



AHELP for CIAO 3.4

setplot

Context: [sherpa](#)

Jump to: [Description](#) [Examples](#) [Parameters](#) [NOTES](#) [Bugs](#) [See Also](#)

Synopsis

An alternative interface to the Sherpa plot customisation variables

Syntax

```
setplot( plotname [, option1, option2, ... ] )
```

Description

One of the major changes in CIAO 3.0 is that the appearance of Sherpa plots, such as those created by the LPLOT command, can be altered by changing the sherpa plot configuration variables (also known as the sherpa state object). This allows you to make the most common changes – e.g. the color and style of lines and symbols – "permanent", in that any plot will be automatically created with your preferred customizations.

While the new system is powerful, and makes setting the behaviour of Sherpa from scripts much easier, it is not the most natural for the occasional user. The setplot.sl file – available from the [CIAO scripts page](#) – provides an alternative interface to setting Sherpa plots that may be easier to use.

This document describes the routine available in setplot.sl; it is also described in the "[Changing the look of Sherpa plots using setplot.sl](#)" thread.

Overview

When run, the setplot() routine will ask you questions on how the plot should look. It will only ask you those values it needs, so if you choose to not use symbols for a plot then it does not ask you for other information on symbols, such as their color or style. The set of questions it asks depends on the plot type you wish to change – which is taken from the first argument of the setplot() call, as described next.

Using setplot()

Once the setplot.sl script has been installed (see the [CIAO scripts page](#) for instructions), the setplot() routine can be used. For instance, to set the look of all "fit" plots you would say

```
sherpa> setplot("fit")
Should the X-axis have logarithmic spacing? (no):
Should the Y-axis have logarithmic spacing? (no):
Draw error bars along the X-axis? (no):
... more questions ...
```

where in this example we have decided not to change the settings of the first three questions.

Default values

The default value for each prompt is taken from the corresponding field in the appropriate "sherpa.*plot" field. So, in the example above, the default values for the first three questions are taken from the values of `sherpa.fitplot.x_log`, `sherpa.fitplot.y_log`, and `sherpa.fitplot.x_errorbars`. See "ahelp sherpa.fitplot" for more information on these fields, and note that the order you are asked for them does not match the order they are listed in the `sherpa.fitplot` structure. There is also ahelp documentation on `sherpa.plot`, `sherpa.dataplot`, and `sherpa.resplot`.

The list of all the parameters prompted for by the `setplot()` routine can be found by "plist setplot". Note that not all these parameters are prompted for each plot type and – as noted in the parameter file – the names of the parameters do not match directly the fields in the sherpa plot variables.

Since the default values are taken from the plot variables and NOT the parameter file there is no point in changing the values in the parameter file using `pset`.

Once you have used this function to set the plots to your own design you can use the `save_state()` function – see "ahelp save_state" – to save the settings so they will be automatically loaded into every new Sherpa session.

Hidden parameters

One parameter is currently hidden, and so will not be prompted for when the `setplot()` command is run. This is the `zlabelsize` parameter – which corresponds to the `.zlabel_size` field in the plot variables. If you do wish to change the value then you either have to do it manually: e.g.

```
sherpa> sherpa.dataplot.zlabel_size = 2
```

or call the function with the parameter as an option:

```
sherpa> setplot("data", "zlabelsize=2")
```

Missing parameters

All the plot configuration variables contain `prefunc` and `postfunc` variables. Since they are only used to create advanced customizations of plots they can not be changed by the `setplot()` command. For further information on these variables see "ahelp sherpa-plot-hooks" and the "[Advanced customization of Sherpa plots](#)" thread.

The fields of the `sherpa.multiplot` plotting variable are currently not settable by the `setplot()` routine. See "ahelp sherpa.multiplot" for more information on this object.

