Subject: Re: need more plotting symbols please
Posted by don.woodraska on Mon, 02 Oct 2006 16:17:54 GMT
View Forum Message <> Reply to Message

David,

I modified your code, symcat.pro, to support negative symbols. For those who don't know, negative symbols produce lines that connect the points in a plot. Thank you David for writing clean code that allowed me to only change 3 lines.

Don Woodraska

Save this file with the same name as David's symcat.pro

```
;+
; NAME:
;       SYMCAT
;
; PURPOSE:
;
;       This function provides a symbol catalog for specifying a number
of plotting symbols.
;
; AUTHOR:
;
;       FANNING SOFTWARE CONSULTING
;       David Fanning, Ph.D.
;       1645 Sheely Drive
;       Fort Collins, CO 80526 USA
;       Phone: 970-221-0438
;       E-mail: davidf@dfanning.com
;       Coyote's Guide to IDL Programming: http://www.dfanning.com
;
; CATEGORY:
;
;       Graphics
;
; CALLING SEQUENCE:
;
;       Plot, findgen(11), PSYM=SYMCAT(theSymbol)
;
; INPUTS:
;
;       theSymbol:    The number of the symbol you wish to use.
Possible values are:
;
;       0  : No symbol.
```

```
;       1  : Plus sign.
;       2  : Asterisk.
;       3  : Dot (period).
;       4  : Open diamond.
;       5  : Open upward triangle.
;       6  : Open square.
;       7  : X.
;       8  : Open circle.
;       9  : Open downward triangle.
;      10  : Open rightfacing triangle.
;      11  : Open leftfacing triangle.
;      12  : Filled diamond.
;      13  : Filled square.
;      14  : Filled circle.
;      15  : Filled upward triangle.
;      16  : Filled downward triangle.
;      17  : Filled rightfacing triangle.
;      18  : Filled leftfacing triangle.
;      19  : Hourglass.
;      20  : Filled Hourglass.
;      21  : Bowtie.
;      22  : Filled bowtie.
;      23  : Standing Bar.
;      24  : Filled Standing Bar.
;      25  : Laying Bar.
;      26  : Filled Laying Bar.
;      27  : Hat up.
;      28  : Hat down.
;      29  : Hat right.
;      30  : Hat down.
;      31  : Big cross.
;      32  : Filled big cross.
;      33  : Circle with plus.
;      34  : Circle with X.
;      35  : Upper half circle.
;      36  : Filled upper half circle.
;      37  : Lower half circle.
;      38  : Filled lower half circle.
;      39  : Left half circle.
;      40  : Filled left half circle.
;      41  : Right half circle.
;      42  : Filled right half circle.
;      43  : Star.
;      44  : Filled star.
;
; RETURN VALUE:
;
;       The return value is whatever is appropriate for passing along
```

```
;       to the PSYM keyword of (for example) a PLOT or OPLOT command.
;       For the vast majority of the symbols, the return value is 8.
;
; KEYWORDS:
;
;       None.
;
; MODIFICATION HISTORY:
;
;       Written by David W. Fanning, 2 September 2006. Loosely based on
the
;         program MC_SYMBOL introduced on the IDL newsgroup 1
September 2006,
;         and MPI_PLOTCONFIG__DEFINE from the Coyote Library.
;
;       Added support for negative arguments (connected symbols) Don
Woodraska
;         September 2, 2006
;-
;
 ;############################################################## ################
;
; LICENSE
;
; This software is OSI Certified Open Source Software.
; OSI Certified is a certification mark of the Open Source Initiative.
;
; Copyright © 2006 Fanning Software Consulting.
;
; This software is provided "as-is", without any express or
; implied warranty. In no event will the authors be held liable
; for any damages arising from the use of this software.
;
; Permission is granted to anyone to use this software for any
; purpose, including commercial applications, and to alter it and
; redistribute it freely, subject to the following restrictions:
;
; 1. The origin of this software must not be misrepresented; you must
;    not claim you wrote the original software. If you use this
software
;    in a product, an acknowledgment in the product documentation
;    would be appreciated, but is not required.
;
; 2. Altered source versions must be plainly marked as such, and must
;    not be misrepresented as being the original software.
;
; 3. This notice may not be removed or altered from any source
distribution.
```

```
;
; For more information on Open Source Software, visit the Open Source
; web site: http://www.opensource.org.
