
Subject: Re: Superscripts in IDL [x-y]title
Posted by [davidf](#) on Tue, 10 Nov 1998 08:00:00 GMT
[View Forum Message](#) <> [Reply to Message](#)

Henry J. P. Smith (hjpsmith@bigfoot.com) writes:

- > Just did (tried to do) a search on dejaneux to see if I could find
- > this there and got not hits. So here goes.
- >
- > My co-author on a paper would like to see superscripts in the axis
- > titles of simple 2-d plots, e.g. instead of something like cm^{-3} or
- > have $cm^{\{-3\}}$ - using TeX notation.
- >
- > I am using PS fonts so I think there must be a way to do this. Is
- > there any way to get IDL to do it directly? I suppose I could do it in
- > POSTSCRIPT but I don't really know PS and don't have time to learn it
- > right now. At least I think I don't - perhaps the same thing? <G>

```
Plot, Findgen(11), XTitle='cm!U-3!N Units'
```

If I were doing this and wanting to use PostScript fonts,
I would type:

```
Plot, Findgen(11), XTitle='cm!U-3!N Units', Font=0
```

If I wanted to use True-Type fonts (which work both on
the display and in PostScript output) I would type:

```
Plot, Findgen(11), XTitle='cm!U-3!N Units', Font=1
```

Cheers,

David

David Fanning, Ph.D.
Fanning Software Consulting
E-Mail: davidf@dfanning.com
Phone: 970-221-0438, Toll-Free Book Orders: 1-888-461-0155
Coyote's Guide to IDL Programming: <http://www.dfanning.com/>

Note: A copy of this article was e-mailed to the original poster.

Subject: Re: Superscripts in IDL [x-y]title

Posted by [Craig Markwardt](#) on Wed, 11 Nov 1998 08:00:00 GMT

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

hjpsmith@bigfoot.com (Henry J. P. Smith) writes:

>
> Hello,
>
> Just did (tried to do) a search on dejanews to see if I could find
> this there and got not hits. So here goes.
>
> My co-author on a paper would like to see superscripts in the axis
> titles of simple 2-d plots, e.g. instead of something like cm^{-3} or
> have cm^{-3} - using TeX notation.
>
> I am using PS fonts so I think there must be a way to do this. Is
> there any way to get IDL to do it directly? I suppose I could do it in
> POSTSCRIPT but I don't really know PS and don't have time to learn it
> right now. At least I think I don't - perhaps the same thing? <G>
>

Other kind users have posted the standard IDL embedded font control technique. I never figured that out, probably because I discovered the "Tex to IDL" translator first, which converts a Tex-like string into the equivalent IDL embedded sequence. It works great for making subscripts and superscripts, and Greek letters.

This may be just what you are looking for. Download it here:

<ftp://coma.berkeley.edu/pub/mcraig/idl/TeXtoIDL/>

(Snipped from the README...)

PURPOSE:

The purpose of the TeXtoIDL routines is to make it simple to use Greek letters, subscripts and superscripts in making labels for plots in IDL. This is accomplished by allowing the user to use TeX control sequences for Greek letters and special symbols and for sub/superscripts. The TeX control sequences are simple and easy to remember, especially if you already use TeX for writing papers (for those unfamiliar with TeX, an explanation of that notation is below). The translation is done for either vector or PostScript fonts.

...

Best wishes, Craig

--

Craig B. Markwardt, Ph.D. EMAIL: craigmnet@astrog.physics.wisc.edu
Astrophysics, IDL, Finance, Derivatives | Remove "net" for better response

Subject: Re: Superscripts in IDL [x-y]title
Posted by [hjpsmith](#) on Wed, 11 Nov 1998 08:00:00 GMT
[View Forum Message](#) <> [Reply to Message](#)

On Wed, 11 Nov 1998 02:16:25 GMT, menakkis@my-dejanews.com wrote:

> hjpsmith@bigfoot.com (Henry J. P. Smith) wrote:
>> My co-author on a paper would like to see superscripts in the axis
>> titles of simple 2-d plots, e.g. instead of something like cm^{*-3} ot
>> have cm^{-3} - using TeX notation.
>>
>> I am using PS fonts so I think there must be a way to do this. Is
>> there any way to get IDL to do it directly? I suppose I could do it in
>> POSTSCRIPT but I don't really know PS and don't have time to learn it
>> right now. At least I think I don't - perhaps the same thing? <G>
>
> I speak from a standpoint of relative ignorance (I usually don't mess with
> fonts :-), but I think something like this IS quite easy. Search the IDL
> online docs for "Embedded Formatting Commands" and all shall be revealed. In
> brief, !E embedded in your character string will kick following text up to
> superscript, and !N will put following text back to normal. This stuff
> reportedly works for various devices (on a pot-luck basis, apparently). It
> DOES work for PS, on IDL/NT at least.
>

Thanks a lot to David and Peter! This should do it.