Example 1

```
sherpa> setplot()
Usage: setplot( plotname [, arg1, ..., argN ] )

where plotname is one of:
bnoise bstaterrors weight noise udata bweight mcounts bfit uback source
bufit bexpmap ratio bstatistic expmap chi squ bmcounnts model bg psf
bsyserrors group ufit bcounts bmodel residuals bpsf arf statistic data
barf berrors fit errors bresiduals dcounts staterrors bgroup back
bdelchi syserrors bchisqu filter bfilter delchi bratio kernel backfit
```

```
background
```

If called without any arguments then a "usage" message will be displayed. It is best to include the "()" when doing this to make sure that the command is interpreted as a S-Lang function and not as an abbreviation of a Sherpa command.

Example 2

```
sherpa> setplot("data")
```

This will prompt you for settings that change any new plots created by "lplot data" – including commands such as "lplot 2 arf data". Once the command has run try "lplot data" – assuming you have a dataset loaded – to see the new settings in action.

See "ahelp sherpa.dataplot" for the set of plots that this call will configure.

Example 3

```
sherpa> setplot("fit")
```

This is similar to the previous example except that the parameters relevant to the fit-style plots – which includes ufit.

See "ahelp sherpa.fitplot" for the set of plots that this call will configure.

Example 4

```
sherpa> setplot("delchi")
```

This is similar to the previous example except that the parameters relevant to the delchi-style plots.

See "ahelp sherpa.fitplot" for the set of plots that this call will configure.

Example 5

```
sherpa> setplot("arf")
```

This is similar to the previous example except that the parameters relevant to ARF plots.

See "ahelp sherpa.plot" for the set of plots that this call will configure.

Parameters

name	type	def
<u>xlog</u>	boolean	no
<u>ylog</u>	boolean	no
<u>xlabelsize</u>	real	1.5
<u>ylabelsize</u>	real	1.5

<u>zlabelsize</u>	real	1.5
<u>titlesize</u>	real	1.5
<u>tickvalssize</u>	real	1.5
<u>xerror</u>	boolean	no
<u>yerror</u>	boolean	yes
<u>errorstyle</u>	string	bar
<u>errortype</u>	string	both
<u>curvestyle</u>	string	step
<u>curvecolor</u>	string	default
<u>symbolstyle</u>	string	none
<u>symbolcolor</u>	string	default
<u>symbolsize</u>	real	2
<u>fitcurvestyle</u>	string	step
<u>fitcurvecolor</u>	string	default
<u>fitsymbolstyle</u>	string	none
<u>fit_symbolcolor</u>	string	default
<u>fit_symbolsize</u>	real	2

Detailed Parameter Descriptions

Parameter=**xlog** (boolean default=no)

Should the X-axis have logarithmic spacing?

Should the X-axis be drawn with a logarithmic – rather than linear – scale? This field can also be set – for all plotting objects – by the `set_log()` and `set_xlog()` functions: see "ahelp set_log" for more details.

Corresponds to the `x_log` field of the `sherpa.*plot` objects.

Parameter=**ylog** (boolean default=no)

Should the Y-axis have logarithmic spacing?

Should the Y-axis be drawn with a logarithmic – rather than linear – scale? This field can also be set – for all plotting objects – by the `set_log()` and `set_ylog()` functions: see "ahelp set_log" for more details.

Corresponds to the `y_log` field of the `sherpa.*plot` objects.

Parameter=**xlabelsize** (real default=1.5)

Size of labels along X-axis

What size should the labels be along the X-axis?

Corresponds to the `xlabel_size` field of the `sherpa.*plot` objects.

Parameter=ylabelsize (real default=1.5)

Size of labels along Y-axis

What size should the labels be along the Y-axis?

Corresponds to the `ylabel_size` field of the `sherpa.*plot` objects.

Parameter=zlabelsize (real default=1.5)

Size of labels along Z-axis

What size should the labels be along the Z-axis? This is a hidden parameter and so will not be prompted for by the `setplot()` command. To change its value you can call the function with the parameter as an extra option:

```
sherpa> setplot("data", "zlabelsize=2")
```

or set its value directly:

```
sherpa> sherpa.dataplot.zlabel_size=2
```

(replacing `dataplot` with the appropriate object name).