;
 ;############################################################## ################

FUNCTION SymCat, theSymbol

   On_Error, 2

   ; Error checking.
   IF N_Elements(theSymbol) EQ 0 THEN RETURN, 0
   IF (abs(theSymbol) GT 44) THEN Message, 'Symbol number out of
defined range.'
   theSymbol = Fix(theSymbol)

   ; Define helper variables for creating circles.
   phi = Findgen(36) * (!PI * 2 / 36.)
   phi = [ phi, phi(0) ]

   ; Use user defined symbol by default.
   result = 8

   CASE abs(theSymbol) OF

      0  : result = 0
   ; No symbol.
      1  : result = 1
   ; Plus sign.
      2  : result = 2
   ; Asterisk.
      3  : result = 3
   ; Dot (period).
      4  : UserSym, [ 0, 1, 0, -1, 0 ], [ 1, 0, -1, 0, 1 ]
   ; Open diamond.
      5  : UserSym, [ -1, 0, 1, -1 ], [ -1, 1, -1, -1 ]
   ; Open upward triangle.
      6  : UserSym, [ -1, 1, 1, -1, -1 ], [ 1, 1, -1, -1, 1 ]
   ; Open square.
      7  : result = 7
   ; X.
      8  : UserSym, cos(phi), sin(phi)
   ; Open circle.
      9  : UserSym, [ -1, 0, 1, -1 ], [  1, -1, 1, 1 ]
   ; Open downward triangle.
     10  : UserSym, [ -1, 1, -1, -1 ], [1, 0, -1, 1 ]
   ; Open rightfacing triangle.
     11  : UserSym, [ 1, -1, 1, 1 ], [1, 0, -1, 1 ]
```

```
   ; Open leftfacing triangle.
     12  : UserSym, [ 0, 1, 0, -1, 0 ], [ 1, 0, -1, 0, 1 ], /Fill
   ; Filled diamond.
     13  : UserSym, [ -1, 1, 1, -1, -1 ], [ 1, 1, -1, -1, 1 ], /Fill
   ; Filled square.
     14  : UserSym, Cos(phi), Sin(phi), /Fill
   ; Filled circle.
     15  : UserSym, [ -1, 0, 1, -1 ], [ -1, 1, -1, -1 ], /Fill
   ; Filled upward triangle.
     16  : UserSym, [ -1, 0, 1, -1 ], [  1, -1, 1, 1 ], /Fill
   ; Filled downward triangle.
     17  : UserSym, [ -1, 1, -1, -1 ], [1, 0, -1, 1 ], /Fill
   ; Filled rightfacing triangle.
     18  : UserSym, [ 1, -1, 1, 1 ], [1, 0, -1, 1 ], /Fill
   ; Filled leftfacing triangle.
     19  : UserSym, [-1, 1,-1,1,-1], [-1,-1, 1,1,-1]
   ; Hourglass.
     20  : UserSym, [-1, 1,-1,1,-1], [-1,-1, 1,1,-1], /Fill
   ; Filled Hourglass.
     21  : UserSym, [-1,-1, 1,1,-1], [-1, 1,-1,1,-1]
   ; Bowtie.
     22  : UserSym, [-1,-1, 1,1,-1], [-1, 1,-1,1,-1], /Fill
   ; Filled bowtie.
     23  : UserSym, [-0.5,-0.5, 0.5, 0.5,-0.5], [-1, 1, 1,-1,-1]
   ; Standing Bar.
     24  : UserSym, [-0.5,-0.5, 0.5, 0.5,-0.5], [-1, 1, 1,-1,-1],
/Fill   ; Filled Standing Bar.
     25  : UserSym, [-1,-1, 1, 1,-1], [-0.5, 0.5, 0.5,-0.5,-0.5]
   ; Laying Bar.
     26  : UserSym, [-1,-1, 1, 1,-1], [-0.5, 0.5, 0.5,-0.5,-0.5],
/Fill   ; Filled Laying Bar.
     27  : UserSym, [-1, -0.5, -0.5, 0.5, 0.5, 1, -1], [-0.7, -0.7,
0.7, 0.7, -0.7, -0.7, -0.7] ; Hat up.
     28  : UserSym, [-1, -0.5, -0.5, 0.5, 0.5, 1, -1], [0.7, 0.7,
-0.7, -0.7, 0.7, 0.7, 0.7]    ; Hat down.
