URL: http://cxc.harvard.edu/ciao3.4/chips_pickpoints.html
Last modified: December 2006

*AHELP for CIAO 3.4*

# chips_pickpoints

Context: chips

***Jump to:*** Description Examples Bugs See Also

## Synopsis

Read 1 or more cursor positions from ChIPS.

## Syntax

```
Float_Type chips_pickpoints( [numpoints] )
```

## Description

This is the S−Lang version of the PICKPOINTS command. It allows a user to select 1 or more positions on a plot and returns the values in an array. The number of points is determined by the optional parameter numpoints; if not supplied it defaults to 1, otherwise it must be an integer with value greater than 0. On error, the routine returns NULL.

The return value is a one−dimensional array of floats where the selected points are stored in (xi,yi) order. The routines reshape() and _reshape() from the S−Lang Run−Time Library can be used to change the format of the data as shown in the examples below; see "ahelp reshape" and "ahelp _reshape" for more information.

## Example 1

```
chips> clear
chips> limits 0 10 0 5
chips> pnts = chips_pickpoints

Click LMB or tap spacebar to pick point.
Click RMB or type 'q' to quit picking points.

Point picked: (5.656155, 4.306668)
chips> print(pnts)
5.65615
4.30667
chips> print(pnts[0])
5.65615
chips> print(pnts[1])
4.30667
```

```
chips> typeof(pnts)
Array_Type
chips> _typeof(pnts)
Float_Type
```

Here we create a plot with limits 0−10 and 0−5 on the X and Y axes and then call the S−Lang version of PICKPOINTS. Since we only wanted to select one point, we did not supply an argument − or "()"'s − to the chips_pickpoints() call.

The screen output and behaviour is the same as if PICKPOINTS had been used. The difference is that the selected values are returned as a one−dimensional array; the variable pnts is used to store the data.

## Example 2

```
chips> clear
chips> limits 0 10 0 5
chips> pnts = chips_pickpoints(3)

Click LMB or tap spacebar to pick point.
Click RMB or type 'q' to quit picking points.

Point picked: (8.651068, 4.245632)
Point picked: (5.343642, 2.890631)
Point picked: (5.031129, 2.133974)
chips> print(pnts)
8.65107
4.24563
5.34364
2.89063
5.03113
2.13397
chips> vmessage("Point 1 = %f,%f",pnts[0],pnts[1])
Point 1 = 8.651068,4.245632
chips> vmessage("Point 2 = %f,%f",pnts[2],pnts[3])
Point 2 = 5.343642,2.890631
chips> vmessage("Point 3 = %f,%f",pnts[4],pnts[5])
Point 3 = 5.031129,2.133974
```

Here we ask for three points rather than 1 (and so need to use "()" around the argument). Note that the output array is still one−dimensional, with the values stored in (xi,yi) order.

## Example 3

```
chips> clear
chips> limits 100 120 200 300
chips> pnts = chips_pickpoints(4)

Click LMB or tap spacebar to pick point.
Click RMB or type 'q' to quit picking points.

Point picked: (106.052490, 277.100006)
Point picked: (105.739983, 227.054245)
Point picked: (115.010376, 237.796600)
Point picked: (110.426857, 209.720001)
chips> reshape(pnts,[4,2])
chips> print(pnts)
```

```
106.052 277.1
105.74 227.054
115.01 237.797
110.427 209.72
chips> xpnts = pnts[*,0]
chips> print(xpnts)
106.052
105.74
115.01
110.427
chips> ypnts = pnts[*,1]
chips> print(ypnts)
277.1
227.054
237.797
209.72
```

Here we use the reshape() routine to change the format of the returned array from one−dimensional − with eight elements − to a two−dimensional array with 4x2 elements.

We then use the array−indexing capabilities of S−Lang to take a "slice" of this array to extract just the x (xpnts) and y (ypnts) values.

Alternatively we could have used

```
pos1 = pnts[0,*]
```

to create a two−element array pos1 which contains the x and y coordinates of the first point in the list.

# Example 4

```
chips> clear
chips> limits 0 10 0.01 10
chips> log y
chips> lpnts = chips_pickpoints(2)

Click LMB or tap spacebar to pick point.
Click RMB or type 'q' to quit picking points.

Point picked: (2.192879, −1.239185)
Point picked: (9.458393, −0.487297)
chips> print(lpnts)
2.19288
−1.23919
9.45839
−0.487297
chips> i=[1,3]
chips> writeascii(stdout,lpnts[i],10^lpnts[i])
−1.23919 0.0576521
−0.487297 0.325614
```

Here we use a plot in which one of the axes (the Y axis) has log spacing. The return values for that axis are still in logarithmic units (which is why they are both less than 0). The writeascii() call writes out the y values of the selected points in the first column, and the corresponding linear values in the second column. We take advantage of S−Lang's array−indexing mechanism to restrict the output to just use the second and fourth elements (ie "[1,3]") of the lpnts array.

# Example 5

The previous examples have shown chips_pickpoints() being used from the ChIPS prompt. It can also be used from a S−Lang script, as the following example shows.

```
unix% cat pick.sl
% load the ChIPS functions
require("chips");

% turn off redrawing until the plot is finished
() = chips_auto_redraw(0);

% set the limits
() = chips_set_xrange(0,10);
() = chips_set_yrange(10,20);

% plot some data
variable x = [1:9];
variable y = x + 10;
chips.symbolstyle = _chips−>none;
chips.linestyle   = _chips−>simpleline;
() = curve(x,y);
() = chips_auto_redraw(1);

% call pickpoints
variable out = chips_pickpoints(3);
if ( andelse { out == NULL }
             { length(out) != 6 } )
  error( "Did not select 3 points from the plot" );

print("\nYou selected the following values:");
variable i;
foreach ( [1:3] ) {
  i = ();
  () = printf( "  point %d  x = %4.2f  y = %5.2f\n",
               i, out[2*(i−1)], out[2*(i−1)+1] );
}
% end of script
```

which, when run using slsh, will create output something like (depending on the actual points selected):

```
unix% slsh pick.sl

Click LMB or tap spacebar to pick point.
Click RMB or type 'q' to quit picking points.

Point picked: (3.468972, 15.146487)
Point picked: (6.880163, 14.975967)
Point picked: (9.093795, 18.173878)

You selected the following values:
  point 1  x = 3.47  y = 15.15
  point 2  x = 6.88  y = 14.98
  point 3  x = 9.09  y = 18.17
```

# Bugs

See the bugs page for ChIPS on the CIAO website for an up−to−date listing of known bugs.

## See Also

*chips*

chips_auto_redraw, chips_clear, chips_color_name, chips_color_value, chips_get_pane, chips_get_xrange, chips_get_xscale, chips_get_yrange, chips_get_yscale, chips_get_zrange, chips_get_zscale, chips_label, chips_line, chips_redraw, chips_set_pane, chips_set_xrange, chips_set_xscale, chips_set_yrange, chips_set_yscale, chips_set_zrange, chips_set_zscale, chips_split, chips_version

The Chandra X−Ray Center (CXC) is operated for NASA by the Smithsonian Astrophysical Observatory.
60 Garden Street, Cambridge, MA 02138 USA.

URL: http://cxc.harvard.edu/ciao3.4/chips_pickpoints.html
Last modified: December 2006