Corresponds to the `zlabel_size` field of the `sherpa.*plot` objects.

Parameter=titlesize (real default=1.5)

Size of title

What size should the title of the plot be?

Corresponds to the `title_size` field of the `sherpa.*plot` objects.

Parameter=tickvalssize (real default=1.5)

Size of numbers along axes

What size should the numeric labels along the axes be?

Corresponds to the `tickvals_size` field of the `sherpa.*plot` objects.

Parameter=xerror (boolean default=no)

Draw error bars along the X-axis?

Should error bars be drawn for the X-axis? This field can also be set – for all plotting objects – by the `set_erron()` and `set_xerron()` functions and unset by the `set_erroff()` and `set_xeroff()` functions: see "ahelp set_erron" and "ahelp set_erroff" for more details.

Corresponds to the `x_errorbars` field of the `sherpa.*plot` objects.

Parameter=yerror (boolean default=yes)

Draw error bars along the Y-axis?

Should error bars be drawn for the Y-axis? This field can also be set – for all plotting objects – by the `set_erron()` and `set_yerron()` functions and unset by the `set_errroff()` and `set_yerroff()` functions: see "ahelp set_erron" and "ahelp set_errroff" for more details.

Corresponds to the `y_errorbars` field of the `sherpa.*plot` objects.

Parameter=errorstyle (string default=bar)

Style of error bars?

If error bars are being drawn then what style should be used? The valid options are `bar` or `standard` and correspond to the `errorstyle` option of the `ERRS` command of ChIPS.

Corresponds to the `errs_style` field of the `sherpa.*plot` objects.

Parameter=errortype (string default=both)

Type of error bars?

If error bars are being drawn then should they be drawn for both upper and lower values? The valid options are `both`, `none`, `up`, `down`, or `dn` and correspond to the `errortype` option of the `ERRS` command of ChIPS. Note that `down` and `dn` are synonyms.

Corresponds to the `errs_type` field of the `sherpa.*plot` objects.

Parameter=curvestyle (string default=step)

Line style to connect points

How should points be connected? The options are `step`, `histo`, `noline`, and `simpleline` and correspond to the `curvestyle` option of the `CURVE` command of ChIPS. Note that `step` and `histo` are synonyms.

Corresponds to the `curvestyle` field of the `sherpa.*plot` objects.

Parameter=curvecolor (string default=default)

Color to draw line connecting points

If a line is used to connect the points – i.e. `curvestyle` is not set to `noline` – then what color should the line be? The options are `default`, `black`, `blue`, `cyan`, `green`, `magenta`, `red`, `white`, and `yellow` and correspond to the `color` option of the `CURVE` command of ChIPS.

Corresponds to the `curvecolor` field of the `sherpa.*plot` objects.

Parameter=symbolstyle (string default=none)

Symbol used to mark points

What symbol – if any – should be used to mark points? This corresponds to the `symbolstyle` option of the `SYMBOL` command of ChIPS. The allowed values are: `none`, `bigpoint`, `block`, `circle`, `cross`, `diamond`, `downtri`, `point`, `soliddiamond`, `soliddowntri`, `soliduptri`, `square`, and `uptri`.

Corresponds to the `symbolstyle` field of the `sherpa.*plot` objects.

Parameter=symbolcolor (string default=default)

Color of symbols

If a symbol is used to mark the points – i.e. symbolstyle is not set to none – then what color should the symbols be? The options are default, black, blue, cyan, green, magenta, red, white, and yellow and correspond to the color option of the SYMBOL command of ChIPS.

Corresponds to the symbolcolor field of the sherpa.*plot objects.

Parameter=symbolsize (real default=2)

Size of symbols

If a symbol is used to mark the points – i.e. symbolstyle is not set to none – then what size should the symbols be? This corresponds to the size_value option of the SYMBOL command of ChIPS.

Corresponds to the symbolsize field of the sherpa.*plot objects.