     29  : UserSym, [-0.7, -0.7, 0.7, 0.7, -0.7, -0.7, -0.7], [-1,
-0.5, -0.5, 0.5, 0.5, 1, -1] ; Hat right.
     30  : UserSym, [0.7, 0.7, -0.7, -0.7, 0.7, 0.7, 0.7], [-1, -0.5,
-0.5, 0.5, 0.5, 1, -1]    ; Hat down.
     31  : UserSym, [1, 0.3, 0.3, -0.3, -0.3, -1, -1, -0.3, -0.3, 0.3,
0.3, 1, 1], $
                 [0.3, 0.3, 1, 1, 0.3, 0.3, -0.3, -0.3, -1, -1,
-0.3, -0.3, 0.3]            ; Big cross.
     32  : UserSym, [1, 0.3, 0.3, -0.3, -0.3, -1, -1, -0.3, -0.3, 0.3,
0.3, 1, 1], $
                 [0.3, 0.3, 1, 1, 0.3, 0.3, -0.3, -0.3, -1, -1,
-0.3, -0.3, 0.3], /Fill      ; Filled big cross.
     33  : UserSym, [1,.866, .707, .500, 0,-.500,-.707,-.866,-1, $
```

```
                -.866,-.707,-.500, 0, .500, .707, .866, 1, -1, 0,
0, 0], $
               [0,.500, .707, .866, 1, .866, .707, .500, 0, $
                -.500,-.707,-.866,-1,-.866,-.707,-.500, 0, 0, 0,
1, -1]                ; Circle with plus.
    34  : UserSym, [1,.866, .707, .500, 0,-.500,-.707,-.866,-1, $
                -.866,-.707,-.500, 0, .500, .707, .866, 1, $
                .866,.707,-.707,    0, .707,-.707], $
               [0,.500, .707, .866, 1, .866, .707, .500, 0, $
                -.500,-.707,-.866,-1,-.866,-.707,-.500, 0, $
                .500,.707,-.707,    0,-.707, .707]
                   ; Circle with X.
    35  : UserSym, [Cos(phi[0:18]), Cos(phi[0])], [Sin(phi[0:18]),
Sin(phi[0])]-0.5          ; Upper half circle.
    36  : UserSym, [Cos(phi[0:18]), Cos(phi[0])], [Sin(phi[0:18]),
Sin(phi[0])]-0.5, /Fill    ; Filled upper half circle.
    37  : UserSym, [Cos(phi[18:36]), Cos(phi[18])], [Sin(phi[18:36]),
Sin(phi[18])]+0.5       ; Lower half circle.
    38  : UserSym, [Cos(phi[18:36]), Cos(phi[18])], [Sin(phi[18:36]),
Sin(phi[18])]+0.5, /Fill ; Filled lower half circle.
    39  : UserSym, [Cos(phi[9:27]), Cos(phi[9])]+0.5,
[Sin(phi[9:27]), Sin(phi[9])]          ; Left half circle.
    40  : UserSym, [Cos(phi[9:27]), Cos(phi[9])]+0.5,
[Sin(phi[9:27]), Sin(phi[9])], /Fill    ; Filled left half circle.
    41  : UserSym, [Cos(phi[27:36]), Cos(phi[0:9]),
Cos(phi[27])]-0.5, $
               [Sin(phi[27:36]), Sin(phi[0:9]), Sin(phi[27])]
                   ; Right half circle.
    42  : UserSym, [Cos(phi[27:36]), Cos(phi[0:9]),
Cos(phi[27])]-0.5, $
               [Sin(phi[27:36]), Sin(phi[0:9]), Sin(phi[27])],
/Fill                  ; Filled right half circle.
    43  : UserSym, [-1,-.33, 0,.33,1, .33, 0,-.33,-1], $
               [ 0, .33, 1,.33,0,-.33,-1,-.33, 0]
                   ; Star.
    44  : UserSym, [-1,-.33, 0,.33,1, .33, 0,-.33,-1], $
               [ 0, .33, 1,.33,0,-.33,-1,-.33, 0], /Fill
                   ; Filled star.
  ENDCASE

  if theSymbol gt 0 then theSign = 1 else theSign = -1
  RETURN, result * theSign
END
 ;------------------------------------------------------- ------------------------------------------------------------ ------
```