Subject: Re: Using [XYZ]TICKFORMAT for dynamic formatting
Posted by Mark[1] on Thu, 21 May 2009 22:32:57 GMT
View Forum Message <> Reply to Message

On May 22, 9:40 am, Paul van Delst <paul.vande...@noaa.gov> wrote:
You can pass whatever data you want to your TICKFORMAT function via
the TICKFRMTDATA keyword. Thus MGH_TF_LINEAR (below) can be called as

...TICKFORMAT='mgh_tf_linear', TICKFRMTDATA={format: '(F0.3)', scale:
1000}

and will show tick values x 1000 to 3 decimal places.

```
;+
; NAME:
;   MGH_TF_LINEAR
;
; PURPOSE:

;   This function is designed for use with the TICKFORMAT and
;   TICKFRMTDATA properties of IDLgrAxis. Given a real value, it
;   applies a linear transformation & returns a string representing
;   the result.
;
; CALLING SEQUENCE:
;   Result = MGH_TF_LINEAR(Direction, Index, Value)
;
; POSITIONAL PARAMETERS:
;   Direction
;     Axis direction, required by the TICKFORMAT interface but
ignored.
;
;   Index
;     Axis index, required by the TICKFORMAT interface but ignored.
;
;   Value
;     The real value to be formatted.
;
;   Level
;     Level is the index of the axis level for the current tick value
;     to be labeled.  (Level indices start at 0.) This parameter takes
;     its value from the dimension of TICKUNITS.
;
; KEYWORD PARAMETERS:
;   DATA
;     Specify this keyword to control the linear transformation and
;     format. The keyword value should be a structure with one or more
;     of the tags "offset", "scale", "format" and "round". The default
```

```
;     is equivalent to {offset: 0., scale: 1., format: '', round: 0}.
;
; RETURN VALUE:
;   The function returns a scalar string representing:
;       output = data.offset + data.scale * input
;   If data.round is set to a 'true' value then this is rounded.  The
;   format is controlled by data.format. If this is not supplied a
;   default format is generated by MGH_FORMAT_FLOAT.
;
 ;############################################################## ##################
;
; This software is provided subject to the following conditions:
;
; 1.  NIWA makes no representations or warranties regarding the
;     accuracy of the software, the use to which the software may
;     be put or the results to be obtained from the use of the
;     software.  Accordingly NIWA accepts no liability for any loss
;     or damage (whether direct of indirect) incurred by any person
;     through the use of or reliance on the software.
;
; 2.  NIWA is to be acknowledged as the original author of the
;     software where the software is used or presented in any form.
;
 ;############################################################## ##################
;
; MODIFICATION HISTORY:
;   Mark Hadfield, 1999-05:
;     Written.
;   Mark Hadfield, 2001-03:
;     Merged changes by George Constantinides to take advantage of
;     multi level axis of IDL 5.4
;   Mark Hadfield, 2001-05:
;     Added rounding via the "round" tag in DATA. I might generalise
;     this some day.
;   Mark Hadfield, 2003-01:
;     Now accepts a "floor" tag in DATA.
;-

function MGH_TF_LINEAR, direction, index, value, level, DATA=data

   compile_opt DEFINT32
   compile_opt STRICTARR

    offset = 0  &  scale = 1.  &  format = ''  &  round = 0  &  floor
= 0

    if size(data, /TYPE) eq 8 then begin
        if n_elements(data) ne 1 then message, 'The DATA structure
```

```
must have one element'
        if mgh_struct_has_tag(data, 'offset') then offset =
data.offset
        if mgh_struct_has_tag(data, 'scale') then scale = data.scale
        if mgh_struct_has_tag(data, 'format') then format =
data.format
        if mgh_struct_has_tag(data, 'round') then round = data.round
        if mgh_struct_has_tag(data, 'floor') then floor = data.floor
    endif

    if n_elements(level) gt 0 then format=format[level]

    rvalue = offset+scale*value

    if keyword_set(round) then rvalue = round(rvalue)
    if keyword_set(floor) then rvalue = floor(rvalue)

    result = strlen(format) gt 0 $
                ? string(rvalue, FORMAT=format) $
                : mgh_format_float(rvalue)

    ; Force scalar output
    return, result[0]

end
```