
Subject: Checking datatypes (datatype.pro)
Posted by [Russ Welti](#) on Thu, 13 Jul 1995 07:00:00 GMT
[View Forum Message](#) <> [Reply to Message](#)

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:

```
;----- --
;+
; NAME:
;   DATATYPE
; PURPOSE:
;   Datatype of variable as a string (3 char or spelled out).
; CATEGORY:
; CALLING SEQUENCE:
;   typ = datatype(var, [flag])
; INPUTS:
;   var = variable to examine.      in
;   flag = output format flag (def=0). in
; KEYWORD PARAMETERS:
;   Keywords:
;   /DESCRIPTOR returns a descriptor for the given variable.
;   If the variable is a scalar the value is returned as
;   a string. If it is an array a description is return
;   just like the HELP command gives. Ex:
;   datatype(ftarr(2,3,5),/desc) gives
;   FLTARR(2,3,5) (flag always defaults to 3 for /DESC).
; OUTPUTS:
;   typ = datatype string or number.  out
;   flag=0  flag=1  flag=2  flag=3
```

```

;      UND      Undefined  0      UND
;      BYT      Byte      1      BYT
;      INT      Integer   2      INT
;      LON      Long      3      LON
;      FLO      Float     4      FLT
;      DOU      Double    5      DBL
;      COM      Complex   6      COMPLEX
;      STR      String    7      STR
;      STC      Structure  8      STC
; COMMON BLOCKS:
; NOTES:
; MODIFICATION HISTORY:
;   Written by R. Sterner, 24 Oct, 1985.
;   RES 29 June, 1988 --- added spelled out TYPE.
;   R. Sterner, 13 Dec 1990 --- Added strings and structures.
;   R. Sterner, 19 Jun, 1991 --- Added format 3.
;   R. Sterner, 18 Mar, 1993 --- Added /DESCRIPTOR.
;   Johns Hopkins University Applied Physics Laboratory.
;
; Copyright (C) 1985, Johns Hopkins University/Applied Physics Laboratory
; This software may be used, copied, or redistributed as long as it is not
; sold and this copyright notice is reproduced on each copy made. This
; routine is provided as is without any express or implied warranties
; whatsoever. Other limitations apply as described in the file disclaimer.txt.
;-
;-----

```

```
function datatype,var, flag0, descriptor=desc, help=help
```

```

if (n_params(0) lt 1) or keyword_set(help) then begin
  print,' Datatype of variable as a string (3 char or spelled out).'
  print,' typ = datatype(var, [flag])'
  print,'   var = variable to examine.      in'
  print,'   flag = output format flag (def=0). in'
  print,'   typ = datatype string or number.  out'
  print,'   flag=0  flag=1  flag=2  flag=3'
  print,'   UND      Undefined  0      UND'
  print,'   BYT      Byte      1      BYT'
  print,'   INT      Integer   2      INT'
  print,'   LON      Long      3      LON'
  print,'   FLO      Float     4      FLT'
  print,'   DOU      Double    5      DBL'
  print,'   COM      Complex   6      COMPLEX'
  print,'   STR      String    7      STR'
  print,'   STC      Structure  8      STC'
  print,' Keywords:'
  print,' /DESCRIPTOR returns a descriptor for the given variable.'
  print,'   If the variable is a scalar the value is returned as'

```

```

    print,'    a string. If it is an array a description is return'
    print,'    just like the HELP command gives. Ex:'
    print,'    datatype(fltarr(2,3,5),/desc) gives'
    print,'    FLTARR(2,3,5) (flag always defaults to 3 for /DESC).'
    return, -1
endif

```

```

if n_params(0) lt 2 then flag0 = 0 ; Default flag.
flag = flag0 ; Make a copy.

```

```

if n_elements(var) eq 0 then begin
    s = [0,0]
endif else begin
    s = size(var)
endelse

```

```

if keyword_set(desc) then flag = 3

```

```

if flag eq 2 then typ = s(s(0)+1)

```

```

if flag eq 0 then begin
    case s(s(0)+1) of
    0:    typ = 'UND'
    7:    typ = 'STR'
    1:    typ = 'BYT'
    2:    typ = 'INT'
    4:    typ = 'FLO'
    3:    typ = 'LON'
    5:    typ = 'DOU'
    6:    typ = 'COM'
    7:    typ = 'STR'
    8:    typ = 'STC'
else:    print,'Error in datatype'
    endcase
endif else if flag eq 1 then begin
    case s(s(0)+1) of
    0:    typ = 'Undefined'
    7:    typ = 'String'
    1:    typ = 'Byte'
    2:    typ = 'Integer'
    4:    typ = 'Float'
    3:    typ = 'Long'
    5:    typ = 'Double'
    6:    typ = 'Complex'
    7:    typ = 'String'
    8:    typ = 'Structure'
else:    print,'Error in datatype'
    endcase

```

```

endif else if flag eq 3 then begin
  case s(s(0)+1) of
    0:  typ = 'UND'
    7:  typ = 'STR'
    1:  typ = 'BYT'
    2:  typ = 'INT'
    4:  typ = 'FLT'
    3:  typ = 'LON'
    5:  typ = 'DBL'
    6:  typ = 'COMPLEX'
    7:  typ = 'STR'
    8:  typ = 'STC'
  else:  print,'Error in datatype'
  endcase
endif

if not keyword_set(desc) then begin
  return, typ ; Return data type.
endif else begin
  if s(0) eq 0 then return, strtrim(var,2) ; Return scalar desc.
  aa = typ+'ARR('
    for i = 1, s(0) do begin
      aa = aa + strtrim(s(i),2)
      if i lt s(0) then aa = aa + ','
    endfor
  aa = aa+')'
  return, aa
endelse

end

```

```

/
Russ Welti      /\
                (c-g)
University of Washington  \-/
Molecular Biotechnology    /
PO Box 352145             /\
Seattle, WA 98195         (a-t)
rwelti@u.washington.edu   \-/
(206) 685 3840 voice (206) 685 7344 FAX /
http://chroma.mbt.washington.edu/graphics/gif/russ.gif

```
