds the type class of a variable.
; CATEGORY:
; Programming.
; CALLING SEQUENCE:
; Result = TYPE(X)
; INPUTS:
;    X
; Arbitrary, doesn't even need to be defined.
; OPTIONAL INPUT PARAMETERS:
```

; None.
; KEYWORD PARAMETERS:
; None.
; OUTPUTS:
; Returns the type of X as a long integer, in the (0,9) range.
; OPTIONAL OUTPUT PARAMETERS:
; None.
; COMMON BLOCKS:
; None.
; SIDE EFFECTS:
; None.
; RESTRICTIONS:
; None.
; PROCEDURE:
; Extracts information from the SIZE function.
; MODIFICATION HISTORY:
; Created 15-JUL-1991 by Mati Meron.
;-

```
   dum = size(x)
   return, dum(dum(0) + 1)
end
```

Mati Meron | "When you argue with a fool,
meron@cars3.uchicago.edu | chances are he is doing just the same"