Why is it so hard sometimes to find such simple solutions? <G> Or is
it just me? <GG>

Regards,

Henry

Subject: Re: Superscripts in IDL [x-y]title
Posted by [Martin Schultz](#) on Wed, 11 Nov 1998 08:00:00 GMT
[View Forum Message](#) <> [Reply to Message](#)

Henry J. P. Smith wrote:

> Hello,
>
> Just did (tried to do) a search on dejaneWS to see if I could find
> this there and got not hits. So here goes.
>
> My co-author on a paper would like to see superscripts in the axis
> titles of simple 2-d plots, e.g. instead of something like cm^{*-3} ot

> have cm^{-3} - using TeX notation.
 >
 > I am using PS fonts so I think there must be a way to do this. Is
 > there any way to get IDL to do it directly? I suppose I could do it in
 > POSTSCRIPT but I don't really know PS and don't have time to learn it
 > right now. At least I think I don't - perhaps the same thing? <G>
 >
 > Any suggestions? TIA
 >
 > Regards,
 >
 > Henry Smith

Please find attached two routines that should facilitate this task: strchem and strsci.

Here is an example that produces a string which will be output as

$$\begin{array}{c} + \\ \text{NH} \\ 4 \end{array}$$

```
print,strchem(strchem('NH4+',/sub),/super,special='+-')
```

```
result: 'NH!!4!n!u+!n'
```

B

... and strsci produces a string in format A x 10 given a numeric argument:

```
print,strsci( 2000000, format='(i1)' )
```

```
result: '2 x 10!u6!n'
```

Hope this helps,

Martin.

--

 Dr. Martin Schultz
 Department for Engineering&Applied Sciences, Harvard University
 109 Pierce Hall, 29 Oxford St., Cambridge, MA-02138, USA

phone: (617)-496-8318

fax : (617)-495-4551

e-mail: mgs@io.harvard.edu

Internet-homepage: <http://www-as.harvard.edu/people/staff/mgs/>

```

; $Id: strchem.pro,v 1.1 1998/10/09 19:53:32 mgs Exp $
;-----
;+
; NAME:
;   STRCHEM (function)
;
; PURPOSE:
;   Superscripts or subscripts numbers and special
;   characters ('x', 'y') found in strings containing
;   names of chemical species.
;
; CATEGORY:
;   String Utilities
;
; CALLING SEQUENCE:
;   Result = STRCHEM( STR [,keywords] )
;
; INPUTS:
;   STR    -> The input string containing the name of the
;            chemical species (e.g. 'NOx', 'H2O', CxO2, etc, )
;
; KEYWORD PARAMETERS:
;   /SUB   -> Will cause numbers and special characters to
;            be subscripted. This is the default.
;
;   /SUPER -> Will cause numbers and special characters to
;            be superscripted.
;
;   SPECIALCHARS -> a string with characters that shall be sub- or
;                   superscripted. Defaults are '0123456789xyXY' for
;                   /SUB and '+-0123456789' for /SUPER
;
;   PROTECT -> internal keyword used to protect certain characters
;              from being super or subscripted. May be useful to
;              circumvent troubles. See example below.
;
;   /TRIM   -> perform a strtrim( ,2) on the result
;
; OUTPUTS:
;   Returns a string with formatting characters included
;
; SUBROUTINES:
;   None
;
; REQUIREMENTS:

```

```

; Example 3 uses STRWHERE function.
;
; NOTES:
;
; EXAMPLE:
; print,strchem('C2H5O2 [pptv]')
;
; ; prints "C!l2!nH!l5!nO!l2!n [pptv]"
;
; print,strchem(strchem('NH4+',/sub),/super,special='+-')
;
; ; prints NH!l4!n!u+!n.
;
; s0 = '(H2O2)2' ; supposed to be H2O2 squared
; protect = strlen(s0)-1 ; protect last character
; s1 = strchem(s0,protect=protect)
; s2 = strchem(s1,/super,protect=protect)
; print,s1,'->',s2
;
; ; prints (H!l2!nO!l2!n)2->(H!l2!nO!l2!n)!u2!n
; ; without protect the "square" would have been subscripted
;
; MODIFICATION HISTORY:
; bmy, 01 Jun 1998: VERSION 1.00
; mgs, 02 Jun 1998: VERSION 1.10 - rewritten
; mgs, 11 Jun 1998:
; - removed IS_ION keyword
; - changed default specialchars for SUPER
; mgs, 22 Sep 1998:
; - added TRIM keyword
;
;-----
; Copyright (C) 1998, Bob Yantosca, Harvard University
; This software is provided as is without any warranty
; whatsoever. It may be freely used, copied or distributed
; for non-commercial purposes. This copyright notice must be
; kept with any copy of this software. If this software shall
; be used commercially or sold as part of a larger package,
; please contact the author to arrange payment.
; Bugs and comments should be directed to bmy@io.harvard.edu
; with subject "IDL routine strchem"
;-----

```

Function StrChem, Str, Super=Super, Sub=Sub, SpecialChars=SpecialChars, \$
Protect=Protect, Trim=Trim

; Error checking

```

on_error, 2

; Return empty string if no string is passed
if ( n_elements( Str ) eq 0 ) then return, ""

; temporary copy of Str
tmp = Str

; Keyword default settings
Sub = ( keyword_set( Sub ) )
Super = ( keyword_set( Super ) ) * ( 1 - Sub)

; set up string with characters to sub(super)script
if (n_elements(specialchars) eq 0) then begin
  if (sub) then $
    specialchars = '0123456789xyXY' $
  else $
    specialchars = '+-0123456789'
endif

; convert to byte array
BStr = byte(specialchars)

; here are the formatting characters
BE = (byte('!'))[0]
BA = (byte('!'))[0]
if (Super) then BA = (byte('u'))[0]
BN = (byte('n'))[0]

; convert string argument to byte array and loop through,
; inserting the formatting characters at each occurrence of
; a specialchar
; (obsolete Trick: loop backwards in order to simplify ionic case)
BStr = byte(tmp)
Res = 0B

; create local protect array and expand protect if passed
LProtect = intarr(n_elements(BStr))
if (n_elements(Protect) gt 0) then $
  LProtect[Protect] = 1

RProtect = 0 ; resulting Protect array

done = 0
i = n_elements(BStr)-1

```

```

while (not done) do begin
  ind = where(BS eq BStr[i])
  if (ind(0) ge 0 AND not LProtect[i]) then begin
    Res = [ Res, BN, BE, BStr[i], BA, BE ]
    RProtect = [ RProtect, 1, 1, 1, 1, 1 ]
  endif else begin
    Res = [ Res, BStr[i] ]
    RProtect = [ RProtect, 0 ]
  endelse

  i = i - 1

  if (i lt 0) then Done = 1
endwhile

; eliminate first (zero) character and revert "string"
Res = Reverse(Res[1:.*])

; same with new Protect array which will be returned
RProtect = Reverse(RProtect[1:.*])
Protect = where(RProtect gt 0)

; convert byte array back to string and return
result = string(Res)
if (keyword_set(TRIM)) then result = strtrim(result,2)
return,result

```

end

```

; $Id: strsci.pro,v 1.1 1998/10/09 19:53:32 mgs Exp $
;----- --
;+
; NAME:
;   STRSCI (function)
;
; PURPOSE:
;   Given a number, returns a string of that      B
;   number in scientific notation format ( e.g. A x 10 )
;
; CATEGORY:
;   String Utilities
;
; CALLING SEQUENCE:
;   Result = STRSCI( DATA [,keywords] )
;
; INPUTS:
;   DATA    -> A floating point or integer number to be

```



```

;         converted into a power of 10.
;
;
; KEYWORD PARAMETERS:
;   FORMAT  -> The format specification used in the string
;             conversion for the coefficient part (i.e. the
;             "A" of "A x 10^B"). Default is 'f12.2'.
;
;
;   /POT_ONLY -> Will return only the "power of 10" part of the
;             string (i.e. the "10^B"). Default is to return
;             the entire string (e.g. "A x 10^B" )
;
;
;   /MANTISSA_ONLY -> return only mantissa of the string
;
;
;   /SHORT -> return 10^0 as '1' and 10^1 as '10'
;
;
;   /TRIM -> don't insert blanks
;
; OUTPUTS:
;   None
;
; SUBROUTINES:
;   None
;
; REQUIREMENTS:
;   None
;
; NOTES:
;   Need a better symbol than the 'x' for the multiplier...
;
; EXAMPLE:
;   Result = STRSCI( 2000000, format='(i1)' )
;   print, result
;           ;
;           ;           6
;   ;   prints 2 x 10!u6!n, which gets plotted as 2 x 10
;
;
; MODIFICATION HISTORY:
;   bmy, 28 May 1998: VERSION 1.00      B
;   - now returns string of the form A x 10
;   mgs, 29 May 1998:
;   - bug fix: now allows negative numbers
;   - keyword MANTISSA_ONLY added
;   - default format changed to f12.2
;   bmy, 02 Jun 1998:
;   - renamed to STRSCI ("STRing SCientific notation"),
;   mgs, 03 Jun 1998:
;   - added TRIM keyword
;   mgs, 22 Sep 1998:

```

```

; - added SHORT keyword
; - modified handling of TRIM keyword
; mgs, 24 Sep 1998:
; - bug fix with SHORT flag
;
;-
; Copyright (C) 1998, Bob Yantosca, Harvard University
; This software is provided as is without any warranty
; whatsoever. It may be freely used, copied or distributed
; for non-commercial purposes. This copyright notice must be
; kept with any copy of this software. If this software shall
; be used commercially or sold as part of a larger package,
; please contact the author to arrange payment.
; Bugs and comments should be directed to bmy@io.harvard.edu
; with subject "IDL routine strsci"
;-----

```

```

function StrSci, Data, Format=Format, POT_Only=POT_Only, $
      MANTISSA_ONLY=MANTISSA_ONLY, SHORT=SHORT, TRIM=TRIM

```

```

; Error checking
on_error, 2

; Make sure DATA is passed
if ( n_elements( Data ) eq 0 ) then begin
;   print, 'DATA must be passed to EXP_STR!!'
   return, ""
endif

; Default value for FORMAT
if ( not keyword_set( Format ) ) then Format = '(f12.2)'
POT_Only = keyword_set( POT_Only )
MANTISSA_Only = keyword_set( MANTISSA_Only )

; Take the common log of Data
Log10Data = alog10( abs(Data) * 1d0 )
sign = (data lt 0.)

; Compute the Mantissa (frac part) and Characteristic (int part)
Characteristic = fix( Log10Data )
Mantissa      = Log10Data - Characteristic

; String for the coefficient part,
; The coefficient is just antilog of the Mantissa
A = strtrim( string( 10d0^Mantissa, Format=Format ), 2 )
if (sign) then A = '-' + A

```

```

; String for the power of 10 part
B = '10!u' + strtrim( string( Characteristic ), 2 ) + '\n'
if (keyword_set(SHORT)) then begin
  if (Characteristic eq 0) then B = '1'
  if (Characteristic eq 1) then B = '10'
endif

; composite string
result = A + ' x ' + B
if (keyword_set(SHORT) AND B eq '1') then result = A
if ( POT_Only ) then result = B
if (MANTISSA_ONLY ) then result = A

; eliminate blanks if TRIM keyword is set
if (keyword_set(TRIM)) then result = strcompress(result,/remove_all)

return, result

end

```

File Attachments

- 1) [strchem.pro](#), downloaded 135 times
 - 2) [strsci.pro](#), downloaded 113 times
-

Subject: Re: Superscripts in IDL [x-y]title
 Posted by [menakkis](#) on Wed, 11 Nov 1998 08:00:00 GMT
[View Forum Message](#) <> [Reply to Message](#)

hjpsmith@bigfoot.com (Henry J. P. Smith) wrote:

- > My co-author on a paper would like to see superscripts in the axis
- > titles of simple 2-d plots, e.g. instead of something like cm**⁻³ ot
- > have cm^{-3} - using TeX notation.
- >
- > I am using PS fonts so I think there must be a way to do this. Is
- > there any way to get IDL to do it directly? I suppose I could do it in
- > POSTSCRIPT but I don't really know PS and don't have time to learn it
- > right now. At least I think I don't - perhaps the same thing? <G>

I speak from a standpoint of relative ignorance (I usually don't mess with fonts :-), but I think something like this IS quite easy. Search the IDL online docs for "Embedded Formatting Commands" and all shall be revealed. In brief, !E embedded in your character string will kick following text up to superscript, and !N will put following text back to normal. This stuff reportedly works for various devices (on a pot-luck basis, apparently). It DOES work for PS, on IDL/NT at least.

Peter Mason

-----== Posted via Deja News, The Discussion Network ==-----
http://www.dejanews.com/ Search, Read, Discuss, or Start Your Own