Parameter=fitcurvestyle (string default=step)

Fit: line style to connect points

How should the points representing the fit values be connected? The options are step, histo, noline, and simpleline and correspond to the curvestyle option of the CURVE command of ChIPS. Note that step and histo are synonyms.

Corresponds to the fit_curvestyle field of the sherpa.fitplot object.

Parameter=fitcurvecolor (string default=default)

Fit: color to draw line connecting points

If a line is used to connect the fitted points – i.e. fitcurvestyle is not set to noline – then what color should the line be? The options are default, black, blue, cyan, green, magenta, red, white, and yellow and correspond to the color option of the CURVE command of ChIPS.

Corresponds to the fit_curvecolor field of the sherpa.fitplot object.

Parameter=fitsymbolstyle (string default=none)

Fit: symbol used to mark points

What symbol – if any – should be used to mark the best-fit points? This corresponds to the symbolstyle option of the SYMBOL command of ChIPS. The allowed values are: none, bigpoint, block, circle, cross, diamond, downtri, point, soliddiamond, soliddowntri, soliduptri, square, and uptri.

Corresponds to the fit_symbolstyle field of the sherpa.fitplot object.

Parameter=fit_symbolcolor (string default=default)

Fit: color of symbols

If a symbol is used to mark the best-fit points – i.e. fitsymbolstyle is not set to none – then what color should

the symbols be? The options are default, black, blue, cyan, green, magenta, red, white, and yellow and correspond to the color option of the SYMBOL command of ChIPS.

Corresponds to the fit_symbolcolor field of the sherpa.fitplot object.

Parameter=fit_symbols_size (real default=2)

Fit: size of symbols

If a symbol is used to mark the best-fit points – i.e. fitsymbolstyle is not set to none – then what size should the symbols be? This corresponds to the size_value option of the SYMBOL command of ChIPS.

Corresponds to the fit_symbols_size field of the sherpa.fitplot object.

NOTES

This script is not an official part of the CIAO release but is made available as "contributed" software via the [CIAO scripts page](#). Please see the [installation instructions page](#) for help on installing the package.

Parameter values and the Sherpa state object

Sherpa plots can be configured using fields in the Sherpa state object (also called a configuration variable). The routine described here hides this interface from the user, instead using the parameter library to provide a more interactive, tool-like, interface. However, they will pick up the current values from the state object and use them as the default values in the parameter prompts, as shown in the example below:

```
sherpa> sherpa.resplot.x_log = 1
sherpa> setplot("delchi")
Should the X-axis have logarithmic spacing? (yes):
Should the Y-axis have logarithmic spacing? (no):
...
```

The names of the parameters of the functions do not exactly match the fields of the state objects (e.g. sherpa.*plot.x_log is represented here by the xlog parameter). See the parameter description above for the relationship between the two sets of names.

Bugs

Parameter values appear in the Sherpa history buffer

Once a routine has finished, and responses that were made to the parameter prompts will be added to the Sherpa history buffer. This means that if you use the "up arrow" to repeat any commands you will first have to scroll through all the responses you made.

See Also

chandra

guide

sherpa

bye, calc_kcorr, dataspace, dcounts, dollarsign, echo, eflux, eqwidth, erase, flux, get, get dcounts sum, get dir, get eflux, get eqwidth, get filename, get flux2d, get flux str, get lfactorial, get mcounts sum, get pflux, get source components, get verbose, groupbycounts, guess, is, journal, list, list par, mcounts, numbersign, paramest, plot eprof, plot rprof, prompt, reset, run, set, set analysis, set axes, set coord, set dataspace, set dir, set verbose, sherpa-module,

Ahelp: setplot – CIAO 3.4

sherpa_plotfns, sherpa_utils, show, simspec, use, version

The Chandra X-Ray Center (CXC) is operated for NASA by the Smithsonian Astrophysical Observatory.
60 Garden Street, Cambridge, MA 02138 USA.
Smithsonian Institution, Copyright © 1998–2006. All rights reserved.

URL:
<http://cxc.harvard.edu/ciao3.4/setplot.html>
Last modified: December 2006